UNIVERSITY OF CALIFORNIA SAN DIEGO

# Research on Polynomial and Tensor Optimization

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Mathematics

by

Zi Yang

Committee in charge:

Professor Jiawang Nie, Chair
Professor Alexander Cloninger
Professor Sonia Martínez
Professor Yixiao Sun
Professor Danna Zhang

2021

The dissertation of Zi Yang is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deep and sincere gratitude to my supervisor, Professor Jiawang Nie, for invaluable advice, encouragement, and continuous support during my five years of Ph.D. studies. His immense knowledge, patient guidance, and plentiful experience helped me go through lots of difficult time in completing the Ph.D. and this dissertation. I am extremely grateful for what he has offered me. Without his tremendous supervision and encouragement, it would be impossible for me to complete the doctoral degree and the dissertation.

I am also grateful to other members of my dissertation committee, Professor Alexander Cloninger, Professor Sonia Martínez, Professor Yixiao Sun, and Professor Danna Zhang. It is a great honor for me to have these outstanding professors from many fields serve on my dissertation committee. Their knowledge and experience are really beneficial in completing my dissertation. The Department of Mathematics at UC San Diego provided fertile ground and an unsurpassable research environment to complete my Ph.D. study. Its fantastic faculty and excellent staff gave generous support to my graduate life.

Moreover, I would like to give my appreciation to my family and friends. Especially, I want to express my deepest gratitude with all my heart to my fiancée Chenyang Duan, for her endless support and love.

In this dissertation, some materials have been published, or been submitted for publication.

The Chapter 2, in full, is a reprint of the material as it appears in *SIAM Journal on Optimization* 2018 [119]. The dissertation author coauthored this paper with Nie, Jiawang and Zhang, Xinzhen.

The Chapter 3, in full, is a reprint of the material as it appears in *Foundations of Computational Mathematics* 2021 [118]. The dissertation author coauthored this paper with Nie, Jiawang and Zhou, Guangming.

The Sections 4.1-4.6 of the Chapter 4 are a reprint of the material as it appears in *SIAM Journal on Matrix Analysis and Applications* 2020 [110]. The dissertation author coauthored this paper with Nie, Jiawang. The Section 4.7 of the Chapter 4 is part of the publication that has been accepted for publication in *Linear and Multilinear Algebra* 2021 [49]. The dissertation author coauthored this paper with Dressler, Mareike and Nie, Jiawang.

The Chapter 5, in full, has been accepted for publication in *Vietnam Journal of Mathematics* 2021 [59]. The dissertation author coauthored this paper with Guo, Bingni and Nie, Jiawang.

## VITA

| | |
|---|---|
| 2016 | B. S. in Mathematics, University of Science and Technology of China |
| 2021 | Ph. D. in Mathematics, University of California San Diego |

## PUBLICATIONS

B. Guo, J. Nie, and Z. Yang, "Learning Diagonal Gaussian Mixture Models and Incomplete Tensor Decompositions", accepted by *Vietnam Journal of Mathematics*, 2021.

M. Dressler, J. Nie, and Z. Yang, "Separability of Hermitian Tensors and PSD Decompositions", accepted by *Linear and Multilinear Linear Algebra*, 2021.

J. Nie, Z. Yang, and G. Zhou, "The Saddle Point Problem of Polynomials", *Foundations of Computational Mathematics*, pp. 1-37, 2021.

J. Nie, and Z. Yang, "Hermitian Tensor Decompositions", *SIAM Journal on Matrix Analysis and Applications*, Vol. 41, pp. 1115-1144, 2020.

J. Nie, Z. Yang, and X. Zhang, "A Complete Semidefinite Algorithm for Detecting Copositive Matrices and Tensors", *SIAM Journal on Optimization*, Vol. 28, pp. 2902-2921, 2018.

B. Chen, Z. Yang, and Z.W. Yang, "An Algorithm for Low-rank Matrix Factorization and Its Applications", *Neurocomputing*, Vol. 275, pp. 1012-1020, 2018.

ABSTRACT OF THE DISSERTATION

**Research on Polynomial and Tensor Optimization**

by

Zi Yang
Doctor of Philosophy in Mathematics
University of California San Diego, 2021
Professor Jiawang Nie, Chair

Polynomial optimization considers optimization problems defined by polynomials. In contrast to classical nonlinear optimization, it aims at finding global optimizers. Tensors are natural higher-order generalizations of matrices and are closely related to polynomials and moments. They are powerful tools in studying tensors. Many tensor problems can be formulated as polynomial optimization problems.

We propose a complete semidefinite relaxation algorithm for detecting the copositivity of a symmetric tensor. We show that the detection can be done by solving a finite number of semidefinite relaxations for all tensors.

For the saddle point problem of polynomials, we give an algorithm for computing saddle points. We show that: i) if there exists a saddle point, our algorithm can get one by solving a finite number of Lasserre type semidefinite relaxations; ii) if there is no saddle point, our algorithm can detect its nonexistence.

Hermitian tensors are generalizations of Hermitian matrices, but they have very different properties. Canonical basis Hermitian tensors, real Hermitian tensors, special matrix flattenings, positive semidefiniteness, and separability are studied. We further study

how to detect separability of Hermitian tensors. We formulate this as a truncated moment problem and then provide a semidefinite relaxation algorithm to solve it.

The problem of learning diagonal Gaussian mixture models can be formulated as computing incomplete symmetric tensor decompositions. We use generating polynomials to compute incomplete symmetric tensor decompositions and approximations. Then the tensor approximation is used to learn diagonal Gaussian mixture models. When the first and third order moments are sufficiently accurate, we show that the obtained parameters for the Gaussian mixture models are also highly accurate.

# Chapter 1

# Introduction

## 1.1 Polynomial optimization

In this section, we review some basics about positive polynoimals, localizing matrices, and polynomial optimization.

Denote by $\mathbb{R}[x]$ the ring of polynomials in $x$ with real coefficients in $\mathbb{R}$. The $\mathbb{R}[x]_d$ is the set of polynomials whose degrees $\leq d$. For $\alpha := (\alpha_1, \ldots, \alpha_n) \in \mathbb{N}^n$ with an integer $n > 0$, denote $|\alpha| := \alpha_1 + \cdots + \alpha_n$. For an integer $d > 0$, denote $\mathbb{N}^n_d := \{\alpha \in \mathbb{N}^n \mid |\alpha| \leq d\}$. For $x = (x_1, \ldots, x_n)$ and $\alpha = (\alpha_1, \ldots, \alpha_n)$, denote

$$x^\alpha := x_1^{\alpha_1} \cdots x_n^{\alpha_n}, \quad [x]_d := \begin{bmatrix} 1 & x_1 & \cdots & x_n & x_1^2 & x_1x_2 & \cdots & x_n^d \end{bmatrix}^T.$$

An ideal $I$ of $\mathbb{R}[x]$ is a subset such that $I \cdot \mathbb{R}[x] \subseteq I$ and $I + I \subseteq I$. For a tuple $p = (p_1, \ldots, p_k)$ of polynomials in $\mathbb{R}[x]$, Ideal$(p)$ denotes the smallest ideal containing all $p_i$, which is the set $p_1 \cdot \mathbb{R}[x] + \cdots + p_k \cdot \mathbb{R}[x]$. In computation, we often need to work with the *truncation*:

$$Ideal(p)_{2k} := p_1 \cdot \mathbb{R}[x]_{2k-\deg(p_1)} + \cdots + p_k \cdot \mathbb{R}[x]_{2k-\deg(p_k)}.$$

A polynomial $\sigma$ is said to be a sum of squares (SOS) if $\sigma = s_1^2 + \cdots + s_k^2$ for some polynomials $s_1, \ldots, s_k$. Checking if a polynomial is SOS can be done by solving a semidefinite program (SDP) [81]. If a polynomial is SOS, then it is nonnegative everywhere. But, the reverse may not be true. The set of all SOS polynomials in $x$ is denoted by $\Sigma[x]$, and its $d$th truncation is $\Sigma[x]_d := \Sigma[x] \cap \mathbb{R}[x]_d$. For a tuple $q = (q_1, \ldots, q_t)$ of polynomials its *quadratic module* is

$$Qmod(q) := \Sigma[x] + q_1 \cdot \Sigma[x] + \cdots + q_t \cdot \Sigma[x].$$

We often need to work with the truncation

$$Qmod(q)_{2k} := \Sigma[x]_{2k} + q_1 \cdot \Sigma[x]_{2k-\deg(g_1)} + \cdots + q_t \cdot \Sigma[x]_{2k-\deg(q_t)}.$$

A subset $A \subseteq \mathbb{R}[x]$ is said to be *archimedean* if there exists $\sigma \in A$ such that $\sigma(x) \geq 0$ defines a compact set in $\mathbb{R}^n$. If $Ideal(p)+Qmod(q)$ is archimedean, then the set $K := \{p(x) = 0,\, q(x) \geq 0\}$ must be compact. The reverse is not always true. When $Ideal(p) + Qmod(q)$ is archimedean, every polynomial that is positive on $K$ must belong to $Ideal(p) + Qmod(q)$. This is the so-called Putinar's Positivstellensatz [124]. Interestingly, under some optimality conditions, if a polynomial is nonnegative (but not strictly positive) over $K$, then it belongs to $Ideal(p) + Qmod(q)$. This is shown in [103].

The set $\mathbb{R}^{\mathbb{N}^n_d}$ is the space of all real vectors that are labeled by $\alpha \in \mathbb{N}^n_d$. That is, every $y \in \mathbb{R}^{\mathbb{N}^n_d}$ can be labeled as

$$y \,=\, (y_\alpha)_{\alpha \in \mathbb{N}^n_d}.$$

Such $y$ is called a *truncated multi-sequence* (tms) of degree $d$ [115]. The tms $y$ is said to *admit* a *Borel measure* $\mu$ if it satisfies that $y_\alpha \,=\, \int x^\alpha \mathtt{d}\mu$, $\forall\, \alpha \in \mathbb{N}^n_d$. If it exists, such a $\mu$ is called a *representing measure* for $y$, and $y$ is said to admit the measure $\mu$.

For a polynomial $f \in \mathbb{R}[x]_r$ that is written as

$$f = \sum_{|\alpha| \leq \mathbb{N}^n_r} f_\alpha x^\alpha,$$

with $r \leq d$, we define the operation

$$\langle f, y \rangle = \sum_{|\alpha| \leq \mathbb{N}^n_r} f_\alpha y_\alpha. \tag{1.1}$$

Note that $\langle f, y \rangle$ is linear in $y$ for fixed $f$, and is linear in $f$ for fixed $y$. For a polynomial $q \in \mathbb{R}[x]_{2k}$ and the integer $t = k - \lceil \deg(q)/2 \rceil$, the outer product $q(x)[x]_t[x]_t^T$ is a symmetric matrix of length $\binom{n+t}{t}$. It can be expanded as

$$q(x)[x]_t[x]_t^T \,=\, \sum_{\alpha \in \mathbb{N}^n_{2k}} x^\alpha Q_\alpha,$$

for constant symmetric matrices $Q_\alpha$. For $y \in \mathbb{R}^{\mathbb{N}^n_{2k}}$, denote the symmetric matrix

$$L_q^{(k)}[y] \,:=\, \sum_{\alpha \in \mathbb{N}^n_{2k}} y_\alpha Q_\alpha. \tag{1.2}$$

It is called the $k$th *localizing matrix* of $q$ and generated by $y$. For given $q$, $L_q^{(k)}[y]$ is linear in $y$. Clearly, if $q(u) \geq 0$ and $y = [u]_{2k}$, then $L_q^{(k)}[y] = q(u)[u]_t[u]_t^T \succeq 0$. ($X \succeq 0$ means that $X$ is positive semidefinite.) For instance, if $n = k = 2$ and $q = 1 - x_1 - x_1x_2$, then

$$L_q^{(2)}[y] = \begin{bmatrix} y_{00} - y_{10} - y_{11} & y_{10} - y_{20} - y_{21} & y_{01} - y_{11} - y_{12} \\ y_{10} - y_{20} - y_{21} & y_{20} - y_{30} - y_{31} & y_{11} - y_{21} - y_{22} \\ y_{01} - y_{11} - y_{12} & y_{11} - y_{21} - y_{22} & y_{02} - y_{12} - y_{13} \end{bmatrix}.$$

When $q = 1$ (the constant one polynomial), the localizing matrix $L_1^{(k)}[y]$ reduces to a moment matrix, which we denote as

$$M_k[y] := L_1^{(k)}[y].$$

For instance, when $n = 2$, $k = 3$, the matrix $M_3[y]$ is

$$M_3[y] = \begin{bmatrix} y_{00} & y_{10} & y_{01} & y_{20} & y_{11} & y_{02} & y_{30} & y_{21} & y_{12} & y_{03} \\ y_{10} & y_{20} & y_{11} & y_{30} & y_{21} & y_{12} & y_{40} & y_{31} & y_{22} & y_{13} \\ y_{01} & y_{11} & y_{02} & y_{21} & y_{12} & y_{03} & y_{31} & y_{22} & y_{13} & y_{04} \\ y_{20} & y_{30} & y_{21} & y_{40} & y_{31} & y_{22} & y_{50} & y_{41} & y_{32} & y_{23} \\ y_{11} & y_{21} & y_{12} & y_{31} & y_{22} & y_{13} & y_{41} & y_{32} & y_{23} & y_{14} \\ y_{02} & y_{12} & y_{03} & y_{22} & y_{13} & y_{04} & y_{32} & y_{23} & y_{14} & y_{05} \\ y_{30} & y_{40} & y_{31} & y_{50} & y_{41} & y_{32} & y_{60} & y_{51} & y_{42} & y_{33} \\ y_{21} & y_{31} & y_{22} & y_{41} & y_{32} & y_{23} & y_{51} & y_{42} & y_{33} & y_{24} \\ y_{12} & y_{22} & y_{13} & y_{32} & y_{23} & y_{14} & y_{42} & y_{33} & y_{24} & y_{15} \\ y_{30} & y_{13} & y_{04} & y_{23} & y_{14} & y_{05} & y_{33} & y_{24} & y_{15} & y_{06} \end{bmatrix}.$$

Moment and localizing matrices can be used to construct semidefinite relaxations for polynomial optimization problems. We refer to [141] for a survey on semidefinite programs. Consider the polynomial optimization problem

$$\begin{cases} f^* := \min & f(x) \\ \text{s.t.} & h_i(x) = 0\,(i = 1, \ldots, m_1), \\ & g_j(x) \geq 0\,(j = 1, \ldots, m_2), \end{cases} \tag{1.3}$$

where $f$ and $g_i, h_j$ are all real polynoimals in $x \in \mathbb{R}^n$. A standard approach to solve (1.3) is Lasserre's hierarchy of sum of squares (SOS) relaxations [81]. Let

$$d_0 := \max\{\lceil \deg(f) \rceil/2, \lceil \deg(g_i) \rceil/2, \lceil \deg(h_j) \rceil/2\}_{i=1,\ldots,m_1, j=1,\ldots,m_2}.$$

For $k_0 \geq d_0$, the $k$th order relaxation is

$$\begin{cases} f_k := \min & \langle f, y \rangle \\ \quad s.t. & L_{h_i}^{(k)}[y] = 0 \, (i = 1, \ldots, m_1) \\ & L_{g_j}^{(k)}[y] \succeq 0 \, (j = 1, \ldots, m_2), \\ & y_0 = 1, \, M_k[y] \succeq 0, \, y \in \mathbb{R}^{\mathbb{N}_{2k}^n}. \end{cases} \tag{1.4}$$

It always holds that $f_k \leq f^*$ and the sequence $\{f_k\}$ is monotonically increasing. Under the archimedean condition, the relaxations have asymptotic convergence, i.e., $\lim_{k\to\infty} f_k = f^*$. When $f_k = f^*$ for some $k$, we say the relaxation has finite convergence. A common criterion to check finite convergence is the flat extension [32]. Let $y \in \mathbb{R}^{\mathbb{N}_{2k}^n}$ be a minimizer of the $k$th order relaxation. We say $y$ satisfies flat extension if the following rank condition

$$\text{rank} M_{k-d_c}[z] = \text{rank} M_k[z] \tag{1.5}$$

is satisfied, where

$$d_c := \max\{\lceil \deg(g_i) \rceil / 2, \lceil \deg(h_j) \rceil / 2\}_{i=1,\ldots,m_1, j=1,\ldots,m_2}.$$

Then, we can extract $r = \text{rank} M_k$ minimizers for (1.3). It is implemented in the software `GloptiPoly3` [66]. When the rank condition (1.5) is satisfied for some $k$, the semidefinite relaxations have finite convergence. In polynomial optimization, a more appropriate condition than flat extension is the flat truncation [102].

We refer to [54, 65, 81, 113] for more work about solving polynoimal optimization.

## 1.2 Tensors

Let $\mathbb{F} = \mathbb{C}$ (the complex field) or $\mathbb{R}$ (the real field) and $V_1, \ldots, V_m$ be finitely dimensional vector spaces over $\mathbb{F}$. The dual space of $V_i$ is the set of all linear functionals on $V_i$. Denote by $V_i^*$ the dual space of $V_i$. For each $v_i \in V_i$, let $v_1 \otimes \cdots \otimes v_m$ be the linear functional on $V_1^* \times \cdots \times V_m^*$ such that

$$(v_1 \otimes \cdots \otimes v_m)(s_1, \ldots, s_m) = s_1(v_1) \cdots s_m(v_m),$$

for all $s_i \in V_i^*$. The span of all such linear functionals $v_1 \otimes \cdots \otimes v_m$ is called the tensor product space of $V_1, \ldots, V_m$, denoted by $V_1 \otimes \cdots \otimes V_m$.

The tensor space $\mathbb{F}^{n_1} \otimes \cdots \otimes \mathbb{F}^{n_m}$ is isomorphic to $\mathbb{F}^{n_1 \times \cdots \times n_m}$. Thus, a tensor $\mathcal{A} \in \mathbb{F}^{n_1} \otimes \cdots \otimes \mathbb{F}^{n_m}$ can be represented as a multi-array in $\mathbb{F}^{n_1 \times \cdots \times n_m}$, i.e., $\mathcal{A} = (\mathcal{A}_{i_1 \ldots i_m})$, with $i_k \in \{1, ..., n_k\}$ for $k = 1, \ldots, m$. For convenience, we also call $\mathbb{F}^{n_1 \times \cdots \times n_m}$ the tensor space of order $m$ and dimension $n_1, \ldots, n_m$. When $m = 3$ (resp., 4), they are called cubic (resp., quartic) tensors. For vectors $u_k \in \mathbb{F}^{n_k}$, $k = 1, \ldots, m$, the $u_1 \otimes \cdots \otimes u_m$ denotes their tensor product, i.e., $(u_1 \otimes \cdots \otimes u_m)_{i_1 \ldots i_m} = (u_1)_{i_1} \cdots (u_m)_{i_m}$ for all $i_1, \ldots, i_m$ in the range. Tensors like $u_1 \otimes \cdots \otimes u_m$ are called rank-1 tensors. The *cp rank* of $\mathcal{A}$, denoted as $\mathrm{rank}(\mathcal{A})$, is the smallest $r$ such that

$$\mathcal{A} = \sum\nolimits_{i=1}^{r} u_i^1 \otimes \cdots \otimes u_i^m, \quad u_i^j \in \mathbb{C}^{n_j}. \tag{1.6}$$

In the literature, the decomposition (1.6) is often called a candecomp-parafac or canonical polyadic (CP) decomposition. We refer to [37, 76, 79, 88, 142] for tensor decompositions, and refer to [16, 37, 38, 137] for tensor decomposition methods. For uniqueness of tensor decompostions, we refer to the work [27, 48, 57, 78, 133].

Symmetric tensors are natural generalizations of symmetric matrices. A tensor $\mathcal{A} \in \mathbb{F}^{n \times \cdots \times n}$ of order $m$ is *symmetric* if $\mathcal{A}_{i_1 \ldots i_m}$ is invariant for all permutations of $(i_1, \ldots, i_m)$. The entries of the form $\mathcal{A}_{jj \ldots j}$ are called *diagonal*, while the other entries are called *off-diagonal*. Rank-1 symmetric tensors are multiples of

$$u^{\otimes m} := u \otimes \cdots \otimes u \quad \text{(repeated } m \text{ times)}.$$

For every symmetric tensor, there exist some $u_i \in \mathbb{C}^n$ and $\lambda_i \in \mathbb{C}$ such that

$$\mathcal{A} = \sum\nolimits_{i=1}^{r} \lambda_i u_i^{\otimes m}.$$

The smallest number $r$ is called the *symmetric rank* of $\mathcal{A}$, denoted by $\mathrm{rank}_S(\mathcal{A})$. We refer to [15, 28, 109, 120] for the work on symmetric tensor decompositions. Symmetric tensors can be generalized to partial symmetric tensors [79] and conjugate partial symmetric tensors [56]. A class of interesting symmetric tensors are Hankel tensors [117]. More work about tensor ranks can be found in [29, 144].

For two tensors $\mathcal{A}, \mathcal{B} \in \mathbb{C}^{[n_1, \ldots, n_m]}$, their *inner product* is defined as

$$\langle \mathcal{A}, \mathcal{B} \rangle := \sum\nolimits_{i_1, \ldots, i_m, j_1, \ldots, j_m} \mathcal{A}_{i_1 \ldots i_m j_1 \ldots j_m} \overline{\mathcal{B}_{i_1 \ldots i_m j_1 \ldots j_m}}, \tag{1.7}$$

where $\bar{a}$ denotes the conjugate of the complex number $a$. The *Hilbert-Schmidt norm* of $\mathcal{A}$ is accordingly defined as $||\mathcal{A}|| := \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}$.

For convenience of operations, we define multilinear matrix multiplications for tensors (see [88]). For matrices $M_k \in \mathbb{C}^{p_k \times q_k}$, $k = 1, \ldots, m$, define the matrix-tensor product $(M_1, \ldots, M_m) \times \mathcal{T}$ for $\mathcal{T} \in \mathbb{C}^{q_1 \times \cdots \times q_m}$ such that it gives a linear map from $\mathbb{C}^{q_1 \times \cdots \times q_m}$ to $\mathbb{C}^{p_1 \times \cdots \times p_m}$ and it satisfies

$$(M_1, \ldots, M_m) \times (u_1 \otimes \cdots \otimes u_m) = (M_1 u_1) \otimes \cdots \otimes (M_m u_m),$$

for all rank-1 tensors $u_1 \otimes \cdots \otimes u_m$. The product $(M_1, \ldots, M_m) \times \mathcal{T}$ is a tensor in $\mathbb{C}^{p_1 \times \cdots \times p_m}$. For two tensors $\mathcal{T}_1, \mathcal{T}_2$ of compatible dimensions, it holds that

$$\langle (M_1, \ldots, M_m) \times \mathcal{T}_1, \mathcal{T}_2 \rangle \; = \; \langle \mathcal{T}_1, (M_1^*, \ldots, M_m^*) \times \mathcal{T}_2 \rangle,$$

where the superscript $^*$ denotes the conjugate transpose.

# Notation

The symbol $\mathbb{N}$ denotes the set of nonnegative integers. For $k = 1, \ldots, m$, the $x_k$ denotes the complex vector variable in $\mathbb{C}^{n_k}$. The tuple of all such complex variables is denoted as $x := (x_1, \ldots, x_m)$. For $\mathbb{F} = \mathbb{R}$ or $\mathbb{C}$, denote by $\mathbb{F}[x]$ the ring of polynomials in $x$ with coefficients in $\mathbb{F}$, while $\mathbb{F}[x, \overline{x}]$ denotes the ring of conjugate polynomials in $x$ and $\overline{x}$ with coefficients in $\mathbb{F}$. In the Euclidean space $\mathbb{F}^n$, denote by $e_i$ the $i$th standard unit vector, i.e., the $i$th entry of $e_i$ is one and all others are zeros, while $e$ stands for the vector of all ones. The $I_k$ denotes the $k$-by-$k$ identity matrix. For a vector $u$ in $\mathbb{R}^n$ or $\mathbb{C}^n$, $\|u\|$ denotes its standard Euclidean norm. For a matrix or vector $a$, the $a^*$ denotes its conjugate transpose, $a^T$ denotes its transpose, while $\overline{a}$ denotes its conjugate entrywisely; we use $\mathrm{Re}(a)$ and $\mathrm{Im}(a)$ to denote its real and complex part respectively. For a complex scalar or vector $z$, denote $|z| := \sqrt{z^* z}$. The $\mathrm{int}(S)$ denotes the interior of a set $S$, under the Euclidean topology. The $\mathbb{M}^n$ denotes the set of $n$-by-$n$ Hermitian matrices, while $\mathcal{S}^n$ denotes the set of $n$-by-$n$ real symmetric matrices. If a Hermitian matrix $X$ is positive semidefinite (resp., positive definite), we write that $X \succeq 0$ (resp., $X \succ 0$). The symbol $\otimes$ denotes the tensor product, while $\boxtimes$ denotes the classical Kronecker product. For a tensor product $u \otimes v \otimes \cdots$, we denote by $\mathrm{vec}(u \otimes v \otimes \cdots)$ the column vector of its coefficients in its representation in terms of the basis tensors. For an integer $k > 0$, denote the set $[k] := \{1, \ldots, k\}$. For a real number $t$, the ceiling $\lceil t \rceil$ denotes the smallest integer that is greater than or equal to $t$.

# Chapter 2

# Detection of Copositive Tensors

## 2.1 Copositive tensors

A real symmetric matrix $A \in \mathbb{R}^{n\times n}$ is said to be *copositive* if

$$x^T Ax \geq 0 \quad \forall\, x \in \mathbb{R}^n_+,$$

where $\mathbb{R}^n_+$ is the nonnegative orthant (i.e., the set of nonnegative vectors). If $x^T Ax > 0$ for all $0 \neq x \in \mathbb{R}^n_+$, then $A$ is said to be *strictly copositive*. The set of all $n \times n$ copositive matrices is a cone in $\mathbb{R}^{n\times n}$, which is denoted as $\mathcal{COP}_n$. Copositive matrices were introduced in [93]. They have broad applications, e.g., in quadratic programming [20], dynamical systems and control theory [72, 91], graph theory [36, 50], complementarity problems and variational inequalities [53]. We refer to [6, 51] for surveys on copositive optimization.

A basic problem in optimization is the detection of copositive matrices. Let $\mathcal{S}^n_+$ be the cone of $n \times n$ real symmetric positive semidefinite (psd) matrices, and $\mathcal{N}^n_+$ be the cone of $n \times n$ real symmetric matrices whose entries are all nonnegative. Clearly, it holds that

$$\mathcal{S}^n_+ + \mathcal{N}^n_+ \subseteq \mathcal{COP}_n. \tag{2.1}$$

For $n \leq 4$, the above inclusion is an equality; for $n \geq 5$, the equality does not hold any more [43]. For instance, the Horn matrix [60] is copositive, but it is not a sum of psd and nonnegative matrices. Checking membership of the cone $\mathcal{COP}_n$ is NP-hard [45, 96]. As shown in [75], a matrix $A$ is copositive if and only if it does not have a principal submatrix that has a negative eigenvalue with a positive eigenvector. To apply this testing, one needs to check eigenvalues for all principal submatrices, which grows exponentially in the dimension.

For the case $n = 5$, when the diagonal entries are all ones, $A$ is copositive if and only if the polynomial $\|x\|^2(\sum_{i,j=1}^5 A_{ij}x_i^2x_j^2)$ is a sum of squares [44]. When off-diagonal entries are nonpositive, $A$ is copositive if and only if $A$ is positive semidefinite [69]. When a matrix is tridiagonal or acyclic, its copositivity can be detected in linear time [5, 71]. For testing copositivity for general matrices, there exist methods based on simplicial partition [19, 138]. Another approach for testing copositivity is to use the difference of convexity [7, 52]. A survey about existing results and open problems for copositive matrices can be found in [3].

The concept of copositivity can be naturally generalized to tensors, as in Qi [125]. Let $\mathtt{S}^m(\mathbb{R}^n)$ be the space of symmetric tensors of order $m$ over the vector space $\mathbb{R}^n$. For $\mathcal{A} \in \mathtt{S}^m(\mathbb{R}^n)$, its associated polynomial is

$$\mathcal{A}(x) := \sum_{1\le i_1,i_2,\cdots,i_m\le n} \mathcal{A}_{i_1i_2\cdots i_m}x_{i_1}x_{i_2}\cdots x_{i_m}. \tag{2.2}$$

If $\mathcal{A}(x) \ge 0$ for all $x \in \mathbb{R}^n$, $\mathcal{A}$ is said to be *positive semidefinite* (psd). If $\mathcal{A}(x) \ge 0$ for all $x \in \mathbb{R}^n_+$, $\mathcal{A}$ is said to be *copositive*. Similarly, if $\mathcal{A}(x) > 0$ for all $0 \ne x \in \mathbb{R}^n_+$, $\mathcal{A}$ is said to be *strictly* copositive. Denote by $\mathcal{COP}_{m,n}$ the cone of all copositive tensors in $\mathtt{S}^m(\mathbb{R}^n)$. Clearly, when the order $m = 2$, positive semidefinite (resp., copositive) tensors are the same as positive semidefinite (resp., copositive) matrices. To be psd, a tensor must have even order. An odd order nonzero tensor can never be psd, but it is possibly copositive. For instance, every nonzero tensor with zero diagonal entries and nonnegative off-diagonal ones is copositive, but not psd.

Copositive tensors have broad applications. For instance, some complementarity problems can be formulated by using copositive tensors [23, 135, 136]. The coclique number of a hypergraph can be bounded by tensor copositivity [24]; see Example 2.6. Copositive tensors are useful in vacuum stability [74]. Moreover, some polynomial optimization problems can be formulated as linear conic programs about copositive tensors [122]. We refer to [25, 125, 134, 135] for more applications of copositive tensors.

Detecting tensor copositivity is also a mathematically challenging question. It is also NP-hard, because testing matrix copositivity is a special case. If the off-diagonal entries of a symmetric tensor $\mathcal{A}$ are nonpositive, then $\mathcal{A}$ is copositive if and only if $\mathcal{A}$ is psd [125]. There also exists a characterization of copositive tensors by the eigenpairs of its principal subtensors [134]. Like the matrix case, tensor copositivity can also be tested by algorithms based on simplicial partition. Typically, when a tensor lies in the interior of the copositive cone, the copositivity can be detected by this kind of algorithms. However, if it lies on the

boundary, they usually have difficulties. We refer to [19, 24, 25, 138] for related work.

## 2.2 A complete semidefinite algorithm

We discuss how to detect copositivity of a given tensor. For a symmetric tensor $\mathcal{A} \in \mathtt{S}^m(\mathbb{R}^n)$, let $\mathcal{A}(x)$ be the homogeneous polynomial defined as in (2.2). Clearly, $\mathcal{A}$ is copositive if and only if $\mathcal{A}(x) \geq 0$ for all $x$ belonging to the standard simplex

$$\Delta \,=\, \{x \in \mathbb{R}^n : \, e^T x = 1, \, x \geq 0\}.$$

Consider the optimization problem

$$\left\{ \begin{array}{rl} v^* := \min & \mathcal{A}(x) \\ \text{s.t.} & e^T x = 1, \, (x_1, \ldots, x_n) \geq 0. \end{array} \right. \tag{2.3}$$

Clearly, $\mathcal{A}$ is copositive if and only if the minimum value $v^* \geq 0$. Therefore, testing the copositivity of $\mathcal{A}$ is the same as determining the sign of $v^*$. The problem (2.3) is a polynomial optimization problem. A standard approach for solving it is to apply classical Lasserre relaxations [81]. Since the feasible set is compact and the archimedean condition holds, its asymptotic convergence is always guaranteed.

As proposed in [108], there exist tight relaxations for solving polynomial optimization, whose constructions are based on optimality conditions and Lagrange multiplier expressions. Since its feasible set is compact and nonempty, problem (2.3) must have a global minimizer, say, $u$. The constraints of (2.3) are all affine linear functions. One can see that the linear independence constraint qualification condition holds at $u$. So we have the following optimality conditions (the notation $\nabla$ denotes the gradient):

$$\left\{ \begin{array}{c} \nabla \mathcal{A}(u) \,=\, \lambda_0 e + \sum_{i=1}^n \lambda_i e_i, \\ \lambda_1 u_1 = \cdots = \lambda_n u_n = 0, \;\; \lambda_1 \geq 0, \ldots, \lambda_n \geq 0, \end{array} \right. \tag{2.4}$$

where $\lambda_0, \lambda_1, \ldots, \lambda_n$ are the Lagrange multipliers. By a simple algebraic computation (also see [108]), one can show that (note that $x^T \nabla f(x) = m f(x)$ for all homogeneous polynomials $f(x)$ of degree $m$, because $x^T \nabla x^\alpha = |\alpha| x^\alpha$)

$$\left\{ \begin{array}{rcl} \lambda_0 & = & u^T \nabla \mathcal{A}(u) = m \mathcal{A}(u), \\ \lambda_i & = & \frac{\partial \mathcal{A}(u)}{\partial x_i} - m \mathcal{A}(u) \;\; (i = 1, 2, \ldots, n). \end{array} \right. \tag{2.5}$$

Because of the above expressions, we define new polynomials:

$$p_i := \frac{\partial \mathcal{A}(x)}{\partial x_i} - m\mathcal{A}(x) \ \ (i = 1, 2, \ldots, n). \tag{2.6}$$

Since every optimizer $u$ must satisfy (2.4) and its norm $\|u\| \leq 1$, the optimization problem (2.3) is equivalent to

$$\begin{cases} \min & \mathcal{A}(x) \\ \text{s.t} & e^T x - 1 = p_1(x)x_1 = \cdots = p_n(x)x_n = 0, \\ & 1 - \|x\|^2 \geq 0,\ x_i \geq 0,\ p_i(x) \geq 0\,(i = 1, \ldots, n). \end{cases} \tag{2.7}$$

Then we apply Lasserre's relaxations to solve (2.7). For the orders $k = 1, 2, \ldots$, solve the semidefinite relaxation problem:

$$\begin{cases} v_k := \min & \langle \mathcal{A}(x), y \rangle \\ \text{s.t} & y_0 = 1, L^{(k)}_{e^T x - 1}[y] = 0,\ L^{(k)}_{x_i p_i}[y] = 0\,(i = 1, \ldots, n), \\ & L^{(k)}_{x_i}[y] \succeq 0,\ L^{(k)}_{p_i}[y] \succeq 0\,(i = 1, \ldots, n), \\ & L^{(k)}_{1-\|x\|^2}[y] \succeq 0,\ M_k[y] \succeq 0,\ y \in \mathbb{R}^{\mathbb{N}^n_{2k}}. \end{cases} \tag{2.8}$$

The ball constraint $1 - \|x\|^2 \geq 0$ is redundant in (2.7). There are two major advantages for using it: i) Adding the ball constraint results in tighter relaxations, i.e., (2.8) is stronger than the one without using $1 - \|x\|^2 \geq 0$. ii) If $1 - \|x\|^2 \geq 0$ is not used, there exist numerical difficulties for solving the semidefinite relaxation (2.8).

Note that $v^*$ is also the optimal value of (2.7). The feasible set of (2.7) is contained in the projection of that of (2.8), so the optimal value $v_k$ of (2.8) satisfies

$$v_1 \leq v_2 \leq \cdots \leq v^*.$$

Clearly, if $v_k \geq 0$ for some $k$, then $\mathcal{A}$ is copositive. Combining the above, we can get the following algorithm.

**Algorithm 2.1.** *For a given tensor $\mathcal{A} \in \mathbb{S}^m(\mathbb{R}^n)$, let $m_0 := \lceil m/2 \rceil$ and $k := m_0$. Choose a generic vector $\xi \in \mathbb{R}^{\mathbb{N}^n_m}$. Test the copositivity of $\mathcal{A}$ as follows:*

*Step 1: Solve the semidefinite relaxation (2.8). If its optimal value $v_k \geq 0$, then $\mathcal{A}$ is copositive and stop. If $v_k < 0$, go to Step 2.*

*Step 2: Solve the following semidefinite program*

$$\begin{cases} \min & \langle \xi^T [x]_m, y \rangle \\ s.t & L^{(k)}_{e^T x - 1}[y] = 0,\ L^{(k)}_{x_i}[y] \succeq 0,\ (i \in [n]), \\ & L^{(k)}_{1-\|x\|^2}[y] \succeq 0,\ L^{(k)}_{v_k - \mathcal{A}(x)}[y] \succeq 0, \\ & y_0 = 1, M_k[y] \succeq 0,\ y \in \mathbb{R}^{\mathbb{N}^n_{2k}}. \end{cases} \tag{2.9}$$

*If it is feasible, compute an optimizer $\hat{y}$. If it is infeasible, let $k := k+1$ and go to Step 1.*

*Step 3: Let $u = \big((\hat{y})_{e_1}, \ldots, (\hat{y})_{e_n}\big)$. If $\mathcal{A}(u) < 0$, then $\mathcal{A}$ is not copositive and stop; otherwise, let $k := k+1$ and go to Step 1.*

In Algorithm 2.1, the vector $\xi$ can be chosen as a random vector obeying normal distribution. In MATLAB, we can use the function `randn` to generate each entry of $\xi$. In Step 2, the copositivity of $\mathcal{A}$ is justified by the relationship $v^* \geq v_k$, for all $k \geq m_0$. In Step 3, the point $u$ must belong to the simplex $\Delta$. This is because of the constraints $L^{(k)}_{e^T x - 1}[y] = 0$ and $L^{(k)}_{x_i}[y] \succeq 0$.

In the following, we show that Algorithm 2.1 must terminate within finitely many iterations, for all tensors $\mathcal{A}$. In other words, the copositivity of every $\mathcal{A}$ can be detected correctly by solving finitely many semidefinite relaxations. This is why we call Algorithm 2.1 a complete semidefinte algorithm for detecting tensor copositivity.

**Theorem 2.2** ( [119]). *For all symmetric tensors $\mathcal{A} \in \mathtt{S}^m(\mathbb{R}^n)$, Algorithm 2.1 has the following properties:*

*(i) For all $k \geq m_0$, the semidefinite relaxation (2.8) is feasible and achieves its optimal value $v_k$; moreover, $v_k = v^*$ for all $k$ sufficiently large.*

*(ii) For all $k \geq m_0$, the semidefinite program (2.9) has an optimizer if it is feasible.*

*(iii) If $\mathcal{A}$ is copositive, then Algorithm 2.1 must stop with $v_k \geq 0$, when $k$ is sufficiently large.*

*(iv) If $\mathcal{A}$ is not copositive, then, for almost all $\xi \in \mathbb{R}^{\mathbb{N}^n_m}$ (i.e., $\xi \in \mathbb{R}^{\mathbb{N}^n_m} \backslash \Theta$ for a subset $\Theta \subseteq \mathbb{R}^{\mathbb{N}^n_m}$ of zero Lebesgue measure), Algorithm 2.1 must return a point $u \in \Delta$ with $f(u) < 0$, when $k$ is sufficiently large.*

*Proof.* (i) The feasible set of (2.3) is compact, so it must have a minimizer, say, $u^*$. Then, $u^*$ satisfies (2.4), and hence $u^*$ is a feasible point for (2.7). So, the feasible set of (2.7) is nonempty. This implies that the semidefinite relaxation (2.8) is always feasible. By the constraint $L^{(k)}_{1-\|x\|^2}[y] \succeq 0$, we can show that the feasible set of (2.8) is compact, as follows. First, we can see that

$$1 = y_0 \geq y_{2e_1} + \cdots + y_{2e_n}.$$

So, $0 \leq y_{2e_i} \leq 1$ since each $y_{2e_i} \geq 0$ (because $M_k[y] \succeq 0$). Second, for all $0 < |\alpha| \leq k-1$, the $(\alpha,\alpha)$th diagonal entry of $L^{(k)}_{1-\|x\|^2}[y]$ is nonnegative, so

$$y_{2\alpha} \geq y_{2\alpha+2e_1} + \cdots + y_{2\alpha+2e_n}. \tag{2.10}$$

By choosing $\alpha = e_1, \ldots, e_n$, the same argument can show that $0 \leq y_{2\beta} \leq 1$ for all $|\beta| \leq 2$. By repeatedly applying (2.10), one can further get that $0 \leq y_{2\beta} \leq 1$ for all $|\beta| \leq k$. Third, note that the diagonal entries of $M_k[y]$ are precisely $y_{2\beta}$ with $|\beta| \leq k$. Since $M_k[y] \succeq 0$, all the entries of $M_k[y]$ must be between $-1$ and 1. This means that $y$ is bounded, hence the feasible set of (2.8) is compact. Therefore, (2.8) must achieve its optimal value $v_k$.

To prove $v_k = v^*$ for all $k$ sufficiently large, note that (2.7) is the same as the optimization

$$\begin{cases} \min & \mathcal{A}(x) \\ \text{s.t} & e^T x - 1 = p_1(x)x_1 = \cdots = p_n(x)x_n = 0, \\ & x_i \geq 0,\, p_i(x) \geq 0,\, i = 1, \ldots, n. \end{cases} \tag{2.11}$$

Its corresponding Lasserre's relaxations are

$$\begin{cases} v'_k := \min & \langle \mathcal{A}(x), y \rangle \\ \text{s.t} & L^{(k)}_{e^T x - 1}[y] = 0,\, L^{(k)}_{x_i p_i}[y] = 0\, (1 \leq i \leq n), \\ & L^{(k)}_{x_i}[y] \succeq 0,\, L^{(k)}_{p_i}[y] \succeq 0\, (1 \leq i \leq n), \\ & y_0 = 1, M_k[y] \succeq 0,\, y \in \mathbb{R}^{\mathbb{N}^n_{2k}}, \end{cases} \tag{2.12}$$

for the orders $k = 1, 2, \ldots$. The optimal value of (2.11) is also $v^*$. The feasible set of (2.8) is contained in that of (2.12), so

$$v'_k \leq v_k \leq v^*, \quad k = m_0, m_0+1, \ldots. \tag{2.13}$$

Next, we show that the set of polynomials

$$F := \left\{ (1 - e^T x)\phi + \sum_{j=1}^n x_j \Big( \sum_{\ell} s^2_{j,\ell} \Big) : \phi \in \mathbb{R}[x],\, s_{j,\ell} \in \mathbb{R}[x] \right\}$$

is archimedean, i.e., there exists $f \in F$ such that the inequality $f(x) \geq 0$ defines a compact set in $\mathbb{R}^n$. This is true for $f = 1 - \|x\|^2$, because of the identity

$$1 - \|x\|^2 = (1 - e^T x)(1 + \|x\|^2) + \sum_{i=1}^{n} x_i(1 - x_i)^2 + \sum_{i \neq j} x_i^2 x_j. \tag{2.14}$$

By Theorem 3.3 of [108], we know that $v_k' = v^*$ when $k$ is sufficiently large. Hence, the relation (2.13) implies that $v_k = v^*$ for all $k$ sufficiently large.

(ii) The semidefinite program (2.9) also has the constraint $L^{(k)}_{1-\|x\|^2}[y] \succeq 0$. By the same argument as in (i), we know that the feasible set of (2.9) is compact. So, it must have an optimizer if it is feasible.

(iii) Clearly, $\mathcal{A}$ is copositive if and only if $v^* \geq 0$. By the item (i), $v_k = v^*$ for all $k$ big enough. Therefore, if $\mathcal{A}$ is copositive, we must have $v_k \geq 0$ for all $k$ large enough.

(iv) If $\mathcal{A}$ is not copositive, then $v^* < 0$. By the item (i), there exists $k_1 \in \mathbb{N}$ such that $v_k = v^*$ for all $k \geq k_1$. Hence, for all $k \geq k_1$, (2.9) is the same as

$$\begin{cases} \min & \langle \xi^T [x]_m, y \rangle \\ \text{s.t} & L^{(k)}_{e^T x - 1}[y] = 0,\ L^{(k)}_{x_i}[y] \succeq 0,\ (i \in [n]), \\ & L^{(k)}_{1-\|x\|^2}[y] \succeq 0,\ L^{(k)}_{v^* - \mathcal{A}(x)}[y] \succeq 0, \\ & (y)_0 = 1, M_k[y] \succeq 0,\ y \in \mathbb{R}^{\mathbb{N}^n_{2k}}. \end{cases} \tag{2.15}$$

It is the $k$th Lasserre's relaxation for the polynomial optimization

$$\begin{cases} \min & \xi^T [x]_m \\ \text{s.t} & e^T x - 1 = 0,\ x \geq 0, v^* - \mathcal{A}(x) \geq 0. \end{cases} \tag{2.16}$$

The feasible set of (2.16) is clearly compact. There exists a subset $\Theta \subseteq \mathbb{R}^{\mathbb{N}^n_m}$ of zero Lebesgue measure [130, §2.2], such that for all $\xi \in \mathbb{R}^{\mathbb{N}^n_m} \backslash \Theta$ the problem (2.16) has a unique optimizer, say, $u^*$. Hence, for almost all $\xi \in \mathbb{R}^{\mathbb{N}^n_m}$, $u^*$ is the unique optimizer. For notation convenience, denote by $\hat{y}^k$ the optimizer of (2.9) with the relaxation order $k$. Let $u^k = \big((\hat{y}^k)_{e_1}, \ldots, (\hat{y}^k)_{e_n}\big)$. By Corollary 3.5 of [131] or Theorem 3.3 of [102], the sequence $\{u^k\}_{k=m_0}^{\infty}$ must converge to $u^*$, the unique optimizer of (2.16). Since $\mathcal{A}(u^*) \leq v^* < 0$, we must have $\mathcal{A}(u^k) < 0$ when $k$ is sufficiently large. Moreover, the constraints $L^{(k)}_{x_i}[y] \succeq 0$ imply that $u^k \geq 0$, and $L^{(k)}_{e^T x - 1}[y] = 0$ implies that $e^T u^k = 1$. Therefore, $u^k \in \Delta$. □

In Step 1 of Algorithm 2.1, we need to test whether or not $v_k \geq 0$. When the absolute value of $v_k$ is big, this testing is easy. However, if its absolute value is very small,

then testing its sign might be difficult. Note that the semidefinite relaxation (2.8) is often solved numerically, i.e., $v_k$ is accurate up to a tiny round-off error. This difficulty is not because of theoretical properties of Algorithm 2.1, but due to round-off errors, which occur in all numerical methods. In practice, if $v_k$ is positive or close to zero (say, $v_k > -10^{-6}$), then it is reasonably well to claim the copositivity of $\mathcal{A}$.

## 2.3 Numerical examples

This section presents numerical experiments of applying Algorithm 2.1 to detect matrix and tensor copositivity. The computation is implemented in MATLAB R2016b, on a Lenovo Laptop with CPU@2.90GHz and RAM 16.0G. Algorithm 2.1 can be implemented by using the software `Gloptipoly 3` [66], which calls the semidefinite program solver `SeDuMi` [139]. For cleanness, we only display 4 decimal digits. The computational time is reported in seconds (s). Recall that $v_k$ is the optimal value of (2.8).

First, we consider some copositive matrices that are not a sum of psd and nonnegative matrices.

**Example 2.3.** *Consider the Horn's matrix [60]*

$$\begin{bmatrix} 1 & -1 & 1 & 1 & -1 \\ -1 & 1 & -1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & 1 & -1 \\ -1 & 1 & 1 & -1 & 1 \end{bmatrix}, \tag{2.17}$$

*the Hoffman-Pereira matrix [70]*

$$\begin{bmatrix} 1 & -1 & 1 & 0 & 0 & 1 & -1 \\ -1 & 1 & -1 & 1 & 0 & 0 & 1 \\ 1 & -1 & 1 & -1 & 1 & 0 & 0 \\ 0 & 1 & -1 & 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & -1 & 1 & -1 & 1 \\ 1 & 0 & 0 & 1 & -1 & 1 & -1 \\ -1 & 1 & 0 & 0 & 1 & -1 & 1 \end{bmatrix}, \tag{2.18}$$

*and the Hildebrand matrix [68]*

$$\begin{bmatrix} 1 & -\cos\psi_4 & \cos(\psi_4+\psi_5) & \cos(\psi_2+\psi_3) & -\cos\psi_3 \\ -\cos\psi_4 & 1 & -\cos\psi_5 & \cos(\psi_1+\psi_5) & \cos(\psi_3+\psi_4) \\ \cos(\psi_4+\psi_5) & -\cos\psi_5 & 1 & -\cos\psi_1 & \cos(\psi_1+\psi_2) \\ \cos(\psi_2+\psi_3) & \cos(\psi_1+\psi_5) & -\cos\psi_1 & 1 & -\cos\psi_2 \\ -\cos\psi_3 & \cos(\psi_3+\psi_4) & \cos(\psi_1+\psi_2) & -\cos\psi_2 & 1 \end{bmatrix}, \tag{2.19}$$

*where each* $\psi_i \geq 0$ *and* $\sum_{i=1}^5 \psi_i < \pi$. *Here, we choose the values*

$$\psi_1 = \psi_2 = \psi_3 = \psi_4 = \psi_5 = \pi/6.$$

*All these matrices are copositive but are not a sum of psd and nonnegative matrices. We apply Algorithm 2.1 to test its copositivity. The lower bounds* $v_k$ *and computational time are shown in Table 2.1. Their copositivity are all confirmed at* $k = 3$, *up to tiny round-off errors.*

Table 2.1: Computational results for matrices in Example 2.3

| | Horn | | Hoffman-Pereira | | Hildebrand | |
|---|---|---|---|---|---|---|
| $k$ | $v_k$ | time(s) | $v_k$ | time(s) | $v_k$ | time(s) |
| 1 | $-0.7889$ | 0.59 | $-0.4503$ | 0.58 | $-0.2218$ | 0.61 |
| 2 | $-0.0472$ | 0.35 | $-0.0250$ | 0.60 | $-0.0153$ | 0.32 |
| 3 | $-7.0 \times 10^{-8}$ | 1.68 | $-2.2 \times 10^{-7}$ | 24.85 | $-1.2 \times 10^{-8}$ | 1.11 |

Copositive matrices have applications in graph theory. Let $G = (V, E)$ be a graph, with $V$ the set of vertices and $E$ the set of edges. Its stability number $\alpha(G)$ is the maximum number of pairwise disjoint vertices. As shown in [36, 94], it holds that

$$\alpha(G)^{-1} = \min_{x \in \Delta} \quad x^T(A_G + I)x,$$

where $A_G$ is the adjacency matrix of $G$. To determine $\alpha(G)$, it is enough to compute the minimum value $v^*$ of (2.3) for the matrix $A := A_G + I$.

**Example 2.4.** *For each integer* $\ell > 0$, *construct a graph* $G_\ell$ *as in [47, §4.2.2], as follows. Let* $K_{\ell+1,\ell+1}$ *be the complete bipartite graph with the vertex set* $\{(-1, i), (1, i) : i = 0, 1, \ldots, \ell\}$. *Its edges are* $\big((-1, i), (1, j)\big)$, *for* $i, j = 0, 1, \ldots, \ell$. *For each* $i = 1, \ldots, \ell$, *add a vertex to the edge of the form* $\big((-1, i), (1, i)\big)$, *which we denote as* $(0, i)$; *then delete the old edge* $\big((-1, i), (1, i)\big)$

*from the graph and add two new ones* $\big((-1,i),(0,i)\big),\big((0,i),(1,i)\big)$. *The resulting graph is* $G_\ell$. *As mentioned in [47],* $\alpha(G_\ell) = \ell + 1$. *For the matrix* $A := A_G + I$, *the optimal value* $v^*$ *of (2.3) is* $\frac{1}{\ell+1}$. *We apply the semidefinite relaxation (2.8) to compute* $\alpha(G_\ell)^{-1}$. *For* $k = 2$, *the lower bound* $v_2$ *is quite accurate. The computational results are reported in Table 2.2.*

Table 2.2: Stability numbers for graphs $G_\ell$.

| $\ell$ | $n = \lvert G_\ell\rvert$ | $v_2$ | $\lvert v_2 - \frac{1}{\ell+1}\rvert$ | time(s) |
|---|---|---|---|---|
| 1 | 5 | 0.5000 | $9.2 \times 10^{-8}$ | 0.53 |
| 2 | 8 | 0.3333 | $1.3 \times 10^{-7}$ | 1.77 |
| 3 | 11 | 0.2500 | $1.5 \times 10^{-6}$ | 10.47 |
| 4 | 14 | 0.2000 | $2.4 \times 10^{-6}$ | 119.25 |

**Example 2.5.** *Consider three tensors* $\mathcal{A} \in \mathbb{S}^3(\mathbb{R}^3)$ *whose polynomials* $\mathcal{A}(x)$ *are respectively given as*

$$\begin{cases} \textit{Motzkin:} & \mathcal{A}(x) := x_1^2x_2 + x_1x_2^2 + x_3^3 - 3x_1x_2x_3, \\ \textit{Robinson:} & \mathcal{A}(x) := x_1^3 + x_2^3 + x_3^3 - x_1^2x_2 - x_1x_2^2 - x_1^2x_3 \\ & \qquad -x_1x_3^2 - x_2^2x_3 - x_2x_3^2 + 3x_1x_2x_3, \\ \textit{Choi-Lam:} & \mathcal{A}(x) := x_1^2x_2 + x_2^2x_3 + x_3^2x_1 - 3x_1x_2x_3. \end{cases} \tag{2.20}$$

*When each* $x_i$ *is replaced by* $x_i^2$, *the polynomials* $\mathcal{A}(x)$ *are respectively the Motzkin, Robinson and Choi-Lam polynomials (they are all nonnegative but not sum of squares [128]). Hence, these tensors are all copositive. We detect their copositivity by Algorithm 2.1. The computational results are shown in Table 2.3. For all these tensors, the copositivity is confirmed for* $k = 3$, *up to tiny round-off errors.*

Table 2.3: Computational results for tensors in Example 2.5

| $\mathcal{A}(x)$ | Motzkin | | Robinson | | Choi-Lam | |
|---|---|---|---|---|---|---|
| $k$ | $v_k$ | time(s) | $v_k$ | time(s) | $v_k$ | time(s) |
| 2 | $-0.0045$ | 0.78 | $-0.0208$ | 0.76 | $-0.0129$ | 0.77 |
| 3 | $-4.3 \times 10^{-8}$ | 0.45 | $-4.9 \times 10^{-8}$ | 0.23 | $-2.1 \times 10^{-8}$ | 0.37 |

Copositive tensors have applications in hypergraph theory [24]. A hypergraph $G = (V, E)$ has a vertex set $V = \{1, \ldots, n\}$ and an edge set $E$, such that each edge in $E$ is an

unordered tuple $(i_1, \ldots, i_\ell)$, with $i_1, \ldots, i_\ell \in V$. It is $m$-uniform if each edge is an unordered $m$-tuple $(i_1, \ldots, i_m)$, for distinct $i_1, \ldots, i_m$. Tensor copositivity can be used to bound coclique numbers for hypergraphs.

**Example 2.6.** *A coclique of a $m$-uniform hypergraph $G$ is a subset $K \subseteq V$ such that $K$ any subset of $K$ with cardinality $m$ does not give an edge of $G$. The largest cardinality of a coclique of $G$ is called the coclique number of $G$, which we denote by $\omega(G)$ [24]. Computing $\omega(G)$ is typically a challenging question. However, we can get a good upper bound for it by using tensor copositivity, as shown in [24]. The adjacency tensor of a $m$-uniform hypergraph $G = (V, E)$ is the symmetric tensor $\mathcal{C} \in \mathtt{S}^m(\mathbb{R}^n)$ such that*

$$\mathcal{C}_{i_1 \ldots i_m} = \begin{cases} 1/(m-1)! & (i_1, \ldots, i_m) \in E, \\ 0 & \text{otherwise.} \end{cases}$$

*Let $\mathcal{I}$ be the identity tensor (i.e., $\mathcal{I}_{i_1 \ldots i_m} = 1$ if $i_1 = \cdots = i_m = 1$ and $\mathcal{I}_{i_1 \ldots i_m} = 0$ otherwise), and let $\mathcal{E}$ be the tensor of all ones. It is shown in [24] that $\omega(G)^{m-1} \leq \rho$ for all $\rho$ such that $\rho(\mathcal{I} + \mathcal{C}) - \mathcal{E}$ is copositive. To get such smallest such $\rho$, we need to compute the largest $\gamma$ such that $(\mathcal{I} + \mathcal{C}) - \gamma\mathcal{E}$ is copositive. Such largest $\gamma$ equals the minimum value $v^*$ of (2.3) for the tensor $\mathcal{A} := \mathcal{I} + \mathcal{C}$. Let $v_k$ be the lower bound given by (2.8), then*

$$\omega(G) \leq (1/v^*)^{1/(m-1)} \leq (1/v_k)^{1/(m-1)}.$$

*Since $\omega(G)$ is an integer, the above implies that*

$$\omega(G) \leq \lfloor (1/v_k)^{1/(m-1)} \rfloor \tag{2.21}$$

*for all $k = m_0, m_0 + 1, \ldots$. We test the above bounds for a class of 3-uniform hypergraphs. Let $G_n = (V_n, E_n)$ be the hypergraph such that $V_n = \{1, \ldots, n\}$ and*

$$E_n = \Big\{(i, i+1, i+2)\Big\}_{i=1}^{n-2}.$$

*For these hypergraphs $G_n$, we solve the relaxation (2.8) for $k = 2$ and get $v_2$, which gives an upper bound for $\omega(G_n)$ by (2.21). The computational results are shown in Table 2.4. For $G_n$ in the table, the upper bounds given by (2.21) are tight. Indeed, for $n \geq 3$, one can verify that $\omega(G_n) = n - \lfloor n/3 \rfloor$. A coclique with maximum cardinality for $G_n$ ($n \geq 3$) is the subset*

$$\{1 \leq i \leq n : \quad \mod (i, 3) \neq 0\}.$$

Table 2.4: Coclqiue numbers of hypergraphs $G_n$

| $n$ | $\omega(G_n)$ | $(1/v_2)^{\frac{1}{m-1}}$ | $\lfloor (1/v_2)^{\frac{1}{m-1}} \rfloor$ | time(s) |
|---|---|---|---|---|
| 3 | 2 | 2.1381 | 2 | 0.12 |
| 4 | 3 | 3.0000 | 3 | 0.13 |
| 5 | 4 | 4.0000 | 4 | 0.16 |
| 6 | 4 | 4.1631 | 4 | 0.26 |
| 7 | 5 | 5.0000 | 5 | 0.37 |
| 8 | 6 | 6.0000 | 6 | 0.63 |
| 9 | 6 | 6.2140 | 6 | 1.41 |
| 10 | 7 | 7.0041 | 7 | 3.07 |
| 11 | 8 | 8.0000 | 8 | 5.39 |
| 12 | 8 | 8.2657 | 8 | 15.61 |
| 13 | 9 | 9.0370 | 9 | 31.57 |
| 14 | 10 | 10.0000 | 10 | 72.08 |
| 15 | 10 | 10.3254 | 10 | 213.15 |
| 16 | 11 | 11.0836 | 11 | 282.55 |
| 17 | 12 | 12.0000 | 12 | 487.77 |

**Example 2.7.** *For every tensor $\mathcal{A} \in \mathtt{S}^m(\mathbb{R}^n)$, there always exists a number $\gamma$ such that $\mathcal{A} + \gamma e^{\otimes m}$ is copositive. The smallest such $\gamma$, which we denote $\gamma_{\min}$, is the negative of the optimal value $v^*$ of (2.3) for the tensor $\mathcal{A}$. Clearly, $\mathcal{A}$ is copositive if and only if $\gamma_{\min} \leq 0$. This example explores the computational cost for computing $\gamma_{\min}$ for randomly generated cubic tensors $\mathcal{A} \in \mathtt{S}^3(\mathbb{R}^n)$ for various $n$. Here, we generate each $\mathcal{A}_{i_1 i_2 i_3}$ randomly, obeying normal distribution (this can be done as $\mathcal{A}_{i_1 i_2 i_3}$ = randn in* `MATLAB`*). For all generated instances, we got $-\gamma_{\min} = v_2$, i.e., the relaxation (2.8) is tight for the order $k = 2$ (this is because rank $M_2[\hat{y}] = 1$ for the optimal solution $\hat{y}$). The computational time is reported in Table 2.5.*

Table 2.5: Computational time (in seconds) for random cubic tensors

| $n$ | 9 | 11 | 13 | 15 | 17 | 19 |
|---|---|---|---|---|---|---|
| time(s) | 0.97 | 4.38 | 23.79 | 116.89 | 327.72 | 1109.81 |
| $n$ | 10 | 12 | 14 | 16 | 18 | 20 |
| time(s) | 1.82 | 10.93 | 50.44 | 229.32 | 633.40 | 2748.65 |

Generally, `SeDuMi` can solve SDPs accurately in the computational environment of double precision. However, if SDPs need to be solved highly accurately, we might use high-

accuracy solvers, e.g., `SDPA-GMP` [97]. Here, we report the experiment of using `SDPA-GMP` in Algorithm 2.1 to solve the SDP relaxations. The matrices/tensors in Examples 2.3 and 2.5 are tested. The results are shown in Table 2.6. For $k = 2$, `SDPA-GMP` gets similar lower bounds as `SeDuMi` does. However, for $k = 3$, `SDPA-GMP` obtains highly accurate lower bounds, compared to those in Tables 2.1,2.3. For the Hildebrand matrix, we got $v_3 \approx -1.2 \times 10^{-17}$; for other matrices/tensors, we got $v_3$ in the magnitude of order $10^{-30}$. We do not know why the accuracy for the Hildebrand matrix is relatively lower. A possible reason is that the Hildebrand matrix is given by cosine values, which might cause extra round-off errors in the computation. The comparison also shows that `SDPA-GMP` takes much more time for solving the SDPs. For Motzkin/Robinson/Choi-Lam tensors, the time is much less than that for others. This is because the sizes of their SDP relaxations are smaller. In some applications, if the copositivity testing needs to be highly accurate, a high-accuracy SDP solver like `SDPA-GMP` might be useful.

Table 2.6: Computational results by `SDPA-GMP`

| matrix/tensor | $k = 2$ | | $k = 3$ | |
|---|---|---|---|---|
| | $v_2$ | time(s) | $v_3$ | time(s) |
| Horn | -0.0472 | 7.28 | $-6.0 \times 10^{-29}$ | 303.33 |
| Hoffman-Pereira | -0.0250 | 76.83 | $-4.6 \times 10^{-29}$ | 12437.55 |
| Hildebrand | -0.0153 | 8.25 | $-1.2 \times 10^{-17}$ | 297.41 |
| Motzkin | -0.0448 | 0.34 | $-7.6 \times 10^{-31}$ | 4.94 |
| Robinson | -0.0208 | 0.37 | $-1.4 \times 10^{-30}$ | 3.90 |
| Choi-Lam | -0.0129 | 0.40 | $-7.7 \times 10^{-31}$ | 4.42 |

**Acknowledgement.** The Chapter 2, in full, is a reprint of the material as it appears in *SIAM Journal on Optimization* 2018 [119]. The dissertation author coauthored this paper with Nie, Jiawang and Zhang, Xinzhen.

# Chapter 3

# The Saddle Point Problem

## 3.1 Saddle point problems

Let $X \subseteq \mathbb{R}^n, Y \subseteq \mathbb{R}^m$ be two sets (for dimensions $n, m > 0$), and let $F(x, y)$ be a continuous function in $(x, y) \in X \times Y$. A pair $(x^*, y^*) \in X \times Y$ is said to be a saddle point of $F(x, y)$ over $X \times Y$ if

$$F(x^*, y) \leq F(x^*, y^*) \leq F(x, y^*) \quad \forall\, x \in X, \forall\, y \in Y. \tag{3.1}$$

The above implies that

$$F(x^*, y^*) = \min_{x \in X} F(x, y^*) \leq \max_{y \in Y} \min_{x \in X} F(x, y),$$

$$F(x^*, y^*) = \max_{y \in Y} F(x^*, y) \geq \min_{x \in X} \max_{y \in Y} F(x, y).$$

On the other hand, it always holds that

$$\max_{y \in Y} \min_{x \in X} F(x, y) \leq \min_{x \in X} \max_{y \in Y} F(x, y).$$

Therefore, if $(x^*, y^*)$ is a saddle point, then

$$\min_{x \in X} \max_{y \in Y} F(x, y) = F(x^*, y^*) = \max_{y \in Y} \min_{x \in X} F(x, y). \tag{3.2}$$

All saddle points share the same objective value, although there may exist different saddle points. The definition of saddle points in (3.1) requires the inequalities to hold for all points in the feasible sets $X, Y$. That is, when $y$ is fixed to $y^*$, $x^*$ is a global minimizer of $F(x, y^*)$ over $X$; when $x$ is fixed to $x^*$, $y^*$ is a global maximizer of $F(x^*, y)$ over $Y$. Certainly, $x^*$

must also be a local minimizer of $F(x, y^*)$ and $y^*$ must be a local maximizer of $F(x^*, y)$. So, the local optimality conditions can be applied at $(x^*, y^*)$. However, they are not sufficient for guaranteeing $(x^*, y^*)$ to be a saddle point, since (3.1) needs to be satisfied for all feasible points.

The saddle point problem of polynomials (SPPP) is for cases that $F(x, y)$ is a polynomial function in $(x, y)$ and $X, Y$ are semialgebraic sets, i.e., they are described by polynomial equalities and/or inequalities. The SPPP concerns the existence of saddle points and the computation of them if they exist. When $F$ is convex-concave in $(x, y)$ and $X, Y$ are nonempty compact convex sets, there exists a saddle point. We refer to [10, §2.6] for the classical theory for convex-concave type saddle point problems. The SPPPs have broad applications. They are of fundamental importance in duality theory for constrained optimization, min-max optimization and game theory [10, 80, 132].

For convex-concave type saddle point problems, most existing methods are based on gradients, subgradients, variational inequalities, or other related techniques. For these classical methods, we refer to the work by Chen, Lan and Ouyang [26], Cox, Juditsky and Nemirovski [30], He and Yuan [63], He and Monteiro [64], Korpelevich [77], Maistroskii [89], Monteiro and Svaiter [92], Nemirovski [99], Nedić and Ozdaglar [98], and Zabotin [146]. For more general cases of non convex-concave type saddle point problems (i.e., $F$ is not convex-concave, and/or one of the sets $X, Y$ is nonconvex), the computational task for solving SPPPs is much harder. A saddle point may, or may not, exist. There is very little work for solving non convex-concave saddle point problems [34, 121].

## 3.2 Optimality conditions

Throughout the paper, a property is said to hold *generically* in a space if it is true everywhere except a subset of zero Lebesgue measure. We refer to [62] for the notion of genericity in algebraic geometry. Assume $X, Y$ are basic closed semialgebraic sets that are given as

$$X = \{x \in \mathbb{R}^n \mid g_i(x) = 0\,(i \in \mathcal{E}_1^X),\ g_i(x) \geq 0\,(i \in \mathcal{E}_2^X)\}, \tag{3.3}$$

$$Y = \{y \in \mathbb{R}^m \mid h_j(y) = 0\,(i \in \mathcal{E}_1^Y),\ h_j(y) \geq 0\,(i \in \mathcal{E}_2^Y)\}. \tag{3.4}$$

Here, each $g_i$ is a polynomial in $x := (x_1, \ldots, x_n)$ and each $h_j$ is a polynomial in $y := (y_1, \ldots, y_m)$. The $\mathcal{E}_1^X, \mathcal{E}_2^X, \mathcal{E}_1^Y, \mathcal{E}_2^Y$ are disjoint labeling sets of finite cardinalities. To distinguish

equality and inequality constraints, denote the tuples

$$\boxed{\begin{array}{ll} g_{eq} := (g_i)_{i\in\mathcal{E}_1^X}, & h_{eq} := (h_j)_{j\in\mathcal{E}_1^Y}, \\ g_{in} := (g_i)_{i\in\mathcal{E}_2^X}, & h_{in} := (h_j)_{j\in\mathcal{E}_2^Y}. \end{array}} \tag{3.5}$$

When $\mathcal{E}_1^X = \emptyset$ (resp., $\mathcal{E}_2^X = \emptyset$), there is no equality (resp., inequality) constraint for $X$. The same holds for $Y$. For convenience, denote the labeling sets

$$\mathcal{E}^X := \mathcal{E}_1^X \cup \mathcal{E}_2^X, \quad \mathcal{E}^Y := \mathcal{E}_1^Y \cup \mathcal{E}_2^Y.$$

Suppose $(x^*, y^*)$ is a saddle point. Then, $x^*$ is a minimizer of

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} & F(x, y^*) \\ \text{subject to} & g_i(x) = 0\,(i \in \mathcal{E}_1^X), \\ & g_i(x) \geq 0\,(i \in \mathcal{E}_2^X), \end{cases} \tag{3.6}$$

and $y^*$ is a maximizer of

$$\begin{cases} \max\limits_{y\in\mathbb{R}^m} & F(x^*, y) \\ \text{subject to} & h_j(y) = 0\,(j \in \mathcal{E}_1^Y), \\ & h_j(y) \geq 0\,(j \in \mathcal{E}_2^Y). \end{cases} \tag{3.7}$$

Under the linear independence constraint qualification (LICQ), or other kinds of constraint qualifications (see [9, §3.3]), there exist Lagrange multipliers $\lambda_i, \mu_j$ such that

$$\nabla_x F(x^*, y^*) = \sum_{i\in\mathcal{E}^X} \lambda_i \nabla_x g_i(x^*), \quad 0 \leq \lambda_i \perp g_i(x^*) \geq 0\,(i \in \mathcal{E}_2^X), \tag{3.8}$$

$$\nabla_y F(x^*, y^*) = \sum_{j\in\mathcal{E}^Y} \mu_j \nabla_y h_j(y^*), \quad 0 \geq \mu_j \perp h_j(y^*) \geq 0\,(j \in \mathcal{E}_2^Y). \tag{3.9}$$

In the above, $a \perp b$ means the product $a \cdot b = 0$ and $\nabla_x F$ (resp., $\nabla_y F$) denotes the gradient of $F(x, y)$ with respect to $x$ (resp., $y$). When $g, h$ are nonsingular (see the below for the definition), we can get explicit expressions for $\lambda_i, \mu_j$ in terms of $x^*, y^*$ (see [108]). For convenience, write the labeling sets as

$$\mathcal{E}^X = \{1, \ldots, \ell_1\}, \quad \mathcal{E}^Y = \{1, \ldots, \ell_2\}.$$

Then, the constraining polynomial tuples can be written as

$$g \,=\, (g_1, \ldots, g_{\ell_1}), \quad h \,=\, (h_1, \ldots, h_{\ell_2}).$$

The Lagrange multipliers can be written as vectors

$$\lambda \,=\, (\lambda_1, \ldots, \lambda_{\ell_1}), \quad \mu \,=\, (\mu_1, \ldots, \mu_{\ell_2}).$$

Denote the matrices

$$G(x) \,:=\, \begin{bmatrix} \nabla_x g_1(x) & \nabla_x g_2(x) & \cdots & \nabla_x g_{\ell_1}(x) \\ g_1(x) & 0 & \cdots & 0 \\ 0 & g_2(x) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & g_{\ell_1}(x) \end{bmatrix}, \tag{3.10}$$

$$H(y) \,:=\, \begin{bmatrix} \nabla_y h_1(y) & \nabla_y h_2(y) & \cdots & \nabla_y h_{\ell_2}(y) \\ h_1(y) & 0 & \cdots & 0 \\ 0 & h_2(y) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{\ell_2}(y) \end{bmatrix}. \tag{3.11}$$

The tuple $g$ is said to be *nonsingular* if $\operatorname{rank} G(x) \,=\, \ell_1$ for all $x \,\in\, \mathbb{C}^n$. Similarly, $h$ is nonsingular if $\operatorname{rank} H(y) = \ell_2$ for all $y \in \mathbb{C}^m$. Note that if $g$ is nonsingular, then LICQ must hold at $x^*$. Similarly, the LICQ holds at $y^*$ if $h$ is nonsingular. When $g, h$ have generic coefficients (i.e., $g, h$ are generic), the tuples $g, h$ are nonsingular. The nonsingularity is a property that holds generically. We refer to the work [108] for more details.

We assume that the sets $X, Y$ are given as in (3.3)-(3.4) and the defining polynomial tuples $g, h$ are nonsingular, i.e., the matrices $G(x), H(y)$ have full column rank everywhere. Then, as shown in [108], there exist matrix polynomials $G_1(x), H_1(y)$ such that ($I_\ell$ denotes the $\ell \times \ell$ identity matrix)

$$G_1(x)G(x) \,=\, I_{\ell_1}, \quad H_1(y)H(y) \,=\, I_{\ell_2}. \tag{3.12}$$

When $g, h$ have generic coefficients, they are nonsingular. Clearly, the above and (3.8)-(3.9) imply that

$$\lambda_i = G_1(x^*)_{i,1:n} \nabla_x F(x^*, y^*), \quad \mu_j = H_1(y^*)_{j,1:m} \nabla_y F(x^*, y^*).$$

(For a matrix $A$, the notation $A_{i,1:n}$ denotes its $i$th row with column indices from 1 to $n$.) Denote the Lagrange polynomial tuples

$$\lambda(x, y) := G_1(x)_{:,1:n} \nabla_x F(x, y), \tag{3.13}$$

$$\mu(x,y) := H_1(y)_{:,1:m}\nabla_y F(x,y). \tag{3.14}$$

(The notation $A_{:,1:n}$ denotes the submatrix of $A$ consisting of its first $n$ columns.) At each saddle point $(x^*, y^*)$, the Lagrange multiplier vectors $\lambda, \mu$ in (3.8)-(3.9) can be expressed as

$$\lambda = \lambda(x^*, y^*), \quad \mu = \mu(x^*, y^*).$$

Therefore, $(x^*, y^*)$ is a solution to the polynomial system

$$\begin{cases} g_i(x) = 0\,(i \in \mathcal{E}_1^X),\ h_j(y) = 0\,(j \in \mathcal{E}_1^Y), \\ \nabla_x F(x,y) = \sum\limits_{i\in\mathcal{E}^X} \lambda_i(x,y)\nabla_x g_i(x), \\ \nabla_y F(x,y) = \sum\limits_{j\in\mathcal{E}^Y} \mu_j(x,y)\nabla_y h_j(y), \\ 0 \le \lambda_i(x,y) \perp g_i(x) \ge 0\,(i \in \mathcal{E}_2^X), \\ 0 \ge \mu_j(x,y) \perp h_j(y) \ge 0\,(j \in \mathcal{E}_2^Y). \end{cases} \tag{3.15}$$

However, not every solution $(x^*, y^*)$ to (3.15) is a saddle point. This is because $x^*$ might not be a minimizer of (3.6), and/or $y^*$ might not be a maximizer of (3.7).

## 3.3 An algorithm for solving SPPPs

Let $F, g, h$ be the polynomial tuples for the saddle point problem (3.1). Assume $g, h$ are nonsingular. So the Lagrange multiplier vectors $\lambda(x,y), \mu(x,y)$ can be expressed as in (3.13)-(3.14). We have seen that each saddle point $(x^*, y^*)$ must satisfy (3.15). This leads us to consider the optimization problem

$$\begin{cases} \min\limits_{x\in X, y\in Y} & F(x,y) \\ \text{subject to} & \nabla_x F(x,y) - \sum_{i\in\mathcal{E}^X} \lambda_i(x,y)\nabla_x g_i(x) = 0, \\ & \nabla_y F(x,y) - \sum_{j\in\mathcal{E}^Y} \mu_j(x,y)\nabla_y h_j(y) = 0, \\ & 0 \le \lambda_i(x,y) \perp g_i(x) \ge 0\,(i \in \mathcal{E}_2^X), \\ & 0 \ge \mu_j(x,y) \perp h_j(y) \ge 0 (j \in \mathcal{E}_2^Y), \end{cases} \tag{3.16}$$

where $\lambda_i(x,y)$ and $\mu_j(x,y)$ are Lagrange polynomials given as in (3.13)-(3.14). The saddle point problem (3.1) is not equivalent to (3.16). However, the optimization problem (3.16) can be used to get a candidate saddle point. Suppose $(x^*, y^*)$ is a minimizer of (3.16). If $x^*$ is a minimizer of $F(x, y^*)$ over $X$ and $y^*$ is a maximizer of $F(x^*, y)$ over $Y$, then $(x^*, y^*)$ is

a saddle point; otherwise, such $(x^*, y^*)$ is not a saddle point, i.e., there exists $u \in X$ and/or there exists $v \in Y$ such that

$$F(u, y^*) - F(x^*, y^*) < 0 \quad \text{and/or} \quad F(x^*, v) - F(x^*, y^*) > 0.$$

The points $u, v$ can be used to give new constraints

$$F(u, y) - F(x, y) \geq 0 \quad \text{and/or} \quad F(x, y) - F(x, v) \geq 0. \tag{3.17}$$

Every saddle point $(x, y)$ must satisfy (3.17), so (3.17) can be added to the optimization problem (3.16) without excluding any true saddle points. For generic polynomials $F, g, h$, the problem (3.16) has only finitely many feasible points (see Theorem 3.3). Therefore, by repeatedly adding new inequalities like (3.17), we can eventually get a saddle point or detect nonexistence of saddle points. This results in the following algorithm.

**Algorithm 3.1.** *(An algorithm for solving saddle point problems.)*

***Input:*** *The polynomials $F, g, h$ as in (3.1), (3.3), (3.4) and Lagrange multiplier expressions as in (3.13)-(3.14).*

***Step 0:*** *Let $K_1 = K_2 = \mathcal{S}_a := \emptyset$ be empty sets.*

***Step 1:*** *If the problem (3.16) is infeasible, then (3.1) does not have a saddle point and stop; otherwise, solve (3.16) for a set $K^0$ of minimizers. Let $k := 0$.*

***Step 2:*** *For each $(x^*, y^*) \in K^k$, do the following:*

*(a): (Lower level minimization) Solve the problem*

$$\begin{cases} \vartheta_1(y^*) := \min\limits_{x \in X} & F(x, y^*) \\ \quad \textit{subject to} & \nabla_x F(x, y^*) - \sum_{i \in \mathcal{E}^X} \lambda_i(x, y^*) \nabla_x g_i(x) = 0, \\ & 0 \leq \lambda_i(x, y^*) \perp g_i(x) \geq 0 \, (i \in \mathcal{E}_2^X), \end{cases} \tag{3.18}$$

*and get a set of minimizers $S_1(y^*)$. If $F(x^*, y^*) > \vartheta_1(y^*)$, update*

$$K_1 := K_1 \cup S_1(y^*).$$

*(b): (Lower level maximization) Solve the problem*

$$\begin{cases} \vartheta_2(x^*) := \max\limits_{y \in Y} & F(x^*, y) \\ \quad \textit{subject to} & \nabla_y F(x^*, y) - \sum_{j \in \mathcal{E}^Y} \mu_j(x^*, y) \nabla_y h_j(y) = 0, \\ & 0 \geq \mu_j(x^*, y) \perp h_j(y) \geq 0 (j \in \mathcal{E}_2^Y) \end{cases} \tag{3.19}$$

*and get a set of maximizers* $S_2(x^*)$. *If* $F(x^*, y^*) < \vartheta_2(x^*)$, *update*

$$K_2 := K_2 \cup S_2(x^*).$$

*(c): If* $\vartheta_1(y^*) = F(x^*, y^*) = \vartheta_2(x^*)$, *update:*

$$\mathcal{S}_a := \mathcal{S}_a \cup \{(x^*, y^*)\}.$$

***Step 3:*** *If* $\mathcal{S}_a \neq \emptyset$, *then each point in* $\mathcal{S}_a$ *is a saddle point and stop; otherwise go to Step 4.*

***Step 4:*** *(Upper level minimization) Solve the optimization problem*

$$\begin{cases} \min\limits_{x\in X, y\in Y} & F(x,y) \\ \text{subject to} & \nabla_x F(x,y) - \sum_{i\in\mathcal{E}^X} \lambda_i(x,y)\nabla_x g_i(x) = 0, \\ & \nabla_y F(x,y) - \sum_{j\in\mathcal{E}^Y} \mu_j(x,y)\nabla_y h_j(y) = 0, \\ & 0 \le \lambda_i(x,y) \perp g_i(x) \ge 0\,(i \in \mathcal{E}_2^X), \\ & 0 \ge \mu_j(x,y) \perp h_j(y) \ge 0 (j \in \mathcal{E}_2^Y), \\ & F(u,y) - F(x,y) \ge 0\,(u \in K_1), \\ & F(x,v) - F(x,y) \le 0\,(v \in K_2). \end{cases} \tag{3.20}$$

*If (3.20) is infeasible, then (3.1) has no saddle points and stop; otherwise, compute a set* $K^{k+1}$ *of optimizers for (3.20). Let* $k := k + 1$ *and go to Step 2.*

***Output:*** *If* $\mathcal{S}_a$ *is nonempty, every point in* $\mathcal{S}_a$ *is a saddle point; otherwise, output that there is no saddle point.*

For generic polynomials, the feasible set $\mathcal{K}_0$ of (3.16) and each $K^k$ in Algorithm 3.1 is finite. The convergence of Algorithm 3.1 is shown as follows.

**Theorem 3.2** ( [118])**.** *Let* $\mathcal{K}_0$ *be the feasible set of (3.16) and let* $\mathcal{S}_a$ *be the set of saddle points for (3.1). If the complement set of* $\mathcal{S}_a$ *in* $\mathcal{K}_0$ *(i.e., the set* $\mathcal{K}_0 \setminus \mathcal{S}_a$*) is finite, then Algorithm 3.1 must terminate after finitely many iterations. Moreover, if* $\mathcal{S}_a \neq \emptyset$, *then each* $(x^*, y^*) \in \mathcal{S}_a$ *is a saddle point; if* $\mathcal{S}_a = \emptyset$, *then there is no saddle point.*

*Proof.* At an iteration, if $\mathcal{S}_a \neq \emptyset$, then Algorithm 3.1 terminates. For each iteration with $\mathcal{S}_a = \emptyset$, each point $(x^*, y^*) \in K^k$ is not feasible for (3.20). When the $k$th iteration goes to the $(k+1)$th one, the nonempty sets

$$K^0, \quad K^1, \quad K^2, \quad K^3, \ldots, \quad K^k$$

are disjoint from each other. All the points in $K^i$ are not saddle points, so

$$\bigcup_{i=0}^{k} K^i \subseteq \mathcal{K}_0 \setminus \mathcal{S}_a.$$

Therefore, when the set $\mathcal{K}_0 \setminus \mathcal{S}_a$ is finite, Algorithm 3.1 must terminate after finitely many iterations.

When $\mathcal{S}_a \neq \emptyset$, each point $(x^*, y^*) \in \mathcal{S}_a$ is verified as a saddle point in Step 2. When $\mathcal{S}_a = \emptyset$, Algorithm 3.1 stops in Step 4 at some iteration, with the case that (3.20) is infeasible. Since every saddle point is feasible for both (3.16) and (3.20), there does not exist a saddle point if $\mathcal{S}_a = \emptyset$. □ □

The number of iterations required by Algorithm 3.1 to terminate is bounded above by the cardinality of the complement set $\mathcal{K}_0 \setminus \mathcal{S}_a$, which is always less than or equal to the cardinality $|\mathcal{K}_0|$ of the feasible set of (3.16). Generally, it is hard to count $|\mathcal{K}_0 \setminus \mathcal{S}_a|$ or $|\mathcal{K}_0|$ accurately. When the polynomials $F, g, h$ are generic, we can prove that the number of solutions for equality constraints in (3.16) is finite. For degrees $a_0, b_0 > 0$, denote the set product $\mathbb{C}[x, y]_{a_0,b_0} := \mathbb{C}[x]_{a_0} \cdot \mathbb{C}[y]_{b_0}$.

**Theorem 3.3** ( [118]). *Let $a_0, b_0$ and $a_i, b_j > 0$ be positive degrees, for $i \in \mathcal{E}^X$ and $j \in \mathcal{E}^Y$. If $F(x, y) \in \mathbb{C}[x, y]_{a_0,b_0}$, $g_i \in \mathbb{C}[x]_{a_i}$, $h_j \in \mathbb{C}[y]_{b_j}$ are generic polynomials, then the polynomial system*

$$\begin{cases} \nabla_x F(x,y) = \sum_{i\in\mathcal{E}^X} \lambda_i(x,y)\nabla_x g_i(x), \\ g_i(x) = 0\,(i \in \mathcal{E}_1^X),\ \lambda_i(x,y)g_i(x) = 0\,(i \in \mathcal{E}_2^X), \\ \nabla_y F(x,y) = \sum_{j\in\mathcal{E}^Y} \mu_j(x,y)\nabla_y h_j(y), \\ h_j(y) = 0\,(j \in \mathcal{E}_1^Y),\ \mu_j(x,y)h_j(y) = 0\,(j \in \mathcal{E}_2^Y) \end{cases} \tag{3.21}$$

*has only finitely many complex solutions in $\mathbb{C}^n \times \mathbb{C}^m$.*

The proof for Theorem 3.3 will be given in Section 3.5. One would like to know what is the number of complex solutions to the polynomial system (3.21) for generic polynomials $F, g, h$. That number is an upper bound for $|\mathcal{K}_0|$ and so is also an upper bound for the number of iterations required by Algorithm 3.1 to terminate. The following theorem gives an upper bound for $|\mathcal{K}_0|$.

**Theorem 3.4** ( [118]). *For the degrees $a_i, b_j$ as in Theorem 3.3, let*

$$M := \sum_{\substack{\{i_1,\ldots,i_{r_1}\}\subseteq[\ell_1],0\le r_1\le n \\ \{j_1,\ldots,j_{r_2}\}\subseteq[\ell_2],0\le r_2\le m}} a_{i_1}\cdots a_{i_{r_1}} b_{j_1}\cdots b_{j_{r_2}} \cdot s \tag{3.22}$$

*where in the above the number $s$ is given as*

$$s = \sum_{\substack{k_0+\cdots+k_{r_1+r_2}=n+m-r_1-r_2 \\ k_0,\ldots,k_{r_1+r_2}\in\mathbb{N}}} (a_0+b_0)^{k_0}(a_{i_1})^{k_1}\cdots(a_{i_{r_1}})^{k_{r_1}}(b_{j_1})^{k_{r_1+1}}\cdots(b_{j_{r_2}})^{k_{r_1+r_2}}.$$

*If $F(x,y)$, $g_i$, $h_j$ are generic, then (3.21) has at most $M$ complex solutions, and hence Algorithm 3.1 must terminate within $M$ iterations.*

The proof for Theorem 3.4 will be given in Section 3.5. We remark that the upper bound $M$ given in (3.22) is not sharp. In our computational practice, Algorithm 3.1 typically terminates after a few iterations. It is an interesting question to obtain accurate upper bounds for the number of iterations required by Algorithm 3.1 to terminate.

## 3.4 Solving optimization problems

We discuss how to solve the optimization problems that appear in Algorithm 3.1. Under some genericity assumptions on $F, g, h$, we show that their optimizers can be computed by solving Lasserre type semidefinite relaxations. Let $X, Y$ be feasible sets given as in (3.3)-(3.4). Assume $g, h$ are nonsingular, so $\lambda(x,y), \mu(x,y)$ can be expressed as in (3.13)-(3.14).

The optimization problem (3.16) is a special case of (3.20), with $K_1 = K_2 = \emptyset$. It suffices to discuss how to solve (3.20) with finite sets $K_1, K_2$. For convenience, we rewrite (3.20) explicitly as

$$\begin{cases} \min\limits_{(x,y)} & F(x,y) \\ \text{subject to} & \nabla_x F(x,y) - \sum_{i\in\mathcal{E}^X} \lambda_i(x,y)\nabla_x g_i(x) = 0, \\ & \nabla_y F(x,y) - \sum_{j\in\mathcal{E}^Y} \mu_j(x,y)\nabla_y h_j(y) = 0, \\ & g_i(x) = 0,\ h_j(y) = 0\ (i\in\mathcal{E}_1^X, j\in\mathcal{E}_1^Y), \\ & \lambda_i(x,y)g_i(x) = 0,\ \mu_j(x,y)h_j(y) = 0\ (i\in\mathcal{E}_2^X, j\in\mathcal{E}_2^Y), \\ & g_i(x)\geq 0,\ \lambda_i(x,y)\geq 0\ (i\in\mathcal{E}_2^X), \\ & h_j(y)\geq 0,\ -\mu_j(x,y)\geq 0\ (j\in\mathcal{E}_2^Y), \\ & F(u,y) - F(x,y)\geq 0\ (\forall\, u\in K_1), \\ & F(x,y) - F(x,v)\geq 0\ (\forall\, v\in K_2). \end{cases} \tag{3.23}$$

Recall that $\lambda_i(x,y)$, $\mu_j(x,y)$ are Lagrange polynomials as in (3.13)-(3.14). Denote by $\phi$ the

tuple of equality constraining polynomials

$$\phi := \Big\{\nabla_x F - \sum\nolimits_{i\in\mathcal{E}^X}\lambda_i(x,y)\nabla_x g_i\Big\} \cup \Big\{\nabla_y F - \sum\nolimits_{j\in\mathcal{E}^Y}\mu_j(x,y)\nabla_y h_j\Big\} \\ \cup \Big\{g_i, h_j\Big\}_{i\in\mathcal{E}_1^X, j\in\mathcal{E}_1^Y} \cup \Big\{\lambda_i(x,y)g_i,\ \mu_j(x,y)h_j\Big\}_{i\in\mathcal{E}_2^X, j\in\mathcal{E}_2^Y}, \quad (3.24)$$

and denote by $\psi$ the tuple of inequality constraining ones

$$\psi := \Big\{g_i,\ h_j,\ \lambda_i(x,y),\ -\mu_j(x,y)\Big\}_{i\in\mathcal{E}_2^X, j\in\mathcal{E}_2^Y} \cup \\ \Big\{F(u,y) - F(x,y),\ F(x,y) - F(x,v)\Big\}_{u\in K_1, v\in K_2}. \quad (3.25)$$

They are polynomials in $(x,y)$. Let

$$d_0 := \lceil \frac{1}{2}\max\{\deg F(x,y), \deg(\phi), \deg(\psi)\}\rceil. \quad (3.26)$$

Then, the optimization problem (3.23) can be simply written as

$$\begin{cases} f_* := \min & F(x,y) \\ \text{subject to} & \phi(x,y) = 0,\ \psi(x,y) \geq 0. \end{cases} \quad (3.27)$$

We apply Lasserre's hierarchy of semidefinite relaxations to solve (3.27). For integers $k = d_0, d_0+1, \cdots$, the $k$th order semidefinite relaxation is

$$\begin{cases} F_k := \min & \langle F, w\rangle \\ \text{subject to} & (w)_0 = 1,\ M_k(w) \succeq 0, \\ & L_\phi^{(k)}(w) = 0, L_\psi^{(k)}(w) \succeq 0,\ w \in \mathbb{R}^{\mathbb{N}_{2k}^{n+m}}. \end{cases} \quad (3.28)$$

The number $k$ is called a relaxation order.

**Algorithm 3.5.** *(An algorithm for solving the optimization (3.23).)*

***Input:*** *Polynomials $F, \phi, \psi$ as in (3.24)-(3.25).*

***Step 0:*** *Let $k := d_0$.*

***Step 1:*** *Solve the semidefinite relaxation (3.28).*

***Step 2:*** *If the relaxation (3.28) is infeasible, then (3.1) has no saddle points and stop; otherwise, solve it for a minimizer $w^*$. Let $t := d_0$.*

***Step 3*** *Check whether or not $w^*$ satisfies the rank condition*

$$rank\, M_t(w^*) \ = \ rank\, M_{t-d_0}(w^*). \quad (3.29)$$

***Step 4*** *If (3.29) holds, extract $r := \operatorname{rank} M_t(w^*)$ minimizers for (3.23) and stop.*

***Step 5*** *If $t < k$, let $t := t + 1$ and go to Step 3; otherwise, let $k := k + 1$ and go to Step 1.*

***Output:*** *Minimizers of the optimization problem (3.23) or a certificate for the infeasibility of (3.23).*

The conclusions in the Steps 2 and 3 are justified by the following Proposition 3.6. The rank condition (3.29) is called flat extension or flat truncation [32, 102]. It is a sufficient and also almost necessary criterion for checking convergence of Lasserre type relaxations [102]. When it is satisfied, the method in [67] can be applied to extract minimizers in Step 4. It was implemented in the software `GloptiPoly 3` [66].

**Proposition 3.6** ( [118])**.** *Suppose $g, h$ are nonsingular polynomial tuples. For the hierarchy of relaxations (3.28), we have the properties:*

*i) If (3.28) is infeasible for some $k$, then (3.23) is infeasible and (3.1) has no saddle points.*

*ii) If (3.28) has a minimizer $w^*$ satisfying (3.29), then $F_k = f_*$ and there are $r := \operatorname{rank} M_t(w^*)$ minimizers for (3.23).*

*Proof.* Since $g, h$ are nonsingular, every saddle point must be a critical point, and Lagrange multipliers can be expressed as in (3.13)-(3.14).

i) For each $(u, v)$ that is feasible for (3.23), $[(u, v)]_{2k}$ satisfies all the constraints of (3.28), for all $k$. Therefore, if (3.28) is infeasible for some $k$, then (3.23) is infeasible.

ii) The conclusion follows from the classical results in [32, 67, 83, 102]. □

The notation IQ is the sum of an ideal and a quadratic module. The polynomial tuples $\phi, \psi$ are from (3.24)-(3.25). Algorithm 3.5 is able to solve (3.23) successfully after finitely many iterations, under the following genericity conditions.

**Condition 3.7.** *The polynomial tuples $g, h$ are nonsingular and $F, g, h$ satisfy one (not necessarily all) of the following:*

*(1) $IQ(g_{eq}, g_{in}) + IQ(h_{eq}, h_{in})$ is archimedean;*

*(2) the equation $\phi(x, y) = 0$ has finitely many real solutions;*

*(3) $IQ(\phi, \psi)$ is archimedean.*

In the above, the item (1) is almost the same as that $X, Y$ are compact sets; the item (2) is the same as that (3.21) has only finitely many real solutions. Also note that the item (1) or (2) implies (3). In Theorem 3.3, we have shown that (3.21) has only finitely many complex solutions when $F, g, h$ are generic. Therefore, Condition 3.7 holds generically. Under Condition 3.7, Algorithm 3.5 can be shown to have finite convergence.

**Theorem 3.8** ( [118]). *Under Condition 3.7, we have that:*

*i) If the problem (3.23) is infeasible, then the semidefinite relaxation (3.28) must be infeasible for all $k$ big enough.*

*ii) Suppose (3.23) is feasible. If (3.23) has only finitely many minimizers and each of them is an isolated critical point (i.e., an isolated real solution of (3.21)), then, for all $k$ big enough, (3.28) has a minimizer and each minimizer must satisfy the rank condition (3.29).*

We would like to remark that when $F, g, h$ are generic, every minimizer of (3.23) is an isolated real solution of (3.21). This is because (3.21) has only finitely many complex solutions for generic $F, g, h$. Therefore, Algorithm 3.5 has finite convergence for generic cases.

For a given pair $(x^*, y^*)$ that is feasible for (3.16) or (3.20), we need to check whether or not $x^*$ is a minimizer of $F(x, y^*)$ over $X$. This requires us to solve the minimization problem

$$\begin{cases} \min\limits_{x \in \mathbb{R}^n} & F(x, y^*) \\ \text{subject to} & g_i(x) = 0 \, (i \in \mathcal{E}_1^X), \\ & g_i(x) \geq 0 \, (i \in \mathcal{E}_2^X). \end{cases} \tag{3.30}$$

When $g$ is nonsingular, if it has a minimizer, the optimization (3.30) is equivalent to (by adding necessary optimality conditions)

$$\begin{cases} \min\limits_{x \in \mathbb{R}^n} & F(x, y^*) \\ \text{subject to} & \nabla_x F(x, y^*) - \sum\limits_{i \in \mathcal{E}^X} \lambda_i(x, y^*) \nabla_x g_i(x) = 0, \\ & g_i(x) = 0 \, (i \in \mathcal{E}_1^X), \, \lambda_i(x, y^*) g_i(x) = 0 \, (i \in \mathcal{E}_2^X), \\ & g_i(x) \geq 0, \, \lambda_i(x, y^*) \geq 0 \, (i \in \mathcal{E}_2^X). \end{cases} \tag{3.31}$$

Denote the tuple of equality constraining polynomials

$$\phi_{y^*} := \Big\{\nabla_x F(x,y^*) - \sum\nolimits_{i\in\mathcal{E}^X}\lambda_i(x,y^*)\nabla_x g_i\Big\} \cup \big\{g_i\big\}_{i\in\mathcal{E}_1^X} \cup \big\{\lambda_i(x,y^*)\cdot g_i\big\}_{i\in\mathcal{E}_2^X}, \quad (3.32)$$

and denote the tuple of inequality ones

$$\psi_{y^*} := \Big\{g_i,\ \lambda_i(x,y^*)\Big\}_{i\in\mathcal{E}_2^X}. \quad (3.33)$$

They are polynomials in $x$ but not in $y$, depending on the value of $y^*$. Let

$$d_1 := \lceil \frac{1}{2}\max\{\deg F(x,y^*), \deg(\phi_{y^*}), \deg(\psi_{y^*})\}\rceil. \quad (3.34)$$

We can rewrite (3.31) equivalently as

$$\begin{cases} \min\limits_{x\in\mathbb{R}^n} & F(x,y^*) \\ \text{subject to} & \phi_{y^*}(x)=0,\ \psi_{y^*}(x)\geq 0. \end{cases} \quad (3.35)$$

Lasserre's hierarchy of semidefinite relaxations for solving (3.35) is

$$\begin{cases} \min\limits_{z} & \langle F(x,y^*), z\rangle \\ \text{subject to} & (z)_0 = 1,\ M_k(z)\succeq 0, \\ & L^{(k)}_{\phi_{y^*}}(z)=0, L^{(k)}_{\psi_{y^*}}(z)\succeq 0,\ z\in\mathbb{R}^{\mathbb{N}^n_{2k}}, \end{cases} \quad (3.36)$$

for relaxation orders $k = d_1, d_1+1, \ldots$. Since $(x^*, y^*)$ is a feasible pair for (3.16) or (3.20), the problems (3.30) and (3.35) are also feasible, hence (3.36) is also feasible. A standard algorithm for solving (3.35) is as follows.

**Algorithm 3.9.** *(An algorithm for solving the problem (3.35).)*

***Input:*** *The point $y^*$ and polynomials $F(x,y^*), \phi_{y^*}, \psi_{y^*}$ as in (3.32)-(3.33).*

***Step 0:*** *Let $k := d_1$.*

***Step 1:*** *Solve the semidefinite relaxation (3.36) for a minimizer $z^*$. Let $t := d_1$.*

***Step 2:*** *Check whether or not $z^*$ satisfies the rank condition*

$$rank\, M_t(z^*) = rank\, M_{t-d_1}(z^*). \quad (3.37)$$

***Step 3:*** *If (3.37) holds, extract $r := rank\, M_t(z^*)$ minimizers and stop.*

***Step 4:*** *If $t < k$, let $t := t+1$ and go to Step 3; otherwise, let $k := k+1$ and go to Step 1.*

***Output:*** *Minimizers of the optimization problem (3.35).*

Similar conclusions as in Proposition 3.6 hold for Algorithm 3.9. For cleanness of the paper, we do not state them again. The method in [67] can be applied to extract minimizers in the Step 3. Moreover, Algorithm 3.9 also terminates within finitely many iterations, under some genericity conditions.

**Condition 3.10.** *The polynomial tuple $g$ is nonsingular and the point $y^*$ satisfies one (not necessarily all) of the following:*

*(1) $IQ(g_{eq}, g_{in})$ is archimedean;*

*(2) the equation $\phi_{y^*}(x) = 0$ has finitely many real solutions;*

*(3) $IQ(\phi_{y^*}, \psi_{y^*})$ is archimedean.*

Since $(x^*, y^*)$ is feasible for (3.16) or (3.20), Condition 3.7 implies Condition 3.10, which also holds generically. The finite convergence of Algorithm 3.9 is summarized as follows.

**Theorem 3.11** ( [118])**.** *Assume the optimization problem (3.30) has a minimizer and Condition 3.10 holds. If each minimizer of (3.30) is an isolated critical point, then, for all $k$ big enough, (3.36) has a minimizer and each of them must satisfy (3.37).*

The proof of Theorem 3.11 will be given in Section 3.5. We would like to remark that every minimizer of (3.35) is an isolated critical point of (3.30), when $F, g, h$ are generic. This is implied by Theorem 3.3.

For a given pair $(x^*, y^*)$ that is feasible for (3.16) or (3.20), we need to check whether or not $y^*$ is a maximizer of $F(x^*, y)$ over $Y$. This requires us to solve the maximization problem

$$\begin{cases} \max\limits_{y \in \mathbb{R}^m} & F(x^*, y) \\ \text{subject to} & h_j(y) = 0\,(j \in \mathcal{E}_1^Y),\ h_j(y) \geq 0\,(j \in \mathcal{E}_2^Y). \end{cases} \tag{3.38}$$

When $h$ is nonsingular, if it has a minimizer, the optimization (3.38) is equivalent to (by adding necessary optimality conditions) the problem

$$\begin{cases} \max\limits_{y\in\mathbb{R}^m} & F(x^*,y) \\ \text{subject to} & \nabla_y F(x^*,y) - \sum_{j\in\mathcal{E}^Y} \mu_j(x^*,y)\nabla_y h_j(y) = 0, \\ & h_j(y) = 0\,(j\in\mathcal{E}_1^Y),\ \mu_j(x^*,y)\cdot h_j(y) = 0\,(j\in\mathcal{E}_2^Y), \\ & h_j(y)\geq 0,\ -\mu_j(x^*,y)\geq 0\,(j\in\mathcal{E}_2^Y). \end{cases} \tag{3.39}$$

Denote the tuple of equality constraining polynomials

$$\phi_{x^*} := \Big\{\nabla_y F(x^*,y) - \sum_{j\in\mathcal{E}^Y}\mu_j(x^*,y)\nabla_y h_j\Big\} \\ \cup \big\{h_j\big\}_{j\in\mathcal{E}_1^Y} \cup \big\{\mu_j(x^*,y)h_j\big\}_{j\in\mathcal{E}_2^Y}, \tag{3.40}$$

and denote the tuple of inequality ones

$$\psi_{x^*} := \Big\{h_j,\ -\mu_j(x^*,y)\Big\}_{j\in\mathcal{E}_2^Y}. \tag{3.41}$$

They are polynomials in $y$ but not in $x$, depending on the value of $x^*$. Let

$$d_2 := \lceil \frac{1}{2}\max\{\deg F(x^*,y), \deg(\phi_{x^*}), \deg(\psi_{x^*})\}\rceil. \tag{3.42}$$

Hence, (3.39) can be simply expressed as

$$\begin{cases} \max\limits_{y\in\mathbb{R}^m} & F(x^*,y) \\ \text{subject to} & \phi_{x^*}(y) = 0,\ \psi_{x^*}(y)\geq 0. \end{cases} \tag{3.43}$$

Lasserre's hierarchy of semidefinite relaxations for solving (3.43) is

$$\begin{cases} \max\limits_{z} & \langle F(x^*,y), z\rangle \\ \text{subject to} & (z)_0 = 1,\ M_k(z)\succeq 0, \\ & L_{\phi_{x^*}}^{(k)}(z) = 0, L_{\psi_{x^*}}^{(k)}(z)\succeq 0, \\ & z\in\mathbb{R}^{\mathbb{N}^m_{2k}}, \end{cases} \tag{3.44}$$

for relaxation orders $k = d_2, d_2+1, \cdots$. Since $(x^*, y^*)$ is feasible for (3.16) or (3.20), the problems (3.38) and (3.43) must also be feasible. Hence, the relaxation (3.44) is always feasible. Similarly, an algorithm for solving (3.43) is as follows.

**Algorithm 3.12.** *(An algorithm for solving the problem (3.43).)*

***Input:*** *The point $x^*$ and polynomials $F(x^*, y), \phi_{x^*}, \psi_{x^*}$ as in (3.40)-(3.41).*

***Step 0:*** *Let $k := d_2$.*

***Step 1:*** *Solve the semidefinite relaxation (3.44) for a maximizer $z^*$. Let $t := d_2$.*

***Step 2:*** *Check whether or not $z^*$ satisfies the rank condition*

$$rank\, M_t(z^*) \;=\; rank\, M_{t-d_2}(z^*). \tag{3.45}$$

***Step 3:*** *If (3.45) holds, extract $r := rank\, M_t(z^*)$ maximizers for (3.43) and stop.*

***Step 4:*** *If $t < k$, let $t := t + 1$ and go to Step 3; otherwise, let $k := k + 1$ and go to Step 1.*

***Output:*** *Maximizers of the optimization problem (3.43).*

The same kind of conclusions like in Proposition 3.6 hold for Algorithm 3.12. The method in [67] can be applied to extract maximizers in Step 3. We can show that it must also terminate within finitely many iterations, under some genericity conditions.

**Condition 3.13.** *The polynomial tuple $h$ is nonsingular and the point $x^*$ satisfies one (not necessarily all) of the following:*

*(1) $IQ(h_{eq}, h_{in})$ is archimedean;*

*(2) the equation $\phi_{x^*}(y) = 0$ has finitely many real solutions;*

*(3) $IQ(\phi_{x^*}, \psi_{x^*})$ is archimedean.*

By the same argument as for Condition 3.10, we can also see that Condition 3.13 holds generically. Similarly, Algorithm 3.12 also terminates within finitely many iterations under some genericity conditions.

**Theorem 3.14** ( [118])**.** *Assume that (3.38) has a maximizer and Condition 3.13 holds. If each maximizer of (3.38) is an isolated critical point, then, for all $k$ big enough, (3.44) has a maximizer and each of them must satisfy (3.45).*

The proof of Theorem 3.14 will be given in Section 3.5. Similarly, when $F, g, h$ are generic, each maximizer of (3.38) is an isolated critical point of (3.38).

## 3.5 Some proofs

This section gives the proofs for some theorems in the previous sections.

*Proof of Theorem 3.3.* Under the genericity assumption, the tuples $g, h$ are nonsingular, so the Lagrange multipliers in (3.8)-(3.9) can be expressed as in (3.13)-(3.14). Hence, (3.21) is equivalent to the polynomial system in $(x, y, \lambda, \mu)$:

$$\begin{cases} \nabla_x F(x,y) = \sum_{i\in\mathcal{E}^X} \lambda_i \nabla_x g_i(x), \\ \nabla_y F(x,y) = \sum_{j\in\mathcal{E}^Y} \mu_j \nabla_y h_j(y), \\ g_i(x) = 0\,(i \in \mathcal{E}_1^X),\ \lambda_i g_i(x) = 0\,(i \in \mathcal{E}_2^X), \\ h_j(y) = 0\,(j \in \mathcal{E}_1^Y),\ \mu_j h_j(y) = 0\,(j \in \mathcal{E}_2^Y). \end{cases} \tag{3.46}$$

Due to the complementarity conditions, $g_i(x) = 0$ or $\lambda_i = 0$ for each $i \in \mathcal{E}_2^X$, and $h_j(x) = 0$ or $\mu_j = 0$ for each $j \in \mathcal{E}_2^Y$. Note that if $g_i(x) \neq 0$ then $\lambda_i = 0$ and if $h_j(x) \neq 0$ then $\mu_j = 0$. Since $\mathcal{E}_2^X, \mathcal{E}_2^Y$ are finite labeling sets, there are only finitely many cases of $g_i(x) = 0$ or $g_i(x) \neq 0$, $h_j(x) = 0$ or $h_j(x) \neq 0$. We prove the conclusion is true for every case. Moreover, if $g_i(x) = 0$ for $i \in \mathcal{E}_2^X$, then the inequality $g_i(x) \geq 0$ can be counted as an equality constraint. The same is true for $h_j(x) = 0$ with $j \in \mathcal{E}_2^Y$. Therefore, we only need to prove the conclusion is true for the case that has only equality constraints. Without loss of generality, assume $\mathcal{E}_2^X = \mathcal{E}_2^Y = \emptyset$ and write the labeling sets as

$$\mathcal{E}_1^X = \{1, \ldots, \ell_1\}, \qquad \mathcal{E}_1^Y = \{1, \ldots, \ell_2\}.$$

When all $g_i$ are generic polynomials, the equations $g_i(x) = 0$ $(i \in \mathcal{E}_1^X)$ have no solutions if $\ell_1 > n$. Similarly, the equations $h_j(x) = 0$ $(j \in \mathcal{E}_1^Y)$ have no solutions if $\ell_2 > m$ and all $h_j$ are generic. Therefore, we only consider the case that $\ell_1 \leq n$ and $\ell_2 \leq m$. When $F, g, h$ are generic, we show that (3.46) cannot have infinitely many solutions. The system (3.46) is the same as

$$\begin{cases} \nabla_x F(x,y) = \sum_{i=1}^{\ell_1} \lambda_i \nabla_x g_i(x),\ g_1(x) = \cdots = g_{\ell_1}(x) = 0, \\ \nabla_y F(x,y) = \sum_{j=1}^{\ell_2} \mu_j \nabla_y h_j(y),\ h_1(y) = \cdots = h_{\ell_2}(y) = 0. \end{cases} \tag{3.47}$$

Let $\tilde{x} = (x_0, x_1, \ldots, x_n)$ and $\tilde{y} = (y_0, y_1, \ldots, y_m)$. We denote the homogenization of $g_i(x)$ (resp., $h_j(y)$) by $\tilde{g}_i(\tilde{x})$ (resp., $\tilde{h}_j(\tilde{y})$). Let $\mathbb{P}^n$ denote the $n$-dimensional complex projective space. Consider the projective variety

$$\mathcal{U} := \big\{(\tilde{x}, \tilde{y}) \in \mathbb{P}^n \times \mathbb{P}^m : \tilde{g}_i(\tilde{x}) = 0\,(i \in \mathcal{E}^X),\ \tilde{h}_j(\tilde{y}) = 0\,(j \in \mathcal{E}^Y)\big\}.$$

It is smooth, by Bertini's theorem [62], under the genericity assumption on $g_i, h_j$. Denote the bi-homogenization of $F(x, y)$

$$\tilde{F}(\tilde{x}, \tilde{y}) := x_0^{a_0} y_0^{b_0} \tilde{F}(x/x_0, y/y_0).$$

When $F(x, y)$ is generic, the projective variety

$$\mathcal{V} := \mathcal{U} \cap \{\tilde{F}(\tilde{x}, \tilde{y}) = 0\}$$

is also smooth. One can directly verify that (for homogeneous polynomials)

$$x^T \nabla_x \tilde{F}(\tilde{x}, \tilde{y}) + x_0 \partial_{x_0} \tilde{F}(\tilde{x}, \tilde{y}) = a_0 \tilde{F}(\tilde{x}, \tilde{y}),$$

$$x^T \nabla_x \tilde{g}_i(\tilde{x}) + x_0 \partial_{x_0} \tilde{g}_i(\tilde{x}) = a_i \tilde{g}_i(\tilde{x}),$$

$$y^T \nabla_y \tilde{F}(\tilde{x}, \tilde{y}) + y_0 \partial_{y_0} \tilde{F}(\tilde{x}, \tilde{y}) = b_0 \tilde{F}(\tilde{x}, \tilde{y}),$$

$$y^T \nabla_y \tilde{h}_j(\tilde{y}) + y_0 \partial_{y_0} \tilde{h}_j(\tilde{y}) = b_i \tilde{h}_j(\tilde{y}).$$

(They are called Euler's identities.) Consider the determinantal variety

$$W := \left\{ (x, y) \in \mathbb{C}^n \times \mathbb{C}^m \;\middle|\; \begin{array}{l} \operatorname{rank} \begin{bmatrix} \nabla_x F(x,y) & \nabla_x g_1(x) & \cdots & \nabla_x g_{\ell_1}(x) \end{bmatrix} \le \ell_1 \\ \operatorname{rank} \begin{bmatrix} \nabla_y F(x,y) & \nabla_y h_1(y) & \cdots & \nabla_y h_{\ell_2}(y) \end{bmatrix} \le \ell_2 \end{array} \right\}.$$

Its homogenization is

$$\widetilde{W} := \left\{ (\tilde{x}, \tilde{y}) \in \mathbb{P}^n \times \mathbb{P}^m \;\middle|\; \begin{array}{l} \operatorname{rank} \begin{bmatrix} \nabla_x \tilde{F}(\tilde{x},\tilde{y}) & \nabla_x \tilde{g}_1(\tilde{x}) & \cdots & \nabla_x \tilde{g}_{\ell_1}(\tilde{x}) \end{bmatrix} \le \ell_1 \\ \operatorname{rank} \begin{bmatrix} \nabla_y \tilde{F}(\tilde{x},\tilde{y}) & \nabla_y \tilde{h}_1(\tilde{y}) & \cdots & \nabla_y \tilde{h}_{\ell_2}(\tilde{y}) \end{bmatrix} \le \ell_2 \end{array} \right\}.$$

The projectivization of (3.47) is the intersection

$$\widetilde{W} \cap \mathcal{U}.$$

If (3.21) has infinitely many complex solutions, so does (3.47). Then, $\widetilde{W} \cap \mathcal{U}$ must intersect the hypersurface $\{\tilde{F}(\tilde{x}, \tilde{y}) = 0\}$. This means that there exists $(\bar{x}, \bar{y}) \in \mathcal{V}$ such that

$$\nabla_x \tilde{F}(\bar{x}, \bar{y}) = \sum_{i=1}^{\ell_1} \lambda_i \nabla_x \tilde{g}_i(\bar{x}), \quad \nabla_y \tilde{F}(\bar{x}, \bar{y}) = \sum_{j=1}^{\ell_2} \mu_j \nabla_y \tilde{h}_j(\bar{y}),$$

for some $\lambda_i, \mu_j$. Also note $\tilde{g}_i(\bar{x}) = \tilde{h}_j(\bar{y}) = \tilde{F}(\bar{x}, \bar{y}) = 0$. Write

$$\bar{x} = (\bar{x}_0, \bar{x}_1, \ldots, \bar{x}_n), \quad \bar{y} = (\bar{y}_0, \bar{y}_1, \ldots, \bar{y}_m).$$

- If $\bar{x}_0 \neq 0$ and $\bar{y}_0 \neq 0$, by Euler's identities, we can further get

$$\partial_{x_0}\tilde{F}(\bar{x},\bar{y}) = \sum_{i=1}^{\ell_1} \lambda_i \partial_{x_0}\tilde{g}_i(\bar{x}), \quad \partial_{y_0}\tilde{F}(\bar{x},\bar{y}) = \sum_{j=1}^{\ell_2} \mu_j \partial_{y_0}\tilde{h}_j(\bar{y}).$$

  This implies that $\mathcal{V}$ is singular, which is a contradiction.

- If $x_0 = 0$ but $y_0 \neq 0$, by Euler's identities, we can also get

$$\partial_{y_0}\tilde{F}(\bar{x},\bar{y}) = \sum_{j=1}^{\ell_2} \mu_j \partial_{y_0}\tilde{h}_j(\bar{y}).$$

  This means the linear section $\mathcal{V} \cap \{x_0 = 0\}$ is singular, which is a contradiction again, by the genericity assumption on $F, g, h$.

- If $x_0 \neq 0$ but $y_0 = 0$, then we can have

$$\partial_{x_0}\tilde{F}(\bar{x},\bar{y}) = \sum_{i=1}^{\ell_1} \lambda_i \partial_{x_0}\tilde{g}_i(\bar{x}).$$

  So the linear section $\mathcal{V} \cap \{y_0 = 0\}$ is singular, which is again a contradiction.

- If $x_0 = y_0 = 0$, then $\mathcal{V} \cap \{x_0 = 0, y_0 = 0\}$ is singular. It is also a contradiction, under the genericity assumption on $F, g, h$.

For every case, we obtain a contradiction. Therefore, the polynomial system (3.21) must have only finitely many complex solutions, when $F, g, h$ are generic. □ □

*Proof of Theorem 3.4.* Each solution of (3.21) is a critical point of $F(x, y)$ over the set $X \times Y$. We count the number of critical points by enumerating all possibilities of active constraints. For an active labeling set $\{i_1, \ldots, i_{r_1}\} \subseteq [\ell_1]$ (for $X$) and an active labeling set $\{j_1, \ldots, j_{r_2}\} \subseteq [\ell_2]$ (for $Y$), an upper bound for the number is critical points is $a_{i_1} \cdots a_{i_{r_1}} b_{j_1} \cdots b_{j_{r_2}} \cdot s$, which is given by Theorem 2.2 of [116]. Summing this upper bound for all possible active constraints, we eventually get the bound $M$. Since $\mathcal{K}_0$ is a subset of (3.21), Algorithm 3.1 must terminate within $M$ iterations, for generic polynomials. □ □

*Proof of Theorem 3.8.* In Condition 3.7, the item (1) or (2) implies (3). Note that the dual optimization problem of (3.28) is

$$\begin{cases} \max & \gamma \\ \text{subject to} & F - \gamma \in \mathrm{IQ}(\phi, \psi)_{2k}. \end{cases} \tag{3.48}$$

i) When (3.23) is infeasible, the set $\{\phi(x,y)=0,\ \psi(x,y)\geq 0\}$ is empty. Since $\mathrm{IQ}(\phi,\psi)$ is archimedean, by the classical Positivstellensatz [13] and Putinar's Positivstellensatz [124], we have $-1\in\mathrm{IQ}(\phi,\psi)$. So, $-1\in\mathrm{IQ}(\phi,\psi)_{2k}$ for all such $k$ big enough. Hence, (3.48) is unbounded from above for all big $k$. By weak duality, we know (3.28) must be infeasible.

ii) When (3.23) is feasible, every feasible point is a critical point. By Lemma 3.3 of [41], $F(x,y)$ achieves finitely many values on $\phi(x,y)=0$, say,

$$c_1<c_2<\cdots<c_N.$$

Recall that $f_*$ is the minimum value of (3.27). So, $f_*$ is one of the $c_i$, say, $c_\ell=f_*$. Since (3.23) has only finitely many minimizers, we can list them as the set

$$O:=\{(u_1,v_1),\ldots,(u_B,v_B)\}.$$

If $(x,y)$ is a feasible point of (3.23), then either $F(x,y)=c_k$ with $k>\ell$, or $(x,y)$ is one of $(u_1,v_1),\ldots,(u_B,v_B)$. Define the polynomial

$$P(x,y):=\Big(\prod_{i=\ell+1}^{N}(F(x,y)-c_i)^2\Big)\cdot\left(\prod_{(u_j,v_j)\in O}\Big(\|x-u_j\|^2+\|y-v_j\|^2\Big)\right).$$

We partition the set $\{\phi(x,y)=0\}$ into four disjoint ones:

$$\begin{aligned}
U_1&:=\{\phi(x,y)=0, c_1\leq F(x,y)\leq c_{\ell-1}\},\\
U_2&:=\{\phi(x,y)=0, F(x,y)=c_\ell, (x,y)\notin O\},\\
U_3&:=\{\phi(x,y)=0, F(x,y)=c_\ell, (x,y)\in O\},\\
U_4&:=\{\phi(x,y)=0, c_{\ell+1}\leq F(x,y)\leq c_N\}.
\end{aligned}$$

Note that $U_3$ is the set of minimizers for (3.27).

- For all $(x,y)\in U_1$ and $i=\ell+1,\ldots,N$,

  $$(F(x,y)-c_i)^2\geq(c_{\ell-1}-c_{\ell+1})^2.$$

  The set $U_1$ is closed and each $(u_j,v_j)\notin U_1$. The distance from $(u_j,v_j)$ to $U_1$ is positive. Hence, there exists $\epsilon_1>0$ such that $P(x,y)>\epsilon_1$ for all $(x,y)\in U_1$.

- For all $(x,y)\in U_2$, $(F(x,y)-c_i)^2=(c_\ell-c_i)^2$. For each $(u_j,v_j)\in O$, its distance to $U_2$ is positive. This is because each $(u_i,v_i)\in O$ is an isolated real critical point. So, there exists $\epsilon_2>0$ such that $P(x,y)>\epsilon_2$ for all $(x,y)\in U_2$.

Denote the new polynomial

$$q(x,y) := \min(\epsilon_1, \epsilon_2) - P(x,y).$$

On the set $\{\phi(x,y) = 0\}$, the inequality $q(x,y) \geq 0$ implies $(x,y) \in U_3 \cup U_4$. Therefore, (3.23) is equivalent to the optimization problem

$$\begin{cases} \min\limits_{x,y} & F(x,y) \\ \text{subject to} & \phi(x,y) = 0,\ q(x,y) \geq 0. \end{cases} \tag{3.49}$$

Note that $q(x,y) > 0$ on the feasible set of (3.23). $\big($This is because if $(x,y)$ is a feasible point of (3.23), then $F(x,y) \geq f_* = c_\ell$, so $(x,y) \notin U_1$. If $F(x,y) = c_\ell$, then $(x,y) \in O$ and $P(x,y) = 0$, so $q(x,y) = \min(\epsilon_1, \epsilon_2) > 0$. If $F(x,y) > c_\ell$, then $P(x,y) = 0$ and we also have $q(x,y) = \min(\epsilon_1, \epsilon_2) > 0.\big)$ By Condition 3.7 and Putinar's Positivstellensatz, it holds that $q \in \mathrm{IQ}(\phi, \psi)$. Now, we consider the hierarchy of Lasserre's relaxations for solving (3.49):

$$\begin{cases} f_k' := \min & \langle F, w \rangle \\ \text{subject to} & (w)_0 = 1, M_k(w) \succeq 0, \\ & L_\phi^{(k)}(w) = 0,\ L_q^{(k)}(w) \succeq 0. \end{cases} \tag{3.50}$$

Its dual optimization problem is

$$\begin{cases} f_k := \max & \gamma \\ \text{subject to} & F - \gamma \in \mathrm{IQ}(\phi, q)_{2k}. \end{cases} \tag{3.51}$$

**Claim:** For all $k$ big enough, it holds that $f_k = f_k' = f_*$.

*Proof.* The possible objective values of (3.49) are $c_\ell, \ldots, c_N$. Let $p_1, \ldots, p_N$ be real univariate polynomials such that $p_i(c_j) = 0$ for $i \neq j$ and $p_i(c_j) = 1$ for $i = j$. Let

$$s_i := (c_i - f_*)\big(p_i(F)\big)^2, \quad i = \ell, \ldots, N.$$

Then $s := s_\ell + \cdots + s_N \in \Sigma[x]_{2k_1}$ for some order $k_1 > 0$. Let

$$\hat{F} := F - f_* - s.$$

Note that $\hat{F}(x) \equiv 0$ on the set

$$\mathcal{K}_2 := \{\phi(x,y) = 0,\ q(x,y) \geq 0\}.$$

It has a single inequality. By the Positivstellensatz [13, Corollary 4.1.8], there exist $0 < t \in \mathbb{N}$ and $Q = b_0 + qb_1$ ($b_0, b_1 \in \Sigma[x]$) such that $\hat{F}^{2t} + Q \in \text{Ideal}(\phi)$. Note that $Q \in \text{Qmod}(q)$. For all $\epsilon > 0$ and $\tau > 0$, we have $\hat{F} + \epsilon = \phi_\epsilon + \theta_\epsilon$ where

$$\phi_\epsilon = -\tau\epsilon^{1-2t}\big(\hat{F}^{2t} + Q\big),$$

$$\theta_\epsilon = \epsilon\Big(1 + \hat{F}/\epsilon + \tau(\hat{F}/\epsilon)^{2t}\Big) + \tau\epsilon^{1-2t}Q.$$

By Lemma 2.1 of [114], when $\tau \geq \frac{1}{2t}$, there exists $k_2$ such that, for all $\epsilon > 0$,

$$\phi_\epsilon \in \text{Ideal}(\phi)_{2k_2}, \quad \theta_\epsilon \in \text{Qmod}(q)_{2k_2}.$$

Hence, we can get

$$F - (f_* - \epsilon) = \phi_\epsilon + \sigma_\epsilon,$$

where $\sigma_\epsilon = \theta_\epsilon + s \in \text{Qmod}(q)_{2k_2}$ for all $\epsilon > 0$. For all $\epsilon > 0$, $\gamma = f_* - \epsilon$ is feasible in (3.51) for the order $k_2$, so $f_{k_2} \geq f_*$. Because $f_k \leq f_{k+1} \leq \cdots \leq f_*$, we have $f_k = f_k' = f_*$ for all $k \geq k_2$. □ □

Because $q \in \text{Qmod}(\psi)$, each $w$, which is feasible for (3.28), is also feasible for (3.50). This can be implied by [102, Lemma 2.5]. So, when $k$ is big, each $w$ is also a minimizer of (3.50). The problem (3.49) also has only finitely many minimizers. By Theorem 2.6 of [102], the condition (3.29) must be satisfied for some $t \in [d_0, k]$, when $k$ is big enough. □ □

*Proof of Theorem 3.11.* The proof is the same as the one for Theorem 3.8. This is because the Lasserre's relaxations (3.36) are constructed by using optimality conditions of (3.30), which is the same as for Theorem 3.8. In other words, Theorem 3.11 can be thought of a special version of Theorem 3.8 with $K_1 = K_2 = \emptyset$, without variable $y$. The assumptions are the same. Therefore, the same proof can be used. □ □

*Proof of Theorem 3.14.* The proof is the same as the one for Theorem 3.11. □ □

## 3.6 Numerical examples

This section presents numerical examples of applying Algorithm 3.1 to solve saddle point problems. The computation is implemented in MATLAB R2012a, on a Lenovo Laptop with CPU@2.90GHz and RAM 16.0G. The Lasserre type moment semidefinite relaxations

are solved by the software `GloptiPoly 3` [66], which calls the semidefinite program solver `SeDuMi` [139]. For cleanness, only four decimal digits are displayed for computational results.

In prior existing references, there are very few examples of non convex-concave type SPPPs. We construct various examples, with different types of functions and constraints. When $g, h$ are nonsingular tuples, the Lagrange multipliers $\lambda(x,y), \mu(x,y)$ can be expressed by polynomials as in (3.13)-(3.14). Here we give some expressions for $\lambda(x,y)$ that will be frequently used in the examples. The expressions are similar for $\mu(x,y)$. Let $F(x,y)$ be the objective.

- For the simplex $\Delta_n = \{x \in \mathbb{R}^n : e^T x = 1, x \geq 0\}$, $g = (e^T x - 1, x_1, \ldots, x_n)$ and we have
$$\lambda(x,y) \;=\; (x^T \nabla_x F, F_{x_1} - x^T \nabla_x F, \ldots, F_{x_n} - x^T \nabla_x F). \tag{3.52}$$

- For the hypercube set $[-1,1]^n$, $g = (1 - x_1^2, \ldots, 1 - x_n^2)$ and
$$\lambda(x,y) \;=\; -\frac{1}{2}(x_1 F_{x_1}, \ldots, x_n F_{x_n}). \tag{3.53}$$

- For the box constraint $[0,1]^n$, $g = (x_1, \ldots, x_n, 1 - x_1, \ldots, 1 - x_n)$ and
$$\lambda(x,y) \;=\; ((1 - x_1)F_{x_1}, \ldots, (1 - x_n)F_{x_n}, -x_1 F_{x_1}, \ldots, -x_n F_{x_n}). \tag{3.54}$$

- For the ball $B_n(0,1) = \{x \in \mathbb{R}^n : \|x\| \leq 1\}$ or sphere $\mathbb{S}^{n-1} = \{x \in \mathbb{R}^n : \|x\| = 1\}$, $g = 1 - x^T x$ and we have
$$\lambda(x,y) \;=\; -\frac{1}{2} x^T \nabla_x F. \tag{3.55}$$

- For the nonnegative orthant $\mathbb{R}^n_+$, $g = (x_1, \ldots, x_n)$ and we have
$$\lambda(x,y) \;=\; (F_{x_1}, \ldots, F_{x_n}). \tag{3.56}$$

We refer to [108] for more details about Lagrange multiplier expressions.

**Example 3.15.** *Consider the simplex feasible sets $X = \Delta_n$, $Y = \Delta_m$. The Lagrange multipliers can be expressed as in (3.52).*
*(i) Let $n = m = 3$ and*

$$F(x,y) \;=\; x_1x_2 + x_2x_3 + x_3y_1 + x_1y_3 + y_1y_2 + y_2y_3.$$

*This function is neither convex in $x$ nor concave in $y$. After 1 iteration by Algorithm 3.1, we got the saddle point:*

$$x^* = (0.0000,\ 1.0000,\ 0.0000), \quad y^* = (0.2500,\ 0.5000,\ 0.2500).$$

*It took about 2 seconds.*
*(ii) Let $n = m = 3$ and*

$$F(x,y) := x_1x_2y_1y_2 + x_2x_3y_2y_3 + x_3x_1y_3y_1 - x_1^2y_3^2 - x_2^2y_1^2 - x_3^2y_2^2.$$

*This function is neither convex in $x$ nor concave in $y$. After 4 iterations by Algorithm 3.1, we got that there is no saddle point. It took about 32 seconds.*

**Example 3.16.** *Consider the box constraints $X = [0,1]^n$ and $Y = [0,1]^m$. The Lagrange multipliers can be expressed as in (3.54).*
*(i) Consider $n = m = 2$ and*

$$F(x,y) := (x_1 + x_2 + y_1 + y_2 + 1)^2 - 4(x_1x_2 + x_2y_1 + y_1y_2 + y_2 + x_1).$$

*This function is convex in $x$ but not concave in $y$. After 2 iterations by Algorithm 3.1, we got the saddle point*

$$x^* = (0.3249, 0.3249), \quad y^* = (1.0000, 0.0000).$$

*It took about 3.7 seconds.*
*(ii) Let $n = m = 3$ and*

$$F(x,y) := \sum\nolimits_{i=1}^{n}(x_i + y_i) + \sum\nolimits_{i<j}(x_i^2y_j^2 - y_i^2x_j^2).$$

*This function is neither convex in $x$ nor concave in $y$. After 3 iterations by Algorithm 3.1, we got that there is no saddle point. It took about 12.8 seconds.*

**Example 3.17.** *Consider the cube constraints $X = Y = [-1,1]^3$. The Lagrange multipliers can be expressed as in (3.53).*
*(i) Consider the function*

$$F(x,y) := \sum\nolimits_{i=1}^{3}(x_i + y_i) - \prod\nolimits_{i=1}^{3}(x_i - y_i).$$

*This function is neither convex in $x$ nor concave in $y$. After 1 iteration by Algorithm 3.1, we got 3 saddle points:*

$$x^* = (-1.0000, -1.0000, 1.0000), \quad y^* = (1.0000, 1.0000, 1.0000),$$

$$x^* = (-1.0000, 1.0000, -1.0000), \quad y^* = (1.0000, 1.0000, 1.0000),$$

$$x^* = (1.0000, -1.0000, -1.0000), \quad y^* = (1.0000, 1.0000, 1.0000).$$

*It took about* 75 *seconds.*

*(ii) Consider the function*

$$F(x, y) := y^T y - x^T x + \sum\nolimits_{1 \le i < j \le 3} (x_i y_j - x_j y_i).$$

*This function is neither convex in* $x$ *nor concave in* $y$*. After* 4 *iterations by Algorithm 3.1, we got the saddle point*

$$x^* = (-1.0000, 1.0000, -1.0000), \quad y^* = (-1.0000, 1.0000, -1.0000).$$

*It took about* 6 *seconds.*

**Example 3.18.** *Consider the sphere constraints* $X = \mathbb{S}^2$ *and* $Y = \mathbb{S}^2$*. They are not convex. The Lagrange multipliers can be expressed as in (3.55).*

*(i) Let* $F(x, y)$ *be the function*

$$x_1^3 + x_2^3 + x_3^3 + y_1^3 + y_2^3 + y_3^3 + 2(x_1 x_2 y_1 y_2 + x_1 x_3 y_1 y_3 + x_2 x_3 y_2 y_3).$$

*After* 2 *iterations by Algorithm 3.1, we got* 9 *saddle points* $(-e_i, e_j)$*, with* $i, j = 1, 2, 3$*. It took about* 64 *seconds.*

*(ii) Let* $F(x, y)$ *be the function*

$$x_1^2 y_1^2 \ + \ x_2^2 y_2^2 \ + \ x_3^2 y_3^2 \ + \ x_1^2 y_2 y_3 \ + \ x_2^2 y_1 y_3 \ + \ x_3^2 y_1 y_2 \ + \ y_1^2 x_2 x_3 \ + \ y_2^2 x_1 x_3 \ + \ y_3^2 x_1 x_2.$$

*After* 4 *iterations by Algorithm 3.1, we got that there is no saddle point. It took about* 127 *seconds.*

**Example 3.19.** *Let* $X = Y = B_3(0, 1)$ *be the ball constraints and*

$$F(x, y) := x_1^2 y_1 + 2x_2^2 y_2 + 3x_3^2 y_3 - x_1 - x_2 - x_3.$$

*The Lagrange multipliers can be expressed as in (3.55). The function* $F$ *is not convex in* $x$ *but is concave in* $y$*. After* 1 *iteration by Algorithm 3.1, we got the saddle point:*

$$x^* = (0.7264, 0.4576, 0.3492), \quad y^* = (0.6883, 0.5463, 0.4772).$$

*It took about* 3.3 *seconds.*

**Example 3.20.** *Consider the function*

$$F(x,y) := x_1^2y_2y_3 + y_1^2x_2x_3 + x_2^2y_1y_3 + y_2^2x_1x_3 + x_3^2y_1y_2 + y_3^2x_1x_2$$

*and the sets*

$$X := \{x \in \mathbb{R}^3 : x^Tx - 1 = 0, x \geq 0\}, \quad Y := \{y \in \mathbb{R}^3 : y^Ty - 1 = 0, y \geq 0\}.$$

*They are nonnegative portions of spheres. The feasible sets $X, Y$ are non-convex. The Lagrange multipliers are expressed as*

$$\lambda(x,y) = (\frac{1}{2}x^T\nabla_x F, F_{x_1} - x_1x^T\nabla_x F, F_{x_2} - x_2x^T\nabla_x F, F_{x_3} - x_3x^T\nabla_x F),$$

$$\mu(x,y) = (\frac{1}{2}y^T\nabla_y F, F_{y_1} - y_1y^T\nabla_y F, F_{y_2} - y_2y^T\nabla_y F, F_{y_3} - y_3y^T\nabla_y F).$$

*After* 3 *iterations by Algorithm 3.1, we got that there is no saddle point. It took about* 37.3 *seconds.*

**Example 3.21.** *Let $X = Y = \mathbb{R}^4_+$ be the nonnegative orthant and $F(x,y)$ be*

$$\begin{aligned} &y_1(x_2+x_3+x_4-1)^2 + y_2(x_1+x_3+x_4-2)^2 + y_3(x_1+x_2+x_4-3)^2 \\ &\quad - y_4(x_1+x_2+x_3-4)^2 - \Big(x_1(y_2+y_3+y_4-1)^2 + x_2(y_1+y_3+y_4-2)^2 \\ &\qquad - x_3(y_1+y_2+y_4-3)^2 + x_4(y_1+y_2+y_3-4)^2\Big). \end{aligned}$$

*The Lagrange multipliers can be expressed as in (3.56). The function $F$ is neither convex in $x$ nor concave in $y$. After* 1 *iteration by Algorithm 3.1, we got the saddle point*

$$x^* = (1.5075, 0.5337, 0.0000, 0.5018), \quad y^* = (2.4143, 1.1463, 0.0000, 0.0000).$$

*It took about* 4.8 *seconds.*

**Example 3.22.** *Let $X = Y = \mathbb{R}^3$ be the entire space, i.e., there are no constraints. There are no needs for Lagrange multiplier expressions. Consider the function*

$$F(x,y) = \sum_{i=1}^{3}(x_i^4 - y_i^4 + x_i + y_i) + \sum_{i\neq j} x_i^3y_j^3.$$

*It is neither convex in $x$ nor concave in $y$. After* 1 *iteration by Algorithm 3.1, we got the saddle point*

$$x^* = -(0.6981, 0.6981, 0.6981), \quad y^* = (0.4979, 0.4979, 0.4979).$$

*It took about* 113 *seconds.*

**Example 3.23.** *Consider the sets and the function*

$$X := \{x \in \mathbb{R}^3 : x_1 \geq 0, x_1x_2 \geq 1, x_2x_3 \geq 1\},$$

$$Y := \{y \in \mathbb{R}^3 : y_1 \geq 0, y_1y_2 \geq 1, y_2y_3 \geq 1\},$$

$$F(x,y) := x_1^3y_1 + x_2^3y_2 + x_3^3y_3 - 3x_1x_2x_3 - y_1^2 - 2y_2^2 - 3y_3^2.$$

*The function $F(x,y)$ is not convex in $x$ but is concave in $y$. The Lagrange multipliers can be expressed as*

$$\lambda_1 = (1 - x_1x_2)F_{x_1}, \quad \lambda_2 = x_1F_{x_1}, \quad \lambda_3 = -x_1F_{x_1} + x_2F_{x_2}.$$

*The same expressions are for $\mu_j(x,y)$. After 9 iterations by Algorithm 3.1, we get the saddle point:*

$$x^* = (1.2599, 1.2181, 1.3032), \quad y^* = (1.0000, 1.1067, 0.9036).$$

*It took about 64 seconds.*

**Example 3.24.** *We consider the saddle point problem arising from zero sum games with two players. Suppose $x \in \mathbb{R}^n$ is the strategy for the first player and $y \in \mathbb{R}^m$ is the strategy for the second one. The usual constraints for strategies are given by simplices, which represent probability measures on finite sets. So we consider feasible sets $X = \Delta_n$, $Y = \Delta_m$. Suppose the profit function of the first player is*

$$f_1(x,y) = x^TA_1x + y^TA_2y + x^TBy,$$

*for matrices $A_1 \in \mathbb{R}^{n\times n}$, $A_2 \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{n\times m}$. For the zero sum game, the profit function for the second player is $f_2(x,y) := -f_1(x,y)$. Each player wants to maximize the profit, for the given strategy of the other player. The Nash equilibrium is a point $(x^*, y^*)$ such that the maximum of $f_1(x, y^*)$ over $\Delta_n$ is achieved at $x^*$, while the maximum of $f_2(x^*, y)$ over $\Delta_m$ is achieved at $y^*$. This is equivalent to that $(x^*, y^*)$ is a saddle point of the function $F := -f_1(x,y)$ over $X, Y$. For instance, we consider the matrices*

$$A_1 = \begin{pmatrix} -4 & 4 & 0 & 3 & -4 \\ 3 & 4 & 3 & -4 & -5 \\ -3 & 0 & -2 & 0 & 4 \\ -4 & -4 & -1 & 3 & -5 \\ 4 & 1 & -3 & 0 & -5 \end{pmatrix}, \quad A_2 = \begin{pmatrix} -4 & 4 & 1 & 0 & 1 \\ -2 & -4 & 2 & -3 & 1 \\ -3 & 1 & 1 & 4 & 4 \\ 3 & -4 & 0 & 1 & -2 \\ -1 & -3 & -1 & 3 & -2 \end{pmatrix},$$

$$B = \begin{pmatrix} -2 & -4 & -2 & -5 & 3 \\ 0 & 0 & 2 & 4 & 2 \\ 0 & -4 & -1 & -5 & 3 \\ 1 & -3 & -4 & 0 & -3 \\ 3 & -1 & -5 & 4 & -4 \end{pmatrix}.$$

*The resulting saddle point problem is of the non convex-concave type. After* 2 *iterations by Algorithm 3.1, we get two Nash equilibria*

$$x^* = (0, 1, 0, 0, 0), \quad y^* = (1, 0, 0, 0, 0),$$

$$x^* = (0, 1, 0, 0, 0), \quad y^* = (0, 1, 0, 0, 0).$$

*It took about* 7 *seconds.*

**Example 3.25.** *Consider the portfolio optimization problem [61, 147]*

$$\min_{x \in X} \quad -\mu^T x + x^T Q x,$$

*where $Q$ is a covariance matrix and $\mu$ is the estimation of some parameters. There often exists a perturbation $(\delta\mu, \delta Q)$ for $(\mu, Q)$. This results in two types of robust optimization problems*

$$\min_{x \in X} \quad \max_{(\delta\mu, \delta Q) \in Y} \quad -(\mu + \delta\mu)^T x + x^T (Q + \delta Q) x,$$
$$\max_{(\delta\mu, \delta Q) \in Y} \quad \min_{x \in X} \quad -(\mu + \delta\mu)^T x + x^T (Q + \delta Q) x.$$

*We look for $x^*$ and $(\delta\mu^*, \delta Q^*)$ that can solve the above two robust optimization problems simultaneously. This is equivalent to the saddle point problem with $F = -(\mu + \delta\mu)^T x + x^T (Q + \delta Q) x$. For instance, consider the case that*

$$Q = \begin{pmatrix} 5 & -4 & -2 \\ -4 & 13 & 10 \\ -2 & 10 & 8 \end{pmatrix}, \ \mu = \begin{pmatrix} 0 \\ -1 \\ 3 \end{pmatrix},$$

*with the feasible sets*

$$X := \{x \in \mathbb{R}^3 \mid -0.5 \leq x_i \leq 0.5, \, i = 1, \ldots, n\},$$

$$Y := \left\{ (\delta\mu, \delta Q) \in \mathbb{R}^3 \times \mathcal{S}\mathbb{R}^{3\times 3} \middle| \begin{array}{c} -0.1 \leq (\delta\mu)_k, \, (\delta Q)_{ij} \leq 0.1, \\ 1 \leq k \leq 3, \, 1 \leq i, j \leq 3 \end{array} \right\}.$$

*In the above, $\mathcal{S}\mathbb{R}^{3\times 3}$ denotes the space of real symmetric 3-by-3 matrices. The Lagrange multipliers can be similarly expressed as in (3.54). After 1 iteration by Algorithm 3.1, we got the saddle point*

$$x^* = \begin{pmatrix} -0.1289 \\ -0.4506 \\ 0.5000 \end{pmatrix}, \delta Q^* = \begin{pmatrix} 0.1 & 0.1 & -0.1 \\ 0.1 & 0.1 & -0.1 \\ -0.1 & -0.1 & 0.1 \end{pmatrix}, \delta\mu^* = \begin{pmatrix} 0.1 \\ 0.1 \\ -0.1 \end{pmatrix}.$$

*It took about 32 seconds. The above two min-max and max-min optimization problems are solved simultaneously by them.*

**Acknowledgement.** The Chapter 3, in full, is a reprint of the material as it appears in *Foundations of Computational Mathematics* 2021 [118]. The dissertation author coauthored this paper with Nie, Jiawang and Zhou, Guangming.

# Chapter 4

# Hermitian Tensors

## 4.1 Hermitian decompositions

Recall that a tensor $\mathcal{H} \in \mathbb{C}^{n_1 \times \cdots \times n_m \times n_1 \times \cdots \times n_m}$ is called *Hermitian* if

$$\mathcal{H}_{i_1 \ldots i_m j_1 \ldots j_m} = \overline{\mathcal{H}_{j_1 \ldots j_m i_1 \ldots i_m}}.$$

The notion $\mathbb{C}^{[n_1,\ldots,n_m]}$ denotes the set of all Hermitian tensors in $\mathbb{C}^{n_1 \times \cdots \times n_m \times n_1 \times \cdots \times n_m}$. For vectors $v_i \in \mathbb{C}^{n_i}$, $i = 1, \ldots, m$, we denote

$$[v_1, v_2, \ldots, v_m]_{\otimes h} := v_1 \otimes v_2 \cdots \otimes v_m \otimes \overline{v_1} \otimes \overline{v_2} \cdots \otimes \overline{v_m}. \tag{4.1}$$

Every rank-1 Hermitian tensor must be in the form of $\lambda \cdot [v_1, v_2, \ldots, v_m]_{\otimes h}$, for a real scalar $\lambda \in \mathbb{R}$. For every $\mathcal{H} \in \mathbb{C}^{[n_1,\ldots,n_m]}$, there exists a *Hermitian decomposition* [101]

$$\mathcal{H} = \sum\nolimits_{i=1}^{r} \lambda_i \, [u_i^1, \ldots, u_i^m]_{\otimes_h}. \tag{4.2}$$

where $u_i^j \in \mathbb{C}^{n_j}$ and real scalars $\lambda_i \in \mathbb{R}$. The smallest $r$ in (4.2) is called the *Hermitian rank* of $\mathcal{H}$, for which we denote hrank($\mathcal{H}$). When $r$ is minimum, (4.2) is called a *Hermitian rank decomposition* for $\mathcal{H}$. Hermitian decompositions can be equivalently expressed by conjugate polynomials. For complex vector variables $x_k \in \mathbb{C}^{n_k}$, $k = 1, \ldots, m$, denote $x := (x_1, \ldots, x_m)$. The inner product

$$\mathcal{H}(x, \overline{x}) := \langle \mathcal{H}, [x_1, \ldots, x_m]_{\otimes h} \rangle$$

is a conjugate symmetric polynomial in $x$, i.e., $\mathcal{H}(x, \overline{x}) = \overline{\mathcal{H}(x, \overline{x})}$. It only achieves real values [73, 101]. The decomposition $\mathcal{H} = \sum_{i=1}^{r} \lambda_i [u_i^1, \ldots, u_i^m]_{\otimes h}$ is equivalent to the polynomial decomposition

$$\mathcal{H}(x, \overline{x}) = \sum\nolimits_{i=1}^{r} \lambda_i |(u_i^1)^* x_1|^2 \cdots |(u_i^m)^* x_m|^2. \tag{4.3}$$

Therefore, a Hermitian decomposition of $\mathcal{H}$ can be equivalently expressed as a real linear combination of conjugate squares like $|(u_i^1)^* x_1|^2 \cdots |(u_i^m)^* x_m|^2$.

For square matrices $Q_k \in \mathbb{C}^{n_k \times n_k}$, $k = 1, \ldots, m$, we define the *multilinear congruent transformation* for $\mathcal{A} \in \mathbb{C}^{[n_1,\ldots,n_m]}$ such that

$$(Q_1, \ldots, Q_m) \times_{cong} \mathcal{A} := (Q_1, \ldots, Q_m, \overline{Q_1}, \ldots, \overline{Q_m}) \times \mathcal{A}. \tag{4.4}$$

If each $Q_k$ is unitary, then $\mathcal{B} := (Q_1, \ldots, Q_m) \times_{cong} \mathcal{A}$ is said to be a *unitary congruent transformation* of $\mathcal{A}$ and $\mathcal{B}$ is said to be *unitarily congruent* to $\mathcal{A}$. It holds that

$$(Q_1^*, \ldots, Q_m^*) \times_{cong} \Big( (Q_1, \ldots, Q_m) \times_{cong} \mathcal{A} \Big) = \mathcal{A}.$$

If each $Q_k$ is real and orthogonal, the tensor $\mathcal{B}$ is said to be *orthogonally congruent* to $\mathcal{A}$. Unitary and orthogonal congruent transformations preserve norms of Hermitian tensors [101].

Hermitian tensors have important applications in quantum physics [101]. An $m$-partite pure state $|\psi\rangle$ of a quantum system can be represented by a tensor in $\mathbb{C}^{n_1 \times \cdots \times n_m}$. The complex conjugate of $|\psi\rangle$ represents another pure state $\langle\psi|$. The conjugate product $|\psi\rangle\langle\psi|$ represents a $2m$-partite pure state in the Hermitian tensor space $\mathbb{C}^{[n_1,\ldots,n_m]}$. A mixed quantum state can be represented by a Hermitian tensor. The state is called unentangled (or separable) if it can be expressed as a sum of rank-1 pure state products like $|\psi\rangle\langle\psi|$; otherwise, the state is called entangled (or not separable). Equivalently, a mixed state $\rho \in \mathbb{C}^{[n_1,\ldots,n_m]}$ is unentangled if and only if

$$\rho = \sum_{i=1}^{k} |\psi_i\rangle\langle\psi_i|$$

for some rank-1 pure sates $|\psi_i\rangle$. Mathematically, the above is equivalent to the Hermitian decomposition

$$\rho = \sum_{i=1}^{k} (u_i^1 \otimes \cdots \otimes u_i^m) \otimes \overline{(u_i^1 \otimes \cdots \otimes u_i^m)} = \sum_{i=1}^{k} [u_i^1, \ldots, u_i^m]_{\otimes h},$$

for complex vectors $u_i^1 \in \mathbb{C}^{n_1}, \ldots, u_i^m \in \mathbb{C}^{n_m}$. Hermitian tensors, which can be decomposed as above, are called separable tensors. Hermitian tensors representing mixed states are also called density matrices. In view of algebra, Hermitian tensors can also be regarded as real valued complex conjugate polynomials. Detection of unentangled mixed states is related to separability of Hermitian tensors. We refer to [1, 12, 21, 33] for applications of density matrices. Quantum information theory is closely related to tensors [42, 86, 100, 101, 127]. The separability issue will be studied in section 4.6.

## 4.2 Basis Hermitian tensors

For convenience, denote

$$N := n_1 \cdots n_m, \quad \mathcal{S} := \Big\{(i_1, \ldots, i_m) : i_1 \in [n_1], \ldots, i_m \in [n_m]\Big\}.$$

The cardinality of the label set $\mathcal{S}$ is $N$. For two labelling tuples $I := (i_1, \ldots, i_m)$ and $J := (j_1, \ldots, j_m)$ in $\mathcal{S}$, define the ordering $I < J$ if the first nonzero entry of $I - J$ is negative. For a scalar $c \in \mathbb{C}$, denote by $\mathcal{E}^{IJ}(c)$ the Hermtian tensor in $\mathbb{C}^{[n_1,\ldots,n_m]}$ such that

$$\big(\mathcal{E}^{IJ}(c)\big)_{i_1\cdots i_m j_1 \cdots j_m} = \overline{\big(\mathcal{E}^{JI}(c)\big)}_{j_1\cdots j_m i_1 \cdots i_m} = c$$

and all other entries are zeros. We adopt the standard scalar multiplication and addition for $\mathbb{C}^{[n_1,\ldots,n_m]}$, so $\mathbb{C}^{[n_1,\ldots,n_m]}$ is a vector space over $\mathbb{R}$. The set

$$E := \Big\{\mathcal{E}_{II}(1)\Big\}_{I\in\mathcal{S}} \bigcup \Big\{\mathcal{E}_{IJ}(1), \mathcal{E}_{IJ}(\sqrt{-1})\Big\}_{I,J\in\mathcal{S}, I<J} \tag{4.5}$$

is the *canonical basis* for $\mathbb{C}^{[n_1,\ldots,n_m]}$. Its dimension is

$$\dim \mathbb{C}^{[n_1,\ldots,n_m]} = N + N(N-1) = N^2.$$

For these basis tensors, we determine their Hermitian ranks as well as the rank decompositions. For a basis tensor $\mathcal{E}^{IJ}(c)$, we are interested in $c = 1$ or $\sqrt{-1}$. Its Hermitian rank can be determined by reduction to the 2-dimensional case.

**Lemma 4.1** ( [110]). *Suppose the dimensions $n_1, \ldots, n_m \geq 2$, $I = (i_1, \ldots, i_m)$, and $J = (j_1, \ldots, j_m)$. For each $k = 1, \ldots, m$, let*

$$(i'_k, j'_k) := (1,1) \quad \text{if } i_k = j_k, \quad (i'_k, j'_k) := (1,2) \quad \text{if } i_k \neq j_k.$$

*Let $I' := (i'_1, \ldots, i'_m)$, $J' := (j'_1, \ldots, j'_m)$. Then, $\mathcal{E}^{I'J'}(c) \in \mathbb{C}^{[2,\ldots,2]}$ and*

$$\operatorname{hrank} \mathcal{E}^{IJ}(c) = \operatorname{hrank} \mathcal{E}^{I'J'}(c).$$

*Proof.* For each $k$, if $i_k = j_k$, let $P_k$ be the permutation matrix that switches the 1st and $i_k$th rows; if $i_k \neq j_k$, let $P_k$ be the permutation matrix that switches $i_k$th row and $j_k$th row to 1st row and 2nd row respectively. Consider the orthogonal congruent transformation

$$\mathcal{F} := (P_1, \ldots, P_m) \times_{cong} \mathcal{E}^{IJ}(c).$$

Then $\mathcal{F}$ is the Hermitian tensor such that $\mathcal{F}_{I'J'} = \overline{\mathcal{F}_{J'I'}} = c$ and all other entries are zeros, so $\mathcal{F}$ is a canonical basis tensor. Note that $\mathcal{E}^{I'J'}(c)$ is the subtensor of $\mathcal{F}$, consisting of the first two labels for each dimension, hence $\mathcal{E}^{I'J'}(c)$ and $\mathcal{F}$ have the same rank. Since nonsingular congruent transformations preserve Hermitian ranks (see Proposition 4.7), hrank $\mathcal{E}^{IJ}(c) =$ hrank $\mathcal{E}^{I'J'}(c)$. □

In the following, for $n_1 = \cdots = n_m = 2$ and $I = (1 \ldots 1)$, $J = (2 \ldots 2)$, we determine the Hermitian rank of the basis tensor $\mathcal{E}^{IJ}(c)$. First, we consider $c = 1$. For each $k = 0, 1, \ldots, m$, let

$$\theta_k := k\pi/m, \quad u_k := (1, \exp\big(\theta_k\sqrt{-1}\big)). \tag{4.6}$$

The following Hermitian tensor

$$\mathcal{A}_k := \frac{1}{2}\big([u_k, u_k, \ldots, u_k]_{\otimes h} + [\overline{u_k}, \overline{u_k}, \ldots, \overline{u_k}]_{\otimes h}\big) \tag{4.7}$$

has rank 1 or 2. For each $s = 0, 1, \ldots, m$, let $J_s := (1, \ldots, 1, 2, \ldots, 2)$ where 2 appears $s$ times. The tensor $\mathcal{A}_k$ has only $m+1$ distinct entries, which are

$$(\mathcal{A}_k)_{IJ_s} = \mathrm{Re}\big((u_k)_2^s\big) = \mathrm{Re}\big(\exp(s\theta_k\sqrt{-1})\big) = \cos(s\theta_k), \quad s = 0, 1, \ldots, m.$$

For each $k$, consider the vector

$$w_k := \big(\cos(0\cdot\theta_k), \cos(1\cdot\theta_k), \ldots, \cos(m\cdot\theta_k)\big).$$

Let $\lambda_k := 2(-1)^k$ for $1 \le k \le m-1$, $\lambda_k := (-1)^k$ for $k = 0, m$, and

$$u := \lambda_0 w_0 + \lambda_1 w_1 + \cdots + \lambda_m w_m. \tag{4.8}$$

For $p = 0, 1, \ldots, m$, the $(p+1)$th entry of $u$ is

$$(u)_{p+1} = \sum\nolimits_{k=0}^{m} \lambda_k \cos(p\theta_k) = \sum\nolimits_{k=0}^{m} \lambda_k \cos(\frac{pk}{m}\pi) = \mathrm{Re}\Big(\sum\nolimits_{k=0}^{m} \lambda_k \exp(\frac{pk}{m}\pi\sqrt{-1})\Big).$$

For each $p = 0, 1, \ldots, m-1$, one can check that (let $\alpha := \frac{p}{m}\pi$ )

$$\begin{aligned}\sum\nolimits_{k=0}^{m} \lambda_k \exp(k\alpha\sqrt{-1}) &= 2\sum\nolimits_{k=0}^{m} (-1)^k \exp(k\alpha\sqrt{-1}) - 1 - (-1)^m \exp(p\pi\sqrt{-1}) \\ &= 2\frac{1 - (-\exp((m+1)\alpha\sqrt{-1})}{1 + \exp(\alpha\sqrt{-1})} - 1 - (-1)^{m+p} \\ &= \begin{cases} 0 & \text{if } m+p \text{ is even,} \\ \frac{-4\sin\alpha}{(1+\cos\alpha)^2 + (\sin\alpha)^2}\sqrt{-1} & \text{if } m+p \text{ is odd.} \end{cases}\end{aligned}$$

Hence, $(u)_{p+1} = 0$ for $0 \le p \le m-1$. Moreover,

$$(u)_{m+1} = \sum\nolimits_{k=0}^{m} \lambda_k \cos(m\theta_k) = \sum\nolimits_{k=0}^{m} \lambda_k \cos(k\pi) = \sum\nolimits_{k=0}^{m} \lambda_k (-1)^k = 2m.$$

Therefore, we have

$$\sum\nolimits_{k=0}^{m} \frac{\lambda_k}{2m} w_k = (0, \ldots, 0, 1), \quad \mathcal{E}^{IJ}(1) = \sum\nolimits_{k=0}^{m} \frac{\lambda_k}{2m} \mathcal{A}_k.$$

This gives the Hermitian decomposition of length $2m$:

$$\mathcal{E}^{IJ}(1) = \frac{1}{2m}\Big([u_0, u_0, \ldots, u_0]_{\otimes h} + (-1)^m [u_m, u_m, \ldots, u_m]_{\otimes h} + \sum\nolimits_{k=1}^{m-1} (-1)^k ([u_k, u_k, \ldots, u_k]_{\otimes h} + [\overline{u_k}, \overline{u_k}, \ldots, \overline{u_k}]_{\otimes h})\Big), \quad (4.9)$$

where $u_k$ is given as in (4.6). For the case $c \neq 0$, one can verify that

$$\mathcal{E}^{IJ}(c) = (\begin{pmatrix} c & 0 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \ldots, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}) \times_{cong} \mathcal{E}^{IJ}(1).$$

Then, the decomposition (4.9) implies that

$$\mathcal{E}^{IJ}(c) = \frac{1}{2m}\Big([\tilde{u}_0, u_0, \ldots, u_0]_{\otimes h} + (-1)^m [\tilde{u}_m, u_m, \ldots, u_m]_{\otimes h} + \sum\nolimits_{k=1}^{m-1} (-1)^k ([\tilde{u}_k, u_k, \ldots, u_k]_{\otimes h} + [\tilde{v}_k, \overline{u_k}, \ldots, \overline{u_k}]_{\otimes h})\Big), \quad (4.10)$$

where $\tilde{u}_k = \big(c, \exp(\frac{k}{m}\pi\sqrt{-1})\big)$ and $\tilde{v}_k = \big(c, \exp(-\frac{k}{m}\pi\sqrt{-1})\big)$.

**Proposition 4.2** ( [110]). *Assume $n_1 = \cdots = n_m = 2$, $I = (1 \ldots 1)$, $J = (2 \ldots 2)$, and $c \neq 0$. Then,* $\mathrm{hrank}(\mathcal{E}(c)) = 2m$ *and (4.10) is a Hermitian rank decomposition.*

*Proof.* The decomposition (4.10) implies $\mathrm{hrank}(\mathcal{E}^{IJ}(c)) \le 2m$, so we only need to show $\mathrm{hrank}(\mathcal{E}^{IJ}(c)) \ge 2m$. We prove it by induction on $m$.

When $m = 1$, $\mathcal{E}^{(12)}(c)$ is a Hermitian matrix of rank 2 and the conclusion is clearly true. Suppose the conclusion holds for $m = 1, 2, \ldots, k$. Assume to the contrary that for $m = k+1$, $r := \mathrm{hrank}(\mathcal{E}^{IJ}(c)) \le 2m-1 = 2k+1$ and $\mathcal{E}^{IJ}(c)$ has the Hermitian decomposition (for nonzero vectors $u_i^j$):

$$\mathcal{E}^{IJ}(c) = \sum\nolimits_{i=1}^{r} \lambda_i [u_i^1, \ldots, u_i^{k+1}]_{\otimes h}.$$

Let $\mathcal{A}_i = \lambda_i [u_i^1, \ldots, u_i^k]_{\otimes h}$, $U_i = u_i^{k+1} \otimes \overline{u_i^{k+1}}$, then $\mathcal{E}^{IJ}(c)$ can be rewritten as (after a reordering of tensor products)

$$\mathcal{E}^{IJ}(c) = \sum\nolimits_{i=1}^{r} \mathcal{A}_i \otimes U_i.$$

Let $p$ be the dimension of $\mathrm{span}\{U_1,\dots,U_r\}$ and one can generally assume $\{U_1,\dots,U_p\}$ is linearly independent. Then $U_j=\sum_{s=1}^p \alpha_s^j U_s$, $j>p$, for some real coefficients $\alpha_s^j$, since each $U_i$ can be viewed as a Hermitian matrix. So we can rewrite that

$$\mathcal{E}^{IJ}(c)=\sum\nolimits_{i=1}^{p}\mathcal{B}_i\otimes U_i \quad \text{where} \quad \mathcal{B}_i:=\mathcal{A}_i+\sum\nolimits_{j=p+1}^{r}\alpha_i^j\mathcal{A}_j.$$

Each $\mathcal{B}_i$ is a Hermitian tensor of order $2k$, and $\mathrm{hrank}(\mathcal{B}_i)\le r-p+1$. For two labels $I',J'\in\mathbb{N}^k$, consider the matrix

$$M^{I'J'}:=\begin{bmatrix}(\mathcal{E}^{IJ}(c))_{(I',1)(J',1)} & (\mathcal{E}^{IJ}(c))_{(I',1)(J',2)}\\ (\mathcal{E}^{IJ}(c))_{(I',2)(J',1)} & (\mathcal{E}^{IJ}(c))_{(I',2)(J',2)}\end{bmatrix}=\sum_{i=1}^{p}(\mathcal{B}_i)_{I'J'}U_i.$$

Note that $M^{I'J'}\neq 0$ if and only if $I'=(1\cdots 1), J'=(2\cdots 2)$ or $I'=(2\cdots 2), J'=(1\cdots 1)$. Since $U_1,\dots,U_p$ are linearly independent, $((\mathcal{B}_1)_{I'J'},\dots,(\mathcal{B}_p)_{I'J'})\neq 0$ if and only if $I'=(1\cdots 1), J'=(2\cdots 2)$ or $I'=(2\cdots 2), J'=(1\cdots 1)$. So each nonzero $\mathcal{B}_i$ is also a canonical basis tensor in $\mathbb{C}^{[2,\dots,2]}$. By induction, we have

$$r-p+1\ge \mathrm{hrank}(\mathcal{B}_i)\ge 2k, \qquad p\le r+1-2k\le 2.$$

By the same argument, we can show that the rank of the set $V_j:=\left\{u_i^j\otimes\overline{u_i^j}\right\}_{i=1}^r$ is at most 2, for all $j=1,\dots,m$. If the rank of $V_j$ is 2, then there exists $t_j\in[r]$ such that $\{u_1^j\otimes\overline{u_1^j},u_{t_j}^j\otimes\overline{u_{t_j}^j}\}$ is linearly independent. If the rank of $V_j$ is 1, we let $t_j:=1$. Thus $u_i^j=u_1^j$ or $u_i^j=u_{t_j}^j$ for each $i=1,\dots,r$. For each $j$, there exists $w^j$ such that $(w^j)^T\overline{u_1^j}=1$, and $(w^j)^T\overline{u_{t_j}^j}=0$ if $t_j>1$. Then, consider the multilinear matrix-tensor product

$$\mathcal{T}:=(I_2,\dots,I_2,(w^1)^T,\dots,(w^{k+1})^T)\times\mathcal{E}^{IJ}(c)=\lambda_1u_1^1\otimes\cdots\otimes u_1^{k+1}\in\mathbb{C}^{2\times\cdots\times 2}.$$

When $(s_1\cdots s_{k+1})\neq(1,\dots,1)$ or $(2,\dots,2)$, we have

$$\mathcal{T}_{s_1\cdots s_{k+1}}=\sum_{j_1,\dots,j_{k+1}=1,2}(w^1)_{j_1}\cdots(w^{k+1})_{j_{k+1}}(\mathcal{E}^{IJ}(c))_{s_1\dots s_{k+1}j_1\dots j_{k+1}}=0.$$

So $\mathcal{T}$ has at most two nonzero entries, which must be $\mathcal{T}_{1\cdots 1}$ and/or $\mathcal{T}_{2\cdots 2}$:

$$\begin{aligned}\mathcal{T}_{1\cdots 1}&=(\mathcal{E}^{IJ}(c))_{(1\cdots 1)(2\cdots 2)}(w^1)_2\cdots(w^{k+1})_2=c(w^1)_2\cdots(w^{k+1})_2,\\ \mathcal{T}_{2\cdots 2}&=(\mathcal{E}^{IJ}(c))_{(2\cdots 2)(1\cdots 1)}(w^1)_1\cdots(w^{k+1})_1=\overline{c}(w^1)_1\cdots(w^{k+1})_1.\end{aligned}$$

Since $\mathcal{T}$ is rank 1, only one of $\mathcal{T}_{1\cdots 1},\mathcal{T}_{2\cdots 2}$ is nonzero, which is also the unique nonzero entry of $\mathcal{T}$. Without loss of generality, assume $\mathcal{T}_{1\cdots 1}\neq 0,\mathcal{T}_{2\cdots 2}=0$. The fact that $(\mathcal{T})_{1\cdots 1}$ is the only one nonzero entry implies $u_1^j=\mu_j e_1, j=1\cdots k+1$ for some $0\neq\mu_j\in\mathbb{C}$. The equation $(w^j)^T\overline{u_1^j}=\overline{\mu}_j(w^j)_1=1$ implies that $(w^j)_1\neq 0$, so $\mathcal{T}_{2\cdots 2}=\overline{c}(w^1)_1\cdots(w^{k+1})_1\neq 0$. But this contradicts $\mathcal{T}_{2\cdots 2}=0$, hence $\mathrm{hrank}(\mathcal{E}^{IJ}(c))\ge 2m$. □

Ranks of basis tensors $\mathcal{E}^{IJ}(c)$ for general dimensions are given as follows.

**Theorem 4.3** ( [110]). *Assume $n_1, \dots, n_m \geq 2$, $I = (i_1, \dots, i_m), J = (j_1, \dots, j_m)$, and $c \neq 0$. If $I = J$, then $\operatorname{hrank} \mathcal{E}^{IJ}(c) = 1$; if $I \neq J$, then $\operatorname{hrank} \mathcal{E}^{IJ}(c) = 2d$ where $d$ is the number of nonzero entries of $I - J$.*

*Proof.* When $I = J$, $\mathcal{E}^{IJ}(c)$ is a Hermitian tensor only if $c$ is real, and $\mathcal{E}^{II}(c) = c[e_{i_1}, \dots, e_{i_m}]_{\otimes h}$. So, $\operatorname{hrank} \mathcal{E}^{II}(c) = 1$. When $I \neq J$, we can generally assume $i_k \neq j_k$ for $k = 1, \dots, d$, and $i_k = j_k$ for $k = d+1, \dots, m$. By Lemma 4.1, $\mathcal{E}^{IJ}(c)$ has the same Hermitian rank as $\mathcal{E}^{I'J'}(c)$, for $I' = (1, \dots, 1)$ and $J' = (2, \dots, 2, 1, \dots, 1)$ (the first $d$ entries of $J'$ are 2's). Let $I_1 = (1, \dots, 1), I_2 = (2, \dots, 2)$, where $1, 2$ are repeated for $d$ times. Then $\operatorname{hrank} \mathcal{E}^{I'J'}(c) = \operatorname{hrank} \mathcal{E}^{I_1 J_1}(c)$. By Proposition 4.2, we know $\operatorname{hrank} \mathcal{E}^{IJ}(c) = \operatorname{hrank} \mathcal{E}^{I_1 J_1}(c) = 2d$. □

The following is an example of Hermitian rank decompositions for basis tensors.

**Example 4.4.** *For $I = (1, 2)$, $J = (3, 4)$ and $c \neq 0$, the basis tensor $\mathcal{E}^{(12)(34)}(c) \in \mathbb{C}^{[4,4]}$ has the Hermitian rank 4, with the following Hermitian rank decomposition (in the following $i := \sqrt{-1}$)*

$$\frac{1}{4}\left[\begin{pmatrix} c \\ 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ 1 \end{pmatrix}\right]_{\otimes h} + \frac{1}{4}\left[\begin{pmatrix} c \\ 0 \\ -1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ -1 \end{pmatrix}\right]_{\otimes h} - \frac{1}{4}\left[\begin{pmatrix} c \\ 0 \\ i \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ i \end{pmatrix}\right]_{\otimes h} - \frac{1}{4}\left[\begin{pmatrix} c \\ 0 \\ -i \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \\ -i \end{pmatrix}\right]_{\otimes h}.$$

In some occasions, a Hermitian tensor may be given by a Hermitian decomposition. One wonders whether that is a rank decomposition or not. This question is related to the classical Kruskal theorem [78, 133]. For a set $S$ of vectors, its *Kruskal rank*, denoted as $k_S$, is the maximum number $k$ such that every subset of $k$ vectors in $S$ is linearly independent.

**Proposition 4.5** ( [110]). *Let $\mathcal{H} = \sum_{j=1}^{r} \lambda_j [u_j^1, \dots, u_j^m]_{\otimes h}$ be a Hermitian tensor, with $0 \neq \lambda_j \in \mathbb{R}$ and $m > 1$. For each $i = 1, \dots, m$, let $U_i := \{u_1^i, \dots, u_r^i\}$. If*

$$k_{U_1} + \cdots + k_{U_m} \geq r + m, \tag{4.11}$$

*then hrank$(\mathcal{H}) = r$ and the Hermitian rank decomposition of $\mathcal{H}$ is essentially unique, i.e., it is unique up to permutation and scaling of decomposing vectors.*

*Proof.* Note that $k_{U_i} = k_{\overline{U_i}}$, where $\overline{U_i} := \{\overline{u_j^i}, \dots, \overline{u_j^i}\}$. The rank condition (4.11) is equivalent to that

$$k_{U_1} + \cdots + k_{U_m} + k_{\overline{U_1}} + \cdots + k_{\overline{U_m}} \geq 2r + 2m - 1.$$

The conclusion is then implied by the classical Kruskal type theorem [78, 133] (or see Theorems 12.5.3.1 and 12.5.3.2 in [79]). □

For instance, for the following vectors

$$u_1 = (1,1,1),\ u_2 = (1,1,0),\ u_3 = (1,0,1),\ u_4 = (0,1,1),$$

the sum $\sum_{i=1}^{4}[u_i,u_i,u_i]_{\otimes h}$ has Hermitian rank 4, by Proposition 4.5. This is because, for $U = \{u_1,u_2,u_3,u_4\}$, the Kruskal rank $k_U = 3$, $m = 3$ and $3k_U = 9 \geq 4 + m = 7$.

A basic question is how to compute Hermitian rank decompositions. This is generally a challenge. When Hermitian ranks are small, we can apply the existing methods for canonical polyadic decompositions (CPDs) for cubic tensors. For convenience, let

$$N_1 := n_1 \cdots n_m, \quad N_3 := \min\{n_1,\ldots,n_m\}, \quad N_2 = N_1/N_3. \tag{4.12}$$

Up to a permutation of dimensions, we can assume $n_m$ is the smallest, i.e., $N_3 = n_m$. A Hermitian tensor can be flattened to a cubic tensor. Define the linear flattening mapping $\psi : \mathbb{C}^{[n_1,\ldots,n_m]} \to \mathbb{C}^{N_1\times N_2\times N_3}$ such that

$$\psi([u^1,\ldots,u^m]_{\otimes h}) \;=\; (u^1\otimes\cdots\otimes u^m)\otimes(\overline{u^1}\otimes\cdots\otimes\overline{u^{m-1}})\otimes\overline{u^m}. \tag{4.13}$$

Then $\mathcal{H} = \sum_{j=1}^{r}\lambda_j[u_j^1,\ldots,u_j^m]_{\otimes h}$ if and only if

$$\psi(\mathcal{H}) = \sum\nolimits_{j=1}^{r}\lambda_j\, a_j\otimes b_j\otimes c_j \tag{4.14}$$

where $a_j = u_j^1\otimes\cdots\otimes u_j^m$, $b_j = \overline{u_j^1}\otimes\cdots\overline{u_j^{m-1}}$, $c_j = \overline{u_j^m}$. The decomposition (4.14) can be obtained by computing the CPD for $\psi(\mathcal{H})$, if the rank decomposition of $\psi(\mathcal{H})$ is unique. We refer to [4, 16, 37, 38, 142] for computing CPDs.

**Example 4.6.** *Consider the tensor $\mathcal{A} \in \mathbb{C}^{[3,3]}$ such that $\mathcal{A}_{i_1i_2j_1j_2} = i_1j_1 + i_2j_2$ for all $i_1,i_2,j_1,j_2$ in the range. A Hermitian decomposition for $\mathcal{A}$ is*

$$\mathcal{A} = \left[\begin{pmatrix}1\\2\\3\end{pmatrix},\begin{pmatrix}1\\1\\1\end{pmatrix}\right]_{\otimes h} + \left[\begin{pmatrix}1\\1\\1\end{pmatrix},\begin{pmatrix}1\\2\\3\end{pmatrix}\right]_{\otimes h}.$$

*By Proposition 4.5, the Hermitian rank is* 2.

The rank of a Hermitian matrix does not change after a nonsingular congruent transformation. The same conclusion holds for Hermitian tensors. We refer to (4.4) for multi-linear congruent transformations.

**Proposition 4.7.** *Let $Q_k \in \mathbb{C}^{n_k \times n_k}$ be nonsingular matrices, for $k = 1, \ldots, m$. Then, for each $\mathcal{H} \in \mathbb{C}^{[n_1,\ldots,n_m]}$, the congruent transformation $(Q_1, \ldots, Q_m) \times_{cong} \mathcal{H}$ has the same Hermitian rank as $\mathcal{H}$ does.*

*Proof.* Let $\mathcal{F} := (Q_1, \ldots, Q_m) \times_{cong} \mathcal{H}$, then $\mathcal{H} = \sum_{i=1}^{r} \lambda_i [u_i^1, \ldots, u_i^m]_{\otimes h}$ if and only if

$$\mathcal{F} = \sum\nolimits_{i=1}^{r} \lambda_i [Q_1 u_i^1, \ldots, Q_m u_i^m]_{\otimes h},$$

because each $Q_i$ is nonsingular. So $\operatorname{hrank}(\mathcal{H}) = \operatorname{hrank}(\mathcal{F})$. □

## 4.3 Real Hermitian tensors

This section discusses real Hermitian tensors, i.e., their entries are all real. The subspace of real Hermitian tensors in $\mathbb{C}^{[n_1,\ldots,n_m]}$ is denoted as

$$\mathbb{R}^{[n_1,\ldots,n_m]} := \mathbb{C}^{[n_1,\ldots,n_m]} \cap \mathbb{R}^{n_1 \times \cdots \times n_m \times n_1 \times \cdots \times n_m}.$$

For real Hermitian tensors, we are interested in their real decompositions.

**Definition 4.8.** *A tensor $\mathcal{H} \in \mathbb{R}^{[n_1,\ldots,n_m]}$ is called* $\mathbb{R}$-Hermitian decomposable *if*

$$\mathcal{H} = \sum\nolimits_{i=1}^{r} \lambda_i [u_i^1, \ldots, u_i^m]_{\otimes h} \tag{4.15}$$

*for real vectors $u_i^j \in \mathbb{R}^{n_j}$ and real scalars $\lambda_i \in \mathbb{R}$. The smallest such $r$ is called the* $\mathbb{R}$-Hermitian rank *of $\mathcal{H}$, for which we denote* $\operatorname{hrank}_{\mathbb{R}}(\mathcal{H})$. *The subspace of $\mathbb{R}$-Hermitian decomposable tensors in $\mathbb{R}^{[n_1,\ldots,n_m]}$ is denoted as $\mathbb{R}_D^{[n_1,\ldots,n_m]}$.*

When it exists, (4.15) is called a $\mathbb{R}$-Hermitian decomposition; if $r$ is minimum, (4.15) is called a $\mathbb{R}$-Hermitian rank decomposition. Clearly, for all $\mathcal{H} \in \mathbb{R}_D^{[n_1,n_2]}$,

$$\operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) \geq \operatorname{hrank}(\mathcal{H}). \tag{4.16}$$

Not every real Hermitian tensor is $\mathbb{R}$-Hermitian decomposable. This is very different from the complex case. We characterize when a tensor is $\mathbb{R}$-Hermitian decomposable.

**Theorem 4.9** ( [110])**.** *A tensor $\mathcal{A} \in \mathbb{R}^{[n_1,\ldots,n_m]}$ is $\mathbb{R}$-Hermitian decomposable, i.e., $\mathcal{A} \in \mathbb{R}_D^{[n_1,\ldots,n_m]}$, if and only if*

$$\mathcal{A}_{i_1 \ldots i_m j_1 \ldots j_m} = \mathcal{A}_{k_1 \ldots k_m l_1 \ldots l_m} \tag{4.17}$$

*for all labels such that $\{i_s, j_s\} = \{k_s, l_s\}$, $s = 1, \ldots, m$.*

*Proof.* For convenience, denote the labeling tuples:

$$\imath = (i_1, \ldots, i_m, j_1, \ldots, j_m),\ \jmath = (k_1, \ldots, k_m, l_1, \ldots, l_m).$$

" $\Rightarrow$ " : If $\mathcal{A}$ has a $\mathbb{R}$-Hermitian decomposition as in (4.15), then

$$\mathcal{A}_\imath = \sum_{i=1}^{r} \lambda_i \prod_{s=1}^{m} (u_i^s)_{i_s} (u_i^s)_{j_s} = \sum_{i=1}^{r} \lambda_i \prod_{s=1}^{m} (u_i^s)_{k_s} (u_i^s)_{l_s} = \mathcal{A}_\jmath$$

when $\{i_s, j_s\} = \{k_s, l_s\}$ for all $s = 1, \ldots, m$.

" $\Leftarrow$ " : Assume (4.17) holds. We prove the conclusion by induction on $m$. For $m = 2$, i.e., the matrix case, the conclusion is clearly true because every real symmetric matrix has a real spectral decomposition. Suppose the conclusion is true for $m$, then we show that it is also true for $m + 1$. For $s, t \in [n_{m+1}]$, let $\mathcal{B}^{s,t}$ be the tensor in $\mathbb{R}^{[n_1,\ldots,n_m]}$ such that

$$(\mathcal{B}^{s,t})_{i_1 \ldots i_m j_1 \ldots j_m} = (\mathcal{A})_{i_1 \ldots i_m s j_1 \ldots j_m t}$$

for all $i_1, \ldots, i_m j_1, \ldots, j_m$ in the range. The condition (4.17) implies that $\mathcal{B}^{s,t} = \mathcal{B}^{t,s}$ and each $\mathcal{B}^{s,t}$ is a real Hermitian tensor. For $s < t$, define the linear map

$$\rho_{s,t} : \mathbb{R}^{[n_1,\ldots,n_m]} \to \mathbb{R}^{[n_1,\ldots,n_m,n_{m+1}]},$$

$$[x_1, \ldots, x_m]_{\otimes h} \mapsto \frac{1}{2}[x_1, \ldots, x_m, e_s + e_t]_{\otimes h} - \frac{1}{2}[x_1, \ldots, x_m, e_s - e_t]_{\otimes h}.$$

For $s = t$, the linear map $\rho_{s,s}$ is then defined such that

$$\rho_{s,s}([x_1, \ldots, x_m]_{\otimes h}) = [x_1, \ldots, x_m, e_s]_{\otimes h}.$$

One can verify that $\mathcal{A} = \sum_{1 \le s \le t \le n_{m+1}} \rho_{s,t}(\mathcal{B}^{s,t})$. By induction, each $\mathcal{B}^{s,t}$ is $\mathbb{R}$-Hermitian decomposable, so each $\rho_{s,t}(\mathcal{B}^{s,t})$, as well as $\mathcal{A}$, is also $\mathbb{R}$-Hermitian decomposable. □

**Example 4.10.** *Consider the real Hermitian tensor $\mathcal{A} \in \mathbb{R}^{[2,2]}$ such that*

$$\mathcal{A}_{ijkl} = i + j + k + l$$

*for all $1 \le i, j, k, l \le 2$. It is a Hankel tensor [117]. By Theorem 4.9, it is $\mathbb{R}$-Hermitian decomposable. In fact, it has the decomposition*

$$\mathcal{A} = \frac{40 - 13\sqrt{10}}{20}\big([u_1, e]_{\otimes h} + [e, u_1]_{\otimes h}\big) + \frac{40 + 13\sqrt{10}}{20}\big([u_2, e]_{\otimes h} + [e, u_2]_{\otimes h}\big),$$

*for* $u_1 = (\frac{-\sqrt{10}-1}{3}, 1)$, $u_2 = (\frac{\sqrt{10}-1}{3}, 1)$. *Clearly,* $\operatorname{hrank}_{\mathbb{R}}(\mathcal{A}) \leq 4$. *Moreover,* $\mathcal{A}$ *can be expressed as the limit*

$$\mathcal{A} = \lim_{\epsilon \to 0} \epsilon^{-1} \Big[ (e + \epsilon f)^{\otimes 4} - e^{\otimes 4} \Big],$$

*for* $f := (1, 2)$. *For this kind of tensors, the cp rank is* 4 *(see [28, §5], [39, §4.7]). Therefore,* $\operatorname{hrank}_{\mathbb{R}}(\mathcal{A}) \geq \mathit{rank}(\mathcal{A}) = 4$ *and hence* $\operatorname{hrank}_{\mathbb{R}}(\mathcal{A}) = 4$.

Not every basis tensor $\mathcal{E}^{IJ}(c)$ is $\mathbb{R}$-Hermitian decomposable. For instance, the basis tensor $\mathcal{A} = \mathcal{E}^{1122}(1)$ is not, because $\mathcal{A}_{1122} = 1 \neq 0 = \mathcal{A}_{1221}$.

**Corollary 4.11** ( [110])**.** *For* $I = (i_1, \ldots, i_m)$ *and* $J = (j_1, \ldots, j_m)$, *the basis tensor* $\mathcal{E}^{IJ}(1)$ *is* $\mathbb{R}$*-Hermitian decomposable if and only if* $I - J$ *has at most one nonzero entry. In particular, if* $I = J$, *then* $\operatorname{hrank}_{\mathbb{R}} \mathcal{E}^{IJ}(1) = 1$*; if* $I$ *and* $J$ *differs for only one entry, then* $\operatorname{hrank}_{\mathbb{R}} \mathcal{E}^{IJ}(1) = 2$.

*Proof.* The necessity direction is a direct consequence of Theorem 4.9. This is because if there are two distinct $k$ such that $i_k \neq j_k$, then the condition (4.17) cannot be satisfied. We prove the sufficiency direction by constructing $\mathbb{R}$-Hermitian decompositions explicitly. If $I = J$, then $\mathcal{E} = [e_{i_1}, e_{i_2}, \ldots, e_{i_m}]_{\otimes h}$ and $\operatorname{hrank}_{\mathbb{R}} \mathcal{E}^{IJ}(1) = 1$. If $I$ and $J$ differs for only one entry, say, $i_k \neq j_k$, then

$$\mathcal{E} = \frac{1}{2}[e_{i_1}, e_{i_2}, \ldots, e_{i_k} + e_{j_k}, \cdots e_{i_m}]_{\otimes h} - \frac{1}{2}[e_{i_1}, e_{i_2}, \ldots, e_{i_k} - e_{j_k}, \cdots e_{i_m}]_{\otimes h}$$

and hence $\operatorname{hrank}_{\mathbb{R}} \mathcal{E}^{IJ}(1) \leq 2$. Since $\operatorname{hrank}_{\mathbb{R}} \mathcal{E}^{IJ}(1) \geq \operatorname{hrank} \mathcal{E}^{IJ}(1) = 2$, we must have $\operatorname{hrank}_{\mathbb{R}} \mathcal{E}^{IJ}(1) = 2$. □

The major reason for not all real Hermitian tensors are $\mathbb{R}$-Hermitian decomposable is because of the dimensional difference. That is, the dimension of $\mathbb{R}_D^{[n_1,\ldots,n_m]}$ is less than that of $\mathbb{R}^{[n_1,\ldots,n_m]}$. By Theorem 4.9, the dimension of $\mathbb{R}_D^{[n_1,\ldots,n_m]}$ is equal to the cardinality of the set $\{(i_1, \ldots, i_m, j_1, \ldots, j_m) : 1 \leq i_k \leq j_k \leq n_k\}$. Thus

$$\dim \mathbb{R}_D^{[n_1,\ldots,n_m]} = \prod_{k=1}^{m} \binom{n_k + 1}{2} = \prod_{k=1}^{m} \frac{n_k(n_k+1)}{2}. \tag{4.18}$$

However, the dimension of $\mathbb{R}^{[n_1,\ldots,n_m]}$ is

$$\dim \mathbb{R}^{[n_1,\ldots,n_m]} = \binom{N+1}{2}, \quad N = n_1 \cdots n_m. \tag{4.19}$$

The dimension of $\mathbb{R}^{[n_1,\ldots,n_m]}$ equals the dimension of $\mathcal{S}^N$, the space of $N$-by-$N$ real symmetric matrices. The dimension of $\mathbb{R}_D^{[n_1,\ldots,n_m]}$ equals the dimension of the tensor product space

$\mathcal{S}^{n_1} \otimes \cdots \otimes \mathcal{S}^{n_m}$. If $m > 1$ and all $n_i > 1$, then

$$\dim \mathbb{R}_D^{[n_1,\ldots,n_m]} < \dim \mathbb{R}^{[n_1,\ldots,n_m]}. \tag{4.20}$$

Real Hermitian decompositions can also be equivalently expressed in terms of real polynomials. Let each $x_i \in \mathbb{R}^{n_i}$ be a real vector variable. The real decomposition (4.15) implies that

$$\mathcal{H}(x,x) = \sum\nolimits_{i=1}^{r} \lambda_i \big((u_i^1)^T x_1\big)^2 \cdots \big((u_i^m)^T x_m\big)^2. \tag{4.21}$$

When $\mathcal{H}$ is $\mathbb{R}$-Hermitian decomposable, (4.21) also implies (4.15).

**Lemma 4.12** ( [110]). *For real vectors $u_i^j$, a tensor $\mathcal{H} \in \mathbb{R}_D^{[n_1,\ldots,n_m]}$ has the decomposition (4.21) if and only if the $\mathbb{R}$-Hermitian decomposition (4.15) holds.*

*Proof.* The "if" direction is obvious. We prove the "only if" direction. Let

$$\mathcal{U} = \sum\nolimits_{i=1}^{r} \lambda_i [u_i^1, \ldots, u_i^m]_{\otimes h}.$$

Then $\langle \mathcal{H} - \mathcal{U}, [x_1, \ldots, x_m]_{\otimes h} \rangle = 0$ for all real $x_i \in \mathbb{R}^{n_i}$. Since $\mathcal{H} - \mathcal{U} \in \mathbb{R}_D^{[n_1,\ldots,n_m]}$, $\langle \mathcal{H} - \mathcal{U}, \mathcal{H} - \mathcal{U} \rangle = 0$, so $\mathcal{H} = \mathcal{U}$ and (4.15) holds. □

In the following, we study the relationship between real and complex Hermitian decompositions.

**Lemma 4.13** ( [110]). *Suppose $\mathcal{H} \in \mathbb{R}_D^{[n_1,\ldots,n_m]}$ has the decomposition*

$$\mathcal{H} = \sum\nolimits_{j=1}^{r} \lambda_j [u_j^1, u_j^2, \ldots, u_j^m]_{\otimes h},$$

*with complex $u_j^i \in \mathbb{C}^{n_i}, 0 \neq \lambda_j \in \mathbb{R}$. Let*

$$U := \left[ (u_1^1 \boxtimes \overline{u_1^1} \boxtimes \cdots u_1^{m-1} \boxtimes \overline{u_1^{m-1}}), \quad \cdots, \quad (u_r^1 \boxtimes \overline{u_r^1} \boxtimes \cdots u_r^{m-1} \boxtimes \overline{u_r^{m-1}}) \right].$$

*If $k := rank(U) \in \{1, 2, r\}$, then*

$$\mathcal{H} = \sum\nolimits_{j=1}^{r} \beta_j [u_j^1, u_j^2, \ldots, u_j^{m-1}, s_j^m]_{\otimes h} \tag{4.22}$$

*for real vectors $s_j^m \in \mathbb{R}^{n_m}$ and real scalars $\beta_j \in \mathbb{R}$.*

*Proof.* Let $\kappa_\phi$ be the canonical Kronecker flattening map in (4.29), then

$$H := \kappa_\phi(\mathcal{H}) = \sum\nolimits_{j=1}^{r} \lambda_j U_j (u_j^m \boxtimes \overline{u_j^m})^T = \sum\nolimits_{j=1}^{r} \lambda_j U_j (\overline{u_j^m} \boxtimes u_j^m)^T,$$

where $U_j$ denotes the $j$th column of $U$. The second equality holds, since $\mathcal{H}$ is $\mathbb{R}$-Hermitian decomposable. Thus, $\sum_{j=1}^{r} \lambda_j U_j (u_j^m \boxtimes \overline{u_j^m} - \overline{u_j^m} \boxtimes u_j^m)^T = 0$.

- If $k = r$, then $\{U_1, \ldots, U_r\}$ is linearly independent, which implies $u_j^m \boxtimes \overline{u_j^m} - \overline{u_j^m} \boxtimes u_j^m = 0$ for all $j$. So $u_j^m \boxtimes \overline{u_j^m}$ is real. There exists $s_j^m \in \mathbb{R}^{n_m}$ such that $u_j^m \boxtimes \overline{u_j^m} = s_j^m \boxtimes s_j^m$. It gives a desired decomposition as in (4.22).

- If $k = 1$, then there exists $\alpha_j \in \mathbb{R}$ such that $U_j = \alpha_j U_1$ for $1 \leq j \leq r$. Thus

$$H = U_1 V_1^T = U_1 \overline{V_1}^T \quad \text{where} \quad V_1 := \sum\nolimits_{j=1}^{r} \alpha_j \lambda_j u_j^m \boxtimes \overline{u_j^m}.$$

  Since $U_1(V_1 - \overline{V_1})^T = 0$, $V_1$ is the vectorization of a real symmetric matrix, then there exist $s_j^m \in \mathbb{R}^{n_m}$ and $\beta_j \in \mathbb{R}$ such that $V_1 = \sum_{j=1}^{r} \beta_j s_j^m \boxtimes s_j^m$. It also gives a desired decomposition as in (4.22).

- If $k = 2$, we can generally assume that $U_1, U_p$ are linearly independent. For each $i \notin \{1, p\}$, $U_i$ is a linear combination of $U_1, U_p$. Since each $U_i$ is the vectorization of a rank-1 Hermitian matrix, $U_i$ must be a multiple of $U_1$ or $U_p$, say, $U_i = U_1$ for $1 \leq i \leq p-1$ and $U_i = U_p$ for $p \leq i \leq r$, up to scaling of $\lambda_i$. Thus,

$$H = U_1 X_1^T + U_p X_2^T = U_1 \overline{X_1}^T + U_p \overline{X_2}^T,$$

  where $X_1 := \sum_{i=1}^{p-1} \lambda_i u_i^m \boxtimes \overline{u_i^m}$, $X_2 := \sum_{j=p}^{r} \lambda_j u_j^m \boxtimes \overline{u_j^m}$. Since $U_1(X_1 - \overline{X_1})^T + U_p(X_2 - \overline{X_2})^T = 0$, we have $X_1 = \overline{X_1}$ and $X_2 = \overline{X_2}$, so $X_1, X_2$ are vectorizations of real symmetric matrices. There exist $s_j^m \in \mathbb{R}^{n_m}, \beta_j \in \mathbb{R}$ such that $X_1 = \sum_{i=1}^{p-1} \beta_i s_i^m \boxtimes \overline{s_i^m}$, $X_2 = \sum_{j=p}^{r} \beta_j s_j^m \boxtimes \overline{s_j^m}$. This also gives a desired decomposition as in (4.22).

For every case of $k = 1, 2, r$, we get a decomposition like (4.22). □

Based on the above lemma, we can get the following conclusion.

**Proposition 4.14** ( [110]). *For* $\mathcal{H} \in \mathbb{R}_D^{[n_1,\ldots,n_m]}$, *if* $\operatorname{hrank}(\mathcal{H}) \leq 3$, *then* $\operatorname{hrank}(\mathcal{H}) = \operatorname{hrank}_{\mathbb{R}}(\mathcal{H})$. *Furthermore, if* $\operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) \leq 4$, *then* $\operatorname{hrank}(\mathcal{H}) = \operatorname{hrank}_{\mathbb{R}}(\mathcal{H})$.

*Proof.* Let $r := \operatorname{hrank}(\mathcal{H})$. We consider $r > 0$ (the case $r = 0$ is trivial). If $r \leq 3$, we can apply Lemma 4.13 to $\mathcal{H}$. Note that $k := \operatorname{rank} U \in \{1, 2, r\}$, since $r \leq 3$. For each $i = 1, \ldots, m$, the set $\{u_j^i\}_{j=1}^r$ can be changed to a set of real vectors while the length of decomposition does not change. As a result, we get a $\mathbb{R}$-Hermitian decomposition for $\mathcal{H}$ with length $r$, so $\operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) = \operatorname{hrank}(\mathcal{H})$.

If $\operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) \leq 4$, then $\operatorname{hrank}(\mathcal{H}) \leq 4$. If $\operatorname{hrank}(\mathcal{H}) \leq 3$, then the previous argument proves $\operatorname{hrank}(\mathcal{H}) = \operatorname{hrank}_{\mathbb{R}}(\mathcal{H})$. If $\operatorname{hrank}(\mathcal{H}) = 4$, then $\operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) \geq 4$, and hence $\operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) = \operatorname{hrank}(\mathcal{H}) = 4$. □

## 4.4 Matrix flattenings

All classical matrix flattenings are applicable to Hermitian tensors. In particular, Hermitian and Kronecker flattenings are special for Hermitian tensors.

### Hermitian flattening

Define the linear map $\mathfrak{m} : \mathbb{C}^{[n_1,\dots,n_m]} \to \mathbb{M}^N$ ($N = n_1 \cdots n_m$) such that for all $v_i \in \mathbb{C}^{n_i}$, $i = 1, \dots, m$,

$$\mathfrak{m}\big([v_1, v_2, \dots, v_m]_{\otimes h}\big) \;=\; (v_1 v_1^*) \boxtimes (v_2 v_2^*) \boxtimes \cdots \boxtimes (v_m v_m^*), \tag{4.23}$$

where $\boxtimes$ denotes the classical Kronecker product. The map $\mathfrak{m}$ is a bijection between $\mathbb{C}^{[n_1,\dots,n_m]}$ and $\mathbb{M}^N \cong \mathbb{M}^{n_1} \otimes \cdots \otimes \mathbb{M}^{n_m}$. The Hermitian decomposition $\mathcal{H} = \sum_{i=1}^r \lambda_i [u_i^1, \dots, u_i^m]_{\otimes h}$ is equivalent to that

$$\left\{ \begin{array}{rcl} \mathfrak{m}(\mathcal{H}) & = & \sum_{i=1}^r \lambda_i \, (u_i^1 (u_i^1)^*) \boxtimes \cdots \boxtimes (u_i^m (u_i^m)^*) \\ & = & \sum_{i=1}^r \lambda_i \, (u_i^1 \boxtimes \cdots \boxtimes u_i^m)(u_i^1 \boxtimes \cdots \boxtimes u_i^m)^*. \end{array} \right. \tag{4.24}$$

The matrix $H := \mathfrak{m}(\mathcal{H})$ is called the *Hermitian flattening matrix* of $\mathcal{H}$. It can be labelled by $I = (i_1, \dots, i_m)$ and $J = (j_1, \dots, j_m)$ such that

$$(H)_{IJ} = \mathcal{H}_{i_1 \dots i_m j_1 \dots j_m}. \tag{4.25}$$

The following is a basic result about flattening and ranks.

**Lemma 4.15** ( [110]). *If $H = \mathfrak{m}(\mathcal{H})$, then* $\operatorname{hrank}(\mathcal{H}) \geq \operatorname{hbrank}(\mathcal{H}) \geq$ *rank*$(H)$.

*Proof.* The first inequality is obvious. We prove the second one. Let $r := \operatorname{hbrank}(\mathcal{H})$, then there is a sequence $\{\mathcal{H}_k\} \subseteq \mathbb{C}^{[n_1,\dots,n_m]}$ such that $\mathcal{H}_k \to \mathcal{H}$ and $\operatorname{hrank} \mathcal{H}_k = r$. Let $H_k := \mathfrak{m}(\mathcal{H}_k)$, then $H_k \to H$ and $\operatorname{rank} H_k \leq r$, so $\operatorname{rank}(H) \leq r$. □

It is possible that $\operatorname{hrank}(\mathcal{H}) > \operatorname{rank}(H)$. For instance, consider the basis tensor $\mathcal{E}^{(11)(22)}(1)$. Its Hermitian flattening matrix has rank 2 while the Hermitian rank is 4 (see Example 4.4).

For each $\mathcal{H} \in \mathbb{R}_D^{[2,2]}$, its Hermitian flattening matrix is in the form

$$\mathfrak{m}(\mathcal{H}) = \begin{pmatrix} A & C \\ C & B \end{pmatrix}, \quad \text{where} \quad A, B, C \in \mathcal{S}^2. \tag{4.26}$$

**Proposition 4.16** ( [110]). *For each $\mathcal{H} \in \mathbb{R}_D^{[2,2]}$ as above, there exist invertible matrices $P, Q \in \mathbb{R}^{2\times 2}$ such that $\tilde{\mathcal{H}} := (P, Q) \times_{cong} \mathcal{H}$ has the flattening*

$$\mathfrak{m}(\tilde{\mathcal{H}}) = \begin{pmatrix} sI_2 & D \\ D & s\tilde{B} \end{pmatrix} - s \begin{pmatrix} uu^T & 0 \\ 0 & 0 \end{pmatrix}, \tag{4.27}$$

*where $s \in \{0, 1, -1\}$, $D$ is real diagonal, $u \in \mathbb{R}^2$ and $\tilde{B} \in \mathcal{S}^2$ . In particular, $u = 0$ if one of $A, B$ is positive (or negative) definite, and $s = 0$ if $A = B = 0$.*

*Proof.* Case I: Assume one of $A, B$ is nonzero, say, $A \neq 0$. If $A$ is not negative semidefinite, there is $v \in \mathbb{R}^2$ such that $A + vv^T \succ 0$. Then there is $U \in \mathbb{R}^{2\times 2}$ such that $U(A + vv^T)U^T = I_2$. There exists an orthogonal matrix $V$ such that $D := V(UCU^T)V^T$ is diagonal. Let $\tilde{\mathcal{H}} := (I_2, VU) \times_{cong} \mathcal{H}$, then

$$\begin{aligned} \mathfrak{m}(\tilde{\mathcal{H}}) &= \begin{pmatrix} V(U(A + vv^T)U^T)V^T & V(UCU^T)V^T \\ V(UCU^T)V^T & V(UBU^T)V^T \end{pmatrix} - \begin{pmatrix} V(Uvv^TU^T)V^T & 0 \\ 0 & 0 \end{pmatrix} \\ &= \begin{pmatrix} sI_2 & D \\ D & s\tilde{B} \end{pmatrix} - s \begin{pmatrix} uu^T & 0 \\ 0 & 0 \end{pmatrix}. \end{aligned}$$

So, the decomposition 4.27 holds for $s = 1, \tilde{B} := V(UBU^T)V^T, u := VUv$. If $A$ is negative semidefinite, then $-A$ is not negative semidefinite. We do the same thing for $-\mathcal{H}$ and can get 4.27 with $s = -1$. In particular, if either $A$ or $B$ is positive (or negative) definite, we can choose $v = 0$ and thus $u = VUv = 0$.

Case II: Assume $A = B = 0$. Since $C$ is real symmetric, there exists a matrix $U$ such that $D := UCU^T$ is diagonal. Let $\tilde{\mathcal{H}} := (I_2, U) \times_{cong} \mathcal{H}$, then

$$\tilde{\mathcal{H}} = \begin{pmatrix} 0 & UCU^T \\ UCU^T & 0 \end{pmatrix} = \begin{pmatrix} 0 & D \\ D & 0 \end{pmatrix}.$$

For this case, $s = 0$. □

Suppose the diagonal matrix $D$ in (4.27) is $D = \mathrm{diag}(d_1, d_2)$. When $s = 0$, the tensor $\tilde{\mathcal{H}}$ has the Hermitian decomposition:

$$\frac{1}{2}d_1 \left( [\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}]_{\otimes h} - [\begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}]_{\otimes h} \right) + \frac{1}{2}d_2 \left( [\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}]_{\otimes h} - [\begin{pmatrix} 1 \\ -1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}]_{\otimes h} \right).$$

Thus, $\operatorname{hrank}_{\mathbb{R}}(\tilde{\mathcal{H}}) \leq 4$. When $s = 1$ or $-1$, let $E := s\tilde{B} - s \cdot \operatorname{diag}(d_1^2, d_2^2)$. Suppose $E = \lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T$ is an orthogonal eigenvalue decomposition. Then, (note that $s^2 = 1$),

$$\mathfrak{m}(\tilde{\mathcal{H}}) = s\begin{pmatrix} 1 & sd_1 \\ sd_1 & s^2d_1^2 \end{pmatrix} \boxtimes \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} + s\begin{pmatrix} 1 & sd_2 \\ sd_2 & s^2d_2^2 \end{pmatrix} \boxtimes \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix} \boxtimes (\lambda_1 v_1 v_1^T + \lambda_2 v_2 v_2^T) - s\begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} \boxtimes (uu^T).$$

The above gives the real Hermitian decomposition for $\tilde{\mathcal{H}}$:

$$\tilde{\mathcal{H}} = s[\begin{pmatrix} 1 \\ sd_1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}]_{\otimes h} + s[\begin{pmatrix} 1 \\ sd_2 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}]_{\otimes h} + \lambda_1[\begin{pmatrix} 0 \\ 1 \end{pmatrix}, v_1]_{\otimes h} + \lambda_2[\begin{pmatrix} 0 \\ 1 \end{pmatrix}, v_2]_{\otimes h} - s[\begin{pmatrix} 1 \\ 0 \end{pmatrix}, u]_{\otimes h}.$$

For all cases, we have $\operatorname{hrank}_{\mathbb{R}}(\tilde{\mathcal{H}}) \leq 5$. Since $\tilde{\mathcal{H}} = (P, Q) \times_{cong} \mathcal{H}$ and $P, Q$ are invertible, $\operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) = \operatorname{hrank}_{\mathbb{R}}(\tilde{\mathcal{H}})$. Therefore, we get the following conclusion.

**Theorem 4.17** ( [110]). *For every $\mathcal{H} \in \mathbb{R}_D^{[2,2]}$, with the flattening as in (4.26), we have* $\operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) \leq 5$. *In particular, we have* $\operatorname{hrank}(\mathcal{H}) = \operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) \leq 4$ *if one of $A, B$ is positive (or negative) definite, or if $A = B = 0$.*

*Proof.* The inequality $\operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) \leq 5$ is implied by

$$\operatorname{hrank}_{\mathbb{R}}(\tilde{\mathcal{H}}) \leq 5, \qquad \operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) = \operatorname{hrank}_{\mathbb{R}}(\tilde{\mathcal{H}}).$$

If one of $A, B$ is positive (or negative) definite, then $u = 0$ by Proposition 4.16 and hence $\operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) = \operatorname{hrank}_{\mathbb{R}}(\tilde{\mathcal{H}}) \leq 4$. If $A = B = 0$, we already have $\operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) = \operatorname{hrank}_{\mathbb{R}}(\tilde{\mathcal{H}}) \leq 4$. By Proposition 4.14, $\operatorname{hrank}(\mathcal{H}) = \operatorname{hrank}_{\mathbb{R}}(\mathcal{H}) \leq 4$ if one of $A, B$ is positive (or negative) definite, or if $A = B = 0$. □

## Kronecker flattening

Every matrix flattening map $\phi$ on the tensor space $\mathbb{C}^{n_1 \times \cdots \times n_m}$ can be used to define a new flattening map $\kappa_\phi$ on $\mathbb{C}^{[n_1,\ldots,n_m]}$. Suppose $\phi$ flattens tensors in $\mathbb{C}^{n_1 \times \cdots \times n_m}$ to matrices of the size $D_1$-by-$D_2$. Then we can define the linear map $\kappa_\phi : \mathbb{C}^{[n_1,\ldots,n_m]} \to \mathbb{C}^{D_1^2 \times D_2^2}$ such that

$$\kappa_\phi\big([u_1, \ldots, u_m]_{\otimes h}\big) = \phi(u_1 \otimes \cdots \otimes u_m) \boxtimes \overline{\phi(u_1 \otimes \cdots \otimes u_m)} \tag{4.28}$$

for all $u_i \in \mathbb{C}^{n_i}$. The map $\kappa_\phi$ is called the *$\phi$-Kronecker* flattening generated by $\phi$. When $\phi$ is the standard flattening such that $\phi(a_1 \otimes \cdots a_{m-1} \otimes a_m) = (a_1 \boxtimes \cdots a_{m-1})(a_m)^T$, then $\kappa_\phi$ is the linear map such that

$$\kappa_\phi\Big(\sum\nolimits_i \lambda_i [u_i^1, \ldots, u_i^m]_{\otimes h}\Big) = \sum\nolimits_i \lambda_i Z_i \boxtimes \overline{Z_i} \tag{4.29}$$

where $Z_i := (u_i^1 \boxtimes \cdots \boxtimes u_i^{m-1})(u_i^m)^T$. The map $\kappa_\phi$ in (4.29) is called the *canonical Kronecker flattening*.

**Lemma 4.18** ( [110]). *Let $\phi$ be a flattening map on $\mathbb{C}^{n_1 \times \cdots n_m}$ and $\kappa_\phi$ be the corresponding $\phi$-Kronecker flattening. Then, for each $\mathcal{H} \in \mathbb{C}^{[n_1,\ldots,n_m]}$,*

$$\mathrm{hrank}(\mathcal{H}) \geq \mathrm{hbrank}(\mathcal{H}) \geq \mathit{rank}\kappa_\phi(\mathcal{H}). \tag{4.30}$$

The above is an analogue of Lemma 4.15. We omit its proof for cleanness of the paper. The Hermitian and Kronecker flattening may give different lower bounds for Hermitian ranks, as shown below.

**Example 4.19.** *For $m = 2$ and $n > 1$, consider the Hermitian tensor in $\mathbb{R}^{[n,n]}$*

$$\mathcal{H} = \sum\nolimits_{i,j=1}^{n} e_i \otimes e_i \otimes e_j \otimes e_j = \big(\sum\nolimits_{i=1}^{n} e_i \otimes e_i\big) \otimes \big(\sum\nolimits_{i=1}^{n} e_i \otimes e_i\big).$$

*Let $\kappa_\phi$ be the canonical Kronecker flattening as in (4.29), then*

$$\mathfrak{m}(\mathcal{H}) = \big(\sum\nolimits_{i=1}^{n} e_i \boxtimes e_i\big)\big(\sum\nolimits_{i=1}^{n} e_i \boxtimes e_i\big)^T,\ \kappa_\phi(\mathcal{H}) = \big(\sum\nolimits_{i=1}^{n} e_i e_i^T\big) \boxtimes \big(\sum\nolimits_{i=1}^{n} e_i e_i^T\big) = I_{n^2}.$$

*For instance, when $n = 2$, we have*

$$\mathfrak{m}(\mathcal{H}) = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix},\ \kappa_\phi(\mathcal{H}) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

*By Lemma 4.18,* $\mathrm{hrank}(\mathcal{H}) \geq \mathit{rank}\kappa_\phi(\mathcal{H}) = n^2$ *while* $\mathit{rank}\,\mathfrak{m}(\mathcal{H}) = 1$ *. Indeed, we further have a sharper lower bound*

$$\mathrm{hrank}(\mathcal{H}) \geq n^2 + 1.$$

*Suppose otherwise that* $\mathrm{hrank}(\mathcal{H}) = n^2$*, say, $\mathcal{H} = \sum_{i=1}^{n^2} \lambda_i [u_i, v_i]_{\otimes h}$ for $\lambda_i \in \mathbb{R}$ and $u_i, v_i \in \mathbb{C}^n$, then*

$$\kappa_\phi(\mathcal{H}) = I_{n^2} = \sum\nolimits_{i=1}^{n^2} \lambda_i (u_i \cdot v_i^T) \boxtimes (\overline{u_i} \cdot \overline{v_i}^T) = \sum\nolimits_{i=1}^{n^2} \lambda_i (u_i \boxtimes \overline{u_i})(v_i \boxtimes \overline{v_i})^T.$$

*Let*

$$U = [\lambda_1 u_1 \boxtimes \overline{u_1}, \ldots, \lambda_{n^2} u_{n^2} \boxtimes \overline{u_{n^2}}], \quad V = [v_1 \boxtimes \overline{v_1}, \ldots, v_{n^2} \boxtimes \overline{v_{n^2}}].$$

*Then $U, V$ are square matrices of length $n^2$ and*

$$UV^T = I_{n^2} \Rightarrow V^T U = I_{n^2} \Rightarrow \lambda_j (v_i \boxtimes \overline{v_i})^T (u_j \boxtimes \overline{u_j}) = \begin{cases} 1 & i = j, \\ 0 & i \neq j. \end{cases}$$

*For $i \neq j$, we have*

$$(v_i \boxtimes \overline{v_i})^T (u_j \boxtimes \overline{u_j}) = (v_i^T u_j) \boxtimes (\overline{v_i}^T \overline{u_j}) = |v_i^T u_j|^2 = 0 \Rightarrow v_i^T u_j = 0.$$

*Thus, $u_2, \ldots, u_{n^2} \in v_1^\perp$ and*

$$r := \dim(span\{u_2, \ldots, u_{n^2}\}) \leq n - 1.$$

*Let $\{s_1, \ldots, s_r\}$ be a basis for $span\{u_2, \ldots, u_{n^2}\}$. For each $i = 2, 3, \ldots n^2$, $u_i \boxtimes \overline{u}_i$ belongs to the span of the set $\{s_p \boxtimes \overline{s}_q\}_{1 \leq p,q \leq r}$, so*

$$\dim\left(span\{u_i \boxtimes \overline{u}_i\}_{i=2}^{n^2}\right) \leq \dim\left(span\{s_p \boxtimes \overline{s}_q\}_{1 \leq p,q \leq r}\right) = r^2.$$

*This implies that*

$$n^2 = rank(U) \leq 1 + \dim\left(span\{u_i \boxtimes \overline{u}_i\}_{i=2}^{n^2}\right) \leq r^2 + 1 \leq (n-1)^2 + 1.$$

*However, $n^2 > (n-1)^2 + 1$ when $n \geq 2$. This is a contradiction, so* $\operatorname{hrank}(\mathcal{H}) \geq n^2 + 1$. *For the case $n = 2$,* $\operatorname{hrank}(\mathcal{H}) = n^2 + 1$, *because we have a Hermitian decomposition of length* 5 *(in the following $c := \sqrt{1 + \sqrt{2}}$):*

$$\frac{1}{2c^4 - 2}\left( \left[ \begin{pmatrix} c \\ 1 \end{pmatrix}, \begin{pmatrix} c \\ 1 \end{pmatrix} \right]_{\otimes h} + \left[ \begin{pmatrix} c \\ -1 \end{pmatrix}, \begin{pmatrix} c \\ -1 \end{pmatrix} \right]_{\otimes h} - \left[ \begin{pmatrix} 1 \\ c\sqrt{-1} \end{pmatrix}, \begin{pmatrix} 1 \\ c\sqrt{-1} \end{pmatrix} \right]_{\otimes h} \right.$$
$$\left. - \left[ \begin{pmatrix} 1 \\ -c\sqrt{-1} \end{pmatrix}, \begin{pmatrix} 1 \\ -c\sqrt{-1} \end{pmatrix} \right]_{\otimes h} \right) + 2 \left[ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right]_{\otimes h}.$$

*When $n > 2$, the true value of* $\operatorname{hrank}(\mathcal{H})$ *is not known to the authors.*

## 4.5 PSD Hermitian tensors

A Hermitian tensor $\mathcal{H}$ can be uniquely determined by the multi-quadratic conjugate polynomial $\mathcal{H}(x, \overline{x}) := \langle \mathcal{H}, [x_1, \ldots, x_m]_{\otimes_h} \rangle$, in the tuple $x := (x_1, \ldots, x_m)$ of complex vector variables $x_i \in \mathbb{C}^{n_i}$. Like the matrix case, positive semidefinite Hermitian tensors can be naturally defined [101].

**Definition 4.20.** *Let $\mathbb{F} = \mathbb{C}$ or $\mathbb{R}$. A Hermitian tensor $\mathcal{H} \in \mathbb{F}^{[n_1,\ldots,n_m]}$ is called $\mathbb{F}$-positive semidefinite ($\mathbb{F}$-psd) if $\mathcal{H}(x,\overline{x}) \geq 0$ for all $x_i \in \mathbb{F}^{n_i}$. Moreover, if $\mathcal{H}(x,\overline{x}) > 0$ for all $0 \neq x_i \in \mathbb{F}^{n_i}$, then $\mathcal{H}$ is called $\mathbb{F}$-positive definite ($\mathbb{F}$-pd).*

For convenience, a complex (resp., real) Hermitian tensor is called psd if it is $\mathbb{C}$-psd (resp., $\mathbb{R}$-psd). Denote the cone of $\mathbb{F}$-psd Hermitian tensors

$$\mathscr{P}_{\mathbb{F}}^{[n_1,\ldots,n_m]} := \left\{\mathcal{H} \in \mathbb{F}^{[n_1,\ldots,n_m]} : \mathcal{H}(x,\overline{x}) \geq 0 \,\forall\, x_i \in \mathbb{F}^{n_i}\right\}. \tag{4.31}$$

**Example 4.21.** *(i) Consider $\mathcal{H} \in \mathbb{C}^{[3,3]}$ such that $\mathcal{H}(x,y) = \langle \mathcal{H}, [x,y]_{\otimes h}\rangle$ is the following conjugate polynomial (for cleanness of display, the variable $x_1$ is changed to $x := (x_1,x_2,x_3)$ and $x_2$ is changed to $y := (y_1,y_2,y_3)$):*

$$\begin{aligned}&|x_1|^2|y_1|^2 + |x_2|^2|y_2|^2 + |x_3|^2|y_3|^2 + 2(|x_1|^2|y_2|^2 + |x_2|^2|y_3|^2 + |x_3|^2|y_1|^2)\\ &\quad -(x_1\overline{x_2}y_1\overline{y_2} + \overline{x_1}x_2\overline{y_1}y_2 + x_1\overline{x_3}y_1\overline{y_3} + \overline{x_1}x_3\overline{y_1}y_3 + x_2\overline{x_3}y_2\overline{y_3} + \overline{x_2}x_3\overline{y_2}y_3).\end{aligned}$$

*Since $\mathcal{H}(x,y) \geq 0$ for all real $x, y$ (see [112]), the tensor $\mathcal{H}$ is $\mathbb{R}$-psd. In fact, it is also $\mathbb{C}$-psd, because*

$$\begin{aligned}&\mathcal{H}(x,y)\\ =\;& |x_1|^2|y_1|^2 + |x_2|^2|y_2|^2 + |x_3|^2|y_3|^2 + 2(|x_1|^2|y_2|^2 + |x_2|^2|y_3|^2 + |x_3|^2|y_1|^2)\\ &-2\big(Re(x_1\overline{x_2}y_1\overline{y_2}) + Re(x_1\overline{x_3}y_1\overline{y_3}) + Re(x_2\overline{x_3}y_2\overline{y_3})\big)\\ \geq\;& |x_1|^2|y_1|^2 + |x_2|^2|y_2|^2 + |x_3|^2|y_3|^2 + 2(|x_1|^2|y_2|^2 + |x_2|^2|y_3|^2 + |x_3|^2|y_1|^2))\\ &-2(|x_1x_2y_1y_2| + |x_1x_3y_1y_3| + |x_2x_3y_2y_3|)\\ =\;& \mathcal{H}(\hat{x},\hat{y}) \geq 0,\end{aligned}$$

*where $\hat{x} := (|x_1|,|x_2|,|x_3|)$ and $\hat{y} := (|y_1|,|y_2|,|y_3|)$ are real.*
*(ii) Consider $\mathcal{H} \in \mathbb{C}^{[2,2]}$ such that*

$$\mathcal{H}_{1111} = \mathcal{H}_{1122} = \mathcal{H}_{2211} = 1, \quad \mathcal{H}_{1221} = \mathcal{H}_{2112} = -1$$

*and all other entries are zeros, so (for cleanness, the variable $x_1$ is changed to $x := (x_1,x_2)$ and $x_2$ is changed to $y := (y_1,y_2)$):*

$$\mathcal{H}(x,y) = |x_1|^2|y_1|^2 + x_1\overline{x}_2y_1\overline{y}_2 + \overline{x}_1x_2\overline{y}_1y_2 - x_1\overline{x}_2\overline{y}_1y_2 - \overline{x}_1x_2y_1\overline{y}_2.$$

*When $x, y$ are real, $\mathcal{H}(x,y) = x_1^2y_1^2 \geq 0$. This tensor is $\mathbb{R}$-psd but not $\mathbb{C}$-psd, because for $x = y = (\sqrt{-1},1)$, $\mathcal{H}(x,y) = 1-1-1-1-1 = -3 < 0$.*

A $\mathbb{R}$-psd Hermitian tensor is not necessarily $\mathbb{C}$-psd. However, they are equivalent for $\mathbb{R}$-Hermitian decomposable tensors.

**Proposition 4.22** ( [110]). *For $\mathcal{H} \in \mathbb{R}_D^{[n_1,\ldots,n_m]}$, $\mathcal{H}$ is $\mathbb{R}$-psd if and only if $\mathcal{H}$ is $\mathbb{C}$-psd.*

*Proof.* The "if" direction is obvious. We prove the "only if" direction. For $v^i \in \mathbb{C}^{n_i}$, write $v^j = x^j + \sqrt{-1}y^j$ with $x^j, y^j \in \mathbb{R}^{n_j}$. Then, we have

$$\begin{aligned} \langle [u^1,\ldots,u^m]_{\otimes h}, [v^1,\ldots,v^m]_{\otimes h}\rangle &= \Pi_{j=1}^m (u^j)^T v^j \cdot (u^j)^T \overline{v}^j = \Pi_{j=1}^m |(u^j)^T v^j|^2 \\ = \Pi_{j=1}^m (|(u^j)^T x^j|^2 + |(u^j)^T y^j|^2) &= \sum_{z^j \in \{x^j, y^j\}} \langle [u^1,\ldots,u^m]_{\otimes h}, [z^1,\ldots,z^m]_{\otimes h}\rangle. \end{aligned}$$

Since $\mathcal{H} \in \mathbb{R}_D^{[n_1,\ldots,n_m]}$, it is a sum of real rank-1 real Hermitian tensors, so

$$\left\langle \mathcal{H}, [v^1,\ldots,v^m]_{\otimes h}\right\rangle = \sum_{z^j \in \{x^j, y^j\}} \left\langle \mathcal{H}, [z^1,\ldots,z^m]_{\otimes h}\right\rangle \geq 0.$$

If $\mathcal{H}$ is $\mathbb{R}$-psd, then $\mathcal{H}$ is also $\mathbb{C}$-psd. □

Clearly, $\mathscr{P}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$ is a closed convex cone. As in [14], a cone is said to be *solid* if it has nonempty interior; it is said to be *pointed* if it does not contain any line through origin; a closed convex cone is said to be *proper* if it is both solid and pointed. The complex cone $\mathscr{P}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$ is proper, as mentioned in [101]. However, the real cone $\mathscr{P}_{\mathbb{R}}^{[n_1,\ldots,n_m]}$ is not proper. In fact, it is solid but not pointed.

**Proposition 4.23** ( [110]). *For $m > 1$ and $n_1,\ldots,n_m > 1$, the cone $\mathscr{P}_{\mathbb{C}}^{n_1,\ldots,n_m}$ is proper, while $\mathscr{P}_{\mathbb{R}}^{[n_1,\ldots,n_m]}$ is solid but not pointed.*

*Proof.* Let $\mathcal{I} \in \mathbb{F}^{[n_1,\ldots,n_m]}$ be the identity tensor, i.e., $\mathcal{I}(x,\overline{x}) = (x_1^* x_1)\cdots(x_m^* x_m)$. The conjugate polynomial $\mathcal{I}(x,\overline{x})$ is positive definite on the spheres $\|x_i\| = 1$. Thus, for $\epsilon > 0$ sufficiently small, all Hermitian tensors $\mathcal{H} \in \mathbb{F}^{[n_1,\ldots,n_m]}$ with $\|\mathcal{H} - \mathcal{I}\| < \epsilon$ belong to the cone $\mathscr{P}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$, for both $\mathbb{F} = \mathbb{C}, \mathbb{R}$. That is, $\mathcal{I}$ is an interior point, and hence $\mathscr{P}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$ is solid.

The complex cone $\mathscr{P}_{\mathbb{C}}^{n_1,\ldots,n_m}$ is pointed. For each $\mathcal{H} \in \mathscr{P}_{\mathbb{C}}^{n_1,\ldots,n_m} \cap -\mathscr{P}_{\mathbb{C}}^{n_1,\ldots,n_m}$, $\mathcal{H}(x,\overline{x})$ must be identically zero for all complex $x_i$. The conjugate polynomial

$$\mathcal{H}(x,\overline{x}) = \sum\nolimits_{i_1\ldots i_m j_1\ldots j_m} \mathcal{H}_{i_1\ldots i_m j_1\ldots j_m} x_{1,i_1}\cdots x_{m,i_m}\overline{x}_{1,j_1}\cdots\overline{x}_{m,j_m}$$

is identically zero if and only all its coefficients are zero, i.e., $\mathcal{H} = 0$. This implies that $\mathscr{P}_{\mathbb{C}}^{n_1,\ldots,n_m}$ does not contain any line through origin, i.e., it is pointed.

The real cone $\mathscr{P}_{\mathbb{R}}^{n_1,\ldots,n_m}$ is not pointed. For $m > 1$ and $n_1, \ldots, n_m > 1$, the set $\mathbb{R}_D^{[n_1,\ldots,n_m]}$ is a proper subspace of $\mathbb{R}^{[n_1,\ldots,n_m]}$. Let $C$ be the orthogonal complement of $\mathbb{R}_D^{[n_1,\ldots,n_m]}$ in $\mathbb{R}^{[n_1,\ldots,n_m]}$. Then, for all $0 \neq \mathcal{X} \in C$ and for all $x_j \in \mathbb{R}^{n_j}$, $\langle \mathcal{X}, [x_1, \ldots, x_m]_{\otimes h} \rangle = 0$ because $[x_1, \ldots, x_m]_{\otimes h} \in \mathbb{R}_D^{[n_1,\ldots,n_m]}$. This implies $C \subseteq \mathscr{P}_{\mathbb{R}}^{n_1,\ldots,n_m}$. So, $\mathscr{P}_{\mathbb{R}}^{n_1,\ldots,n_m}$ contains a line through the origin and hence it is not pointed. □

## 4.6 Separable Hermitian tensors

A basic topic in quantum physics is tensor entanglement. It requires to decide whether or not a given Hermitian tensor can be written as a sum of rank-1 Hermitian tensors with positive coefficients. This leads to the concept of *separable* tensors.

**Definition 4.24.** *[101] A Hermitian tensor $\mathcal{H} \in \mathbb{C}^{[n_1,\ldots,n_m]}$ is called separable if*

$$\mathcal{H} = [u_1^1, \ldots, u_1^m]_{\otimes h} + \cdots + [u_r^1, \ldots, u_r^m]_{\otimes h} \tag{4.32}$$

*for some vectors $u_i^j \in \mathbb{C}^{n_j}$. When such decomposition exists, (4.32) is called a positive $\mathbb{C}$-Hermitian decomposition and $\mathcal{H}$ is called $\mathbb{C}$-separable. Moreover, if each $u_i^j$ in (4.32) is real, then $\mathcal{H}$ is called $\mathbb{R}$-separable and (4.32) is called a positive $\mathbb{R}$-Hermitian decomposition.*

Let $\mathbb{F} = \mathbb{C}$ or $\mathbb{R}$. The set of $\mathbb{F}$-separable tensors in $\mathbb{F}^{[n_1,\ldots,n_m]}$ is denoted as $\mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$. The decomposition (4.32) is equivalent to that

$$\mathcal{H}(x, \overline{x}) = \sum\nolimits_{i=1}^{r} |(u_i^1)^* x_1|^2 \cdots |(u_i^m)^* x_m|^2.$$

All $\mathbb{F}$-separable tensors must be HSOS. To be $\mathbb{R}$-separable, a tensor must be $\mathbb{R}$-Hermitian decomposable. The following is the relationship between $\mathbb{C}$-separability and $\mathbb{R}$-separability.

**Lemma 4.25** ( [110])**.** *For $\mathcal{H} \in \mathbb{R}_D^{[n_1,\ldots,n_m]}$, $\mathcal{H}$ is $\mathbb{R}$-separable if and only if it is $\mathbb{C}$-separable.*

*Proof.* The "only if" direction is obvious. We prove the "if" direction. Assume $\mathcal{H}$ is $\mathbb{C}$-separable, then (4.32) holds for some complex vectors $u_i^j$. Let $s_i^j := \mathrm{Re}(u_i^j)$ and $t_i^j := \mathrm{Im}(u_i^j)$. For all *real* vector variables $x_i \in \mathbb{R}^{n_i}$, the inner product $\langle [u_i^1, \ldots, u_i^m]_{\otimes h}, [x_1, \ldots, x_m]_{\otimes h} \rangle = \prod_{j=1}^m |(u_i^j)^* x^j|^2$, which can be expanded as

$$\prod_{j=1}^{m} \left( |(s_i^j)^T x_j|^2 + |(t_i^j)^T x_j|^2 \right) = \sum_{z_i^j \in \{s_i^j, t_i^j\}} \langle [z_i^1, \ldots, z_i^m]_{\otimes h}, [x_1, \ldots, x_m]_{\otimes h} \rangle.$$

The equation (4.32) implies that, for all real vectors $x_i$,

$$\langle \mathcal{H}, [x_1, \ldots, x_m]_{\otimes h} \rangle = \sum_{i=1}^{r} \sum_{z_i^j \in \{s_i^j, t_i^j\}} \langle [z_i^1, \ldots, z_i^m]_{\otimes h}, [x_1, \ldots, x_m]_{\otimes h} \rangle.$$

Since $\mathcal{H}$ is $\mathbb{R}$-separable, by Lemma 4.12, $\mathcal{H} = \sum_{i=1}^{r} \sum_{z_i^j \in \{s_i^j, t_i^j\}} [z_i^1, \ldots, z_i^m]_{\otimes h}$. Hence, $\mathcal{H}$ is also $\mathbb{R}$-separable. □

The complex separable tensor cone $\mathscr{S}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$ is dual to $\mathscr{P}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$, as noted in [101]. The duality also holds for the real case. Let $\mathbb{F} = \mathbb{C}$ or $\mathbb{R}$. By the definition (see [8]), the dual cone of $\mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$ is the set

$$\left(\mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]}\right)^{\star} := \left\{ X \in \mathbb{F}^{[n_1,\ldots,n_m]} : \langle X, Y \rangle \geq 0 \, \forall \, Y \in \mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]} \right\}.$$

Recall that a closed convex cone is proper if it is solid (has nonempty interior) and pointed (does not contain any line through the origin). The complex cone $\mathscr{S}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$ is proper [101], but $\mathscr{S}_{\mathbb{R}}^{[n_1,\ldots,n_m]}$ is not.

**Theorem 4.26** ( [110]). *For $\mathbb{F} = \mathbb{R}, \mathbb{C}$, the cone $\mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$ is dual to $\mathscr{P}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$, i.e.,*

$$\left(\mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]}\right)^{\star} = \mathscr{P}_{\mathbb{F}}^{[n_1,\ldots,n_m]}, \quad \left(\mathscr{P}_{\mathbb{F}}^{[n_1,\ldots,n_m]}\right)^{\star} = \mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]}. \tag{4.33}$$

*Moreover, the complex cone $\mathscr{S}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$ is proper, while the real one $\mathscr{S}_{\mathbb{R}}^{[n_1,\ldots,n_m]}$ is not proper. In fact, $\mathscr{S}_{\mathbb{R}}^{[n_1,\ldots,n_m]}$ is pointed but not solid.*

*Proof.* Observe that $\mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$ equals the conic hull of the compact set

$$U := \left( [u_1, \ldots, u_m]_{\otimes h} : u_i \in \mathbb{F}^{n_i}, \|u_i\| = 1 \right), \tag{4.34}$$

so it is a closed convex cone [8]. A tensor $X \in \mathbb{F}^{[n_1,\ldots,n_m]}$ belongs to the dual cone of $\mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$ if and only if $\langle X, [u_1, \ldots, u_m]_{\otimes h} \rangle \geq 0$ for all $u_i \in \mathbb{F}^{n_i}$, which is equivalent to that $X$ is $\mathbb{F}$-psd. Therefore, the dual cone of $\mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$ is $\mathscr{P}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$. Since $\mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$ and $\mathscr{P}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$ are both closed convex cones, the dual cone of $\mathscr{P}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$ is also equal to $\mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$, by the bi-duality theorem [8]. Hence, the dual relationship (4.33) holds. By Proposition 4.23, the cone $\mathscr{P}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$ is proper, while $\mathscr{P}_{\mathbb{R}}^{[n_1,\ldots,n_m]}$ is solid but not pointed. By the duality, $\mathscr{S}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$ is also proper, while $\mathscr{S}_{\mathbb{R}}^{[n_1,\ldots,n_m]}$ is pointed but not solid [2]. □

Theorem 4.26 tells that a Hermitian tensor is $\mathbb{F}$-separable if and only if it belongs to the dual cone of $\mathscr{P}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$. Therefore, for $\mathcal{A} \in \mathbb{F}^{[n_1,\ldots,n_m]}$, if there exists $\mathcal{B} \in \mathbb{F}^{[n_1,\ldots,n_m]}$ such

that $\mathcal{B}(x,\overline{x}) \in \Sigma[x,\overline{x}]$ and $\langle \mathcal{A}, \mathcal{B}\rangle < 0$, then $\mathcal{A}$ is not $\mathbb{F}$-separable. For instance, consider the Hankel tensor $\mathcal{A} \in \mathbb{C}^{[2,2]}$ such that $\mathcal{A}_{ijkl} = i+j+k+l$ for all $i,j,k,l$. Let $\mathcal{B}$ be the Hermitian tensor such that

$$\langle \mathcal{B}, [x_1, x_2]_{\otimes h}\rangle = |x_{11}x_{21} - \frac{5}{6}x_{11}x_{22}|^2.$$

Since $\mathcal{B}(x) \in \Sigma[x]$ and $\langle \mathcal{A}, \mathcal{B}\rangle = -\frac{1}{6} < 0$, $\mathcal{A}$ is not $\mathbb{F}$-separable for $\mathbb{F} = \mathbb{C}, \mathbb{R}$.

An important computational task is to determine whether or not a Hermitian tensor is separable. If it is, we need a positive Hermitian decomposition. This is an interesting future work.

Let $\mathbb{F} = \mathbb{C}, \mathbb{R}$. In the proof of Theorem 4.26, we have seen that the $\mathbb{F}$-separable Hermitian tensor cone $\mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$ equals the conic hull of the compact set $U$, that is, (cone denotes the conic hull)

$$\mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]} \;=\; \mathrm{cone}\Big([u_1,\ldots,u_r]_{\otimes h} : u_i \in \mathbb{F}^{n_i}, \|u_i\| = 1\Big). \tag{4.35}$$

Equivalently, we have $\mathcal{A} \in \mathscr{S}_{\mathbb{F}}^{[n_1,\ldots,n_m]}$ if and only if there exist positive scalars $\lambda_i > 0$ and unit length vectors $u_i^j \in \mathbb{F}^{n_j}$ such that

$$\mathcal{A} = \sum\nolimits_{i=1}^{r} \lambda_i [u_i^1, \ldots, u_i^m]_{\otimes h}. \tag{4.36}$$

If we let $\mu := \sum_{i=1}^r \lambda_i \delta_{(u_i^1,\ldots,u_i^m)}$ be the weighted sum of Dirac measures, then (4.36) is equivalent to

$$\mathcal{A} = \int [x_1,\ldots,x_m]_{\otimes h} \mathtt{d}\mu. \tag{4.37}$$

The support $supp(\mu)$ of the measure $\mu$ is contained in the multi-sphere

$$\mathbb{S}_{\mathbb{F}}^{n_1,\ldots,n_m} \;:=\; \{(x_1,\ldots,x_m) \in \mathbb{F}^{n_1} \times \cdots \mathbb{F}^{n_m} : \|x_1\| = \cdots \|x_m\| = 1\}.$$

Interestingly, if there is a Borel measure $\mu$ supported in $\mathbb{S}_{\mathbb{F}}^{n_1,\ldots,n_m}$, then there must exist $\lambda_i > 0$ and unit length vectors $u_i^j$ satisfying (4.36). This can be implied by the proof of Theorem 5.9 of [84]. Therefore, we have the following theorem.

**Theorem 4.27** ( [110]). *For $\mathbb{F} = \mathbb{C}$ or $\mathbb{R}$, a tensor $\mathcal{A} \in \mathbb{F}^{[n_1,\ldots,n_m]}$ is $\mathbb{F}$-separable if and only if there exists a Borel measure $\mu$ such that (4.37) holds and $supp(\mu) \subseteq \mathbb{S}_{\mathbb{F}}^{n_1,\ldots,n_m}$.*

The task of checking existence of $\mu$ in Theorem 4.27 is a truncated moment problem. We refer to [82, 84, 112, 113, 115] for related work. Interestingly, separable Hermitian tensors

can also be characterized by the Hermitian flattening map $\mathfrak{m}$. As in (4.24), the decomposition (4.36) is equivalent to that

$$\mathfrak{m}(\mathcal{A}) \;=\; \sum\nolimits_{i=1}^{r} \lambda_i \big(u_i^1 (u_i^1)^*\big) \boxtimes \cdots \boxtimes \big(u_i^m (u_i^m)^*\big). \tag{4.38}$$

The Theorem 4.27 immediately implies the following.

**Theorem 4.28** ( [110]). *For $\mathbb{F} = \mathbb{C}$ or $\mathbb{R}$, a tensor $\mathcal{A} \in \mathbb{F}^{[n_1,\ldots,n_m]}$ is $\mathbb{F}$-separable if and only if there exist Hermitian psd matrices $0 \preceq B_{ij} \in \mathbb{F}^{n_j \times n_j}$, for $i = 1, \ldots, s$ and $j = 1, \ldots, m$, such that*

$$\mathfrak{m}(\mathcal{A}) \;=\; \sum\nolimits_{i=1}^{s} B_{i1} \boxtimes \cdots \boxtimes B_{im}. \tag{4.39}$$

The smallest integer $s$ in (4.39) is called the $\mathbb{F}$-*psd rank* for the tensor $\mathcal{A}$. How to determine $\mathbb{F}$-psd ranks is mostly an open question.

**Example 4.29.** *Consider the tensor $\mathcal{A} \in \mathbb{C}^{[2,2]}$ with the Hermitian flattening*

$$\mathfrak{m}(\mathcal{A}) = \begin{pmatrix} 5 & -4 & 1 & -5 \\ -4 & 21 & -5 & 7 \\ 1 & -5 & 3 & -3 \\ -5 & 7 & -3 & 13 \end{pmatrix}.$$

*It is $\mathbb{R}$-separable, because*

$$\mathfrak{m}(\mathcal{A}) = \begin{pmatrix} 2 & -1 \\ -1 & 1 \end{pmatrix} \boxtimes \begin{pmatrix} 1 & 1 \\ 1 & 3 \end{pmatrix} + \begin{pmatrix} 3 & 2 \\ 2 & 2 \end{pmatrix} \boxtimes \begin{pmatrix} 1 & -2 \\ -2 & 5 \end{pmatrix}.$$

*The $\mathbb{R}$-psd rank is 2, since $\mathcal{A}$ does not have a decomposition like (4.39) for $s = 1$.*

## 4.7 Detecting separability

The separability of Hermitian tensors can be detected by solving moment optimization problems, which then can be solved by Lasserre type semidefinite relaxations. This is done by Li and Ni [87], based on the results in [104, 105]. In this section, we review this method and provide an improved formulation of the moment optimization. Furthermore, we prove stronger convergence results.

Recall that a Hermitian tensor $\mathcal{H} \in \mathbb{C}^{[n_1,\ldots,n_m]}$ is separable if and only if there exist vectors $u_i^j \in \mathbb{C}^{n_j}$ such that

$$\mathcal{H} \;=\; [u_1^1, \ldots, u_1^m]_{\otimes h} + \cdots + [u_r^1, \ldots, u_r^m]_{\otimes h}. \tag{4.40}$$

A complex vector can be written as a sum of its real and imaginary parts. For $u^j := ((u^j)_1, \ldots, (u^j)_{n_j}) \in \mathbb{C}^{n_j}$, one can write that

$$u^j \;=\; x_j^{\mathrm{Re}} + \sqrt{-1} x_j^{\mathrm{Im}}, \quad x_j^{\mathrm{Re}} \in \mathcal{R}^{n_j}, \quad x_j^{\mathrm{Im}} \in \mathcal{R}^{n_j}.$$

The coordinates of $x_j^{\mathrm{Re}}, x_j^{\mathrm{Im}}$ can be labelled as

$$x_j^{\mathrm{Re}} = \big((x_j^{\mathrm{Re}})_1, \ldots, (x_j^{\mathrm{Re}})_{n_j}\big), \quad x_j^{\mathrm{Im}} = \big((x_j^{\mathrm{Im}})_1, \ldots, (x_j^{\mathrm{Im}})_{n_j}\big).$$

It is interesting to note that, for all unitary scalars $\tau_i^j$ (i.e., $|\tau_i^j| = 1$), the above decomposition for $\mathcal{H}$ is the same as

$$\mathcal{H} \;=\; [\tau_1^1 u_1^1, \ldots, \tau_1^m u_1^m]_{\otimes h} + \cdots + [\tau_r^1 u_r^1, \ldots, \tau_r^m u_r^m]_{\otimes h}.$$

For each $u_i^j$, there exists a unitary scalar $\tau_i^j$ such that the first entry of $\tau_i^j u_i^j$ is real and nonnegative, i.e., $(x_j^{\mathrm{Re}})_1 \geq 0$, $(x_j^{\mathrm{Im}})_1 = 0$. By Theorem 4.27, a Hermitian tensor $\mathcal{H} \in \mathbb{C}^{[n_1,\ldots,n_m]}$ is separable if and only if

$$\mathcal{H} = \int z_1 \otimes \cdots \otimes z_m \otimes \overline{z}_1 \otimes \cdots \otimes \overline{z}_m \mathtt{d}\mu,$$

for a Borel measure $\mu$ supported in the multi-sphere $\mathbb{S}_{\mathbb{C}}^{n_1,\ldots,n_m}$. In view of the above observation, such a measure $\mu$ can be further chosen to be supported in the set

$$\mathbb{S}_{\mathbb{C},+}^{n_1,\ldots,n_m} := \Big\{(u^1, \ldots, u^m) : \;\; u^j \in \mathbb{C}^{n_j},\, \|u^j\| = 1,\, (x_j^{\mathrm{Re}})_1 \geq 0, (x_j^{\mathrm{Im}})_1 = 0 \;\Big\}.$$

For convenience of notation, for each $j = 1, \ldots, m$, we denote that

$$x_j := (x_j^{\mathrm{Re}}, x_j^{\mathrm{Im}}) = \big((x_j^{\mathrm{Re}})_1, \ldots, (x_j^{\mathrm{Re}})_{n_j},\, (x_j^{\mathrm{Im}})_2, \ldots, (x_j^{\mathrm{Im}})_{n_j}\big) \in \mathcal{R}^{2n_j-1}.$$

For neatness of labelling, we also write that

$$x_j := \big((x_j)_1, \ldots, (x_j)_{n_j},\, (x_j)_{n_j+1}, \ldots, (x_j)_{2n_j-1}\big).$$

Then $\mathbb{S}_{\mathbb{C},+}^{n_1,\ldots,n_m}$ can be equivalently written as the semialgebraic set

$$K \;:=\; \Big\{(x_1, \ldots, x_m) : \;\; x_j \in \mathcal{R}^{2n_j-1},\, \|x_j\| = 1,\, (x_j)_1 \geq 0. \;\Big\}. \tag{4.41}$$

Let $\mathscr{B}(K)$ denote the set of all Borel measures supported in $K$ and

$$x \;:=\; (x_1, \ldots, x_m) \in \mathcal{R}^{2N-m}, \tag{4.42}$$

with $N := \sum_{j=1}^{m} n_j$. The set $K$ can be equivalently given as

$$K = \{x \in \mathcal{R}^{2N-m} : h(x) = 0, g(x) \geq 0\},$$

where $h := \big(\|x_1\|^2 - 1, \ldots, \|x_m\|^2 - 1\big)$ and $g(x) := \big((x_1)_1, \ldots, (x_m)_1\big)$.

Next, we consider the label set

$$S \;:=\; \{(i_1, \ldots, i_m) : i_1 \in [n_1], \ldots, i_m \in [n_m]\}. \tag{4.43}$$

Its cardinality is $M = n_1 \cdots n_m$. For two labeling tuples in $S$

$$I := (i_1, \ldots, i_m), \quad J := (j_1, \ldots, j_m),$$

we define the ordering $I < J$ if the first nonzero entry of $I - J$ is negative. For $I < J$, let $P_{IJ}$ denote the polynomial

$$P_{IJ}(x) \;:=\; \prod_{s=1}^{m} \big(x_s^{\mathrm{Re}} + \sqrt{-1}x_s^{\mathrm{Im}}\big)_{i_s} \cdot \big(x_s^{\mathrm{Re}} - \sqrt{-1}x_s^{\mathrm{Im}}\big)_{j_s}. \tag{4.44}$$

Therefore, the positive Hermitian decomposition (4.40) is equivalent to that

$$\mathcal{H}_{IJ} = \int_K P_{IJ}(x)\mathtt{d}\mu, \quad \text{for all } I, J \in S, \tag{4.45}$$

for a Borel measure $\mu$ supported in $K$. Then Theorem 4.27 implies the following.

**Corollary 4.30.** *A tensor $\mathcal{H} \in \mathbb{C}^{[n_1,\ldots,n_m]}$ is separable if and only if there exists a measure $\mu \in \mathscr{B}(K)$ such that* (4.45) *is satisfied.*

Next, we formulate the above as a moment optimization problem, following similar ideas in [87] adapted to our new formulation of the moment problem. We write each $P_{IJ}$ as a sum of real and imaginary parts

$$P_{IJ}(x) \;=\; R_{IJ}(x) + \sqrt{-1}T_{IJ}(x)$$

for real polynomials $R_{IJ}, T_{IJ} \in \mathcal{R}[x] = \mathcal{R}[x_1, \ldots, x_m]$. Likewise, the tensor entries $\mathcal{H}_{IJ}$ of $\mathcal{H}$ can be written as

$$\mathcal{H}_{IJ} \;=\; \mathcal{H}_{IJ}^{\mathrm{Re}} + \sqrt{-1}\mathcal{H}_{IJ}^{\mathrm{Im}}, \tag{4.46}$$

for real entries $\mathcal{H}_{IJ}^{\mathrm{Re}}, \mathcal{H}_{IJ}^{\mathrm{Im}}$. Since $\mathcal{H}$ is Hermitian, it holds that

$$\mathcal{H}_{IJ}^{\mathrm{Re}} = \mathcal{H}_{JI}^{\mathrm{Re}}, \quad \mathcal{H}_{IJ}^{\mathrm{Im}} = -\mathcal{H}_{JI}^{\mathrm{Im}}, \quad \mathcal{H}_{II}^{\mathrm{Im}} = 0.$$

Therefore, it suffices to consider $\mathcal{H}_{IJ}^{\mathrm{Re}}$ with $I \le J$ and $\mathcal{H}_{IJ}^{\mathrm{Im}}$ with $I < J$. For a polynomial $F(x) \in \mathcal{R}[x]$, we consider the moment optimization problem

$$\begin{cases} \min\limits_{\mu} & \int_K F(x)\mathtt{d}\mu \\ \text{s.t.} & \mathcal{H}_{IJ}^{\mathrm{Re}} = \int_K R_{IJ}(x)\mathtt{d}\mu, \ \ (I \le J), \\ & \mathcal{H}_{IJ}^{\mathrm{Im}} = \int_K T_{IJ}(x)\mathtt{d}\mu, \ \ (I < J), \\ & \mu \in \mathscr{B}(K). \end{cases} \tag{4.47}$$

To ensure that (4.47) has a unique minimizer, one can choose $F(x)$ to be a generic polynomial in $\Sigma[x]_{2m}$. We introduce the moment cone

$$\mathscr{R}_{2m}(K) := \left\{ y = (y_\alpha) : \begin{array}{l} \exists \mu \in \mathscr{B}(K), \ \text{such that} \\ (y)_\alpha = \int x^\alpha \mathtt{d}\mu, \ \forall \alpha \in \mathbb{N}^{2N-m}_{2m} \end{array} \right\}. \tag{4.48}$$

Then, (4.47) is equivalent to the following optimization

$$\begin{cases} \min\limits_{y} & \langle F, y \rangle \\ \text{s.t.} & \mathcal{H}_{IJ}^{\mathrm{Re}} = \langle R_{IJ}, y \rangle, \ \ (I \le J), \\ & \mathcal{H}_{IJ}^{\mathrm{Im}} = \langle T_{IJ}, y \rangle, \ \ (I < J), \\ & y \in \mathscr{R}_{2m}(K). \end{cases} \tag{4.49}$$

For the coefficient vector $\mathbf{f} := (f^{\mathrm{Re}}, f^{\mathrm{Im}})$ with

$$f^{\mathrm{Re}} := \left(f_{IJ}^{\mathrm{Re}}\right)_{I \le J}, \quad f^{\mathrm{Im}} := \left(f_{IJ}^{\mathrm{Im}}\right)_{I < J},$$

denote the polynomials

$$G(\mathbf{f}) := F(x) - \sum_{I \le J} f_{IJ}^{\mathrm{Re}} \cdot R_{IJ}(x) - \sum_{I < J} f_{IJ}^{\Im} \cdot T_{IJ}(x).$$

Then the optimization problem dual to (4.49) is

$$\begin{cases} \max\limits_{\mathbf{f}} & \sum_{I \le J} f_{IJ}^{\mathrm{Re}} \mathcal{H}_{IJ}^{\mathrm{Re}} + \sum_{I < J} f_{IJ}^{\mathrm{Im}} \mathcal{H}_{IJ}^{\mathrm{Im}} \\ \text{s.t.} & G(\mathbf{f}) \in \mathscr{P}_{2m}(K), \end{cases} \tag{4.50}$$

where $\mathscr{P}_{2m}(K)$ denotes the cone of polynomials in $\mathcal{R}[x]_{2m}$ that are nonnegative on $K$.

The moment cone $\mathscr{R}_{2m}(K)$ can be approximated well by semidefinite relaxations. Select a generic $F(x) \in \Sigma[x]_{2m}$. Consider the hierarchy of semidefinite relaxations

$$\begin{cases} \min\limits_{w} & \langle F, w\rangle \\ \text{s.t.} & \langle R_{IJ}, w\rangle = \mathcal{H}_{IJ}^{\mathrm{Re}}, \ \ I, J \in S, I \le J \\ & \langle T_{IJ}, w\rangle = \mathcal{H}_{IJ}^{\mathrm{Im}}, \ \ I, J \in S, I < J \\ & L_h^{(k)}(w) = 0, M_k(w) \succeq 0, L_g^{(k)} \succeq 0, \\ & w \in \mathcal{R}^{\mathbb{N}_{2k}^{2N-m}}, \end{cases} \tag{4.51}$$

for relaxation orders $k = m, m+1, \cdots$. The dual optimization of the above is

$$\begin{cases} \max\limits_{\mathbf{f}} & \sum_{I \le J} f_{IJ}^{\mathrm{Re}} \mathcal{H}_{IJ}^{\mathrm{Re}} + \sum_{I<J} f_{IJ}^{\mathrm{Im}} \mathcal{H}_{IJ}^{\mathrm{Im}} \\ \text{s.t.} & G(\mathbf{f}) \in I_{2k}(h) + Q_k(g). \end{cases} \tag{4.52}$$

This yields the following algorithm.

**Algorithm 4.31.** Detecting separability for Hermitian tensors.

*Input: A Hermitian tensor $\mathcal{H} \in \mathbb{C}^{[n_1,\ldots,n_m]}$.*

*Output: Either a positive $\mathbb{C}$-Hermitian decomposition of $\mathcal{H}$, particularly affirming membership in $\mathscr{S}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$, or an answer that $\mathcal{H}$ is not separable.*

*Step 0: Let $k = m$. Choose a generic $F(x) \in \Sigma[x]_{2m}$.*

*Step 1: Solve the semidefinite optimization* (4.51)*. If it is infeasible, output that $\mathcal{H}$ is not separable, and stop; otherwise, solve it for a minimizer $w^{\star,k}$ and let $t := 1$.*

*Step 2: Let $w := w^{\star,k}|_{2t}$. Check whether or not the rank condition*

$$rank M_{t-1}(w) \ = \ rank M_t(w) \tag{4.53}$$

*holds. If it does, go to Step 4; otherwise, go to Step 3.*

*Step 3: If $t < k$, set $t = t+1$ and go to Step 2; otherwise, set $k = k+1$ and go to Step 1.*

*Step 4: Let $r := rank M_t(w)$. Compute the weights $\lambda_1 > 0, \ldots, \lambda_r > 0$ and $v^{(1)}, \ldots, v^{(r)} \in K$ such that*

$$w \ = \ \lambda_1 [v^{(1)}]_{2t} + \cdots + \lambda_r [v^{(r)}]_{2t}. \tag{4.54}$$

*For each* $i = 1, \ldots, r$, *write that* $v^{(i)} = (v_1^{(i)}, \ldots, v_m^{(i)})$ *with each* $v_j^{(i)} \in \mathcal{R}^{2n_j-1}$ *and for* $j = 1, \ldots, m$, *let*

$$u_i^j := \big((v_j^{(i)})_1, \ldots, (v_j^{(i)})_{n_j}\big) + \sqrt{-1}\big(0, (v_j^{(i)})_{n_j+1}, \ldots, (v_j^{(i)})_{2n_j-1}\big).$$

*Output the positive decomposition* $\mathcal{H} = \sum_{i=1}^r \lambda_i [u_i^1, \ldots, u_i^m]_{\otimes h}$.

In the Step 0, we can choose the generic polynomial $F \in \Sigma[x]_{2m}$ as $F = [x]_m^T (G^T G)[x]_m$ with a random square matrix $G$ of length $\binom{2N-m+d/2}{d/2}$, i.e., each entry of $G$ is a real random variable fulfilling normal (Gaussian) distribution. The Step 1 is justified by Theorem 4.32. The Step 2 requires to check if $w$ satisfies the rank condition (4.53). When the rank condition (4.53) is satisfied, one can use the method in [67] to get a positive $\mathbb{C}$-Hermitian decomposition in (4.54). This method is implemented in the software `GloptiPoly3` [66]. We point out that the vectors $u_i^j$ must belong to the set $\mathbb{S}_{\mathbb{C},+}^{n_1,\ldots,n_m}$ if (4.53) holds (see [102]). Therefore, Algorithm 4.31 can be conveniently programmed in `GloptiPoly3`.

Now we study the convergence of Algorithm 4.31. In [87, Theorem 2], Li and Ni proved the subsequent properties for their semidefinite relaxations: (I) If the semidefinite relaxation is infeasible for some order $k$, then the Hermitian tensor $\mathcal{H}$ is not separable. (II) If $\mathcal{H}$ is separable, then their relaxations can asymptotically get a positive Hermitian decomposition. Their proof uses the results in [104]. We prove stronger convergence results for Algorithm 4.31. In fact, if $\mathcal{H}$ is not separable, we show that the semidefinite relaxation (4.51) must be infeasible for all $k$ large enough. Furthermore, we prove the finite convergence for Algorithm 4.31 under some conditions.

First, we show that non-separability of a Hermitian tensor is equivalent to infeasibility of the semidefinite relaxation (4.51) for some order $k$.

**Theorem 4.32** ( [49]). *Let* $\mathcal{H}$, $\mathcal{H}_{IJ}^{\mathrm{Re}}$, $\mathcal{H}_{IJ}^{\mathrm{Im}}$ *be as in* (4.46). *Then,* $\mathcal{H}$ *is not separable (i.e.,* $\mathcal{H} \notin \mathscr{S}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$*) if and only if the semidefinite relaxation* (4.51) *is infeasible for some* $k$.

*Proof.* "if" direction: Note that (4.51) is a relaxation of (4.49). If (4.51) is infeasible, then (4.49) must be infeasible and hence $\mathcal{H}$ is not separable.

"only if" direction: Recall that $\mathscr{P}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$ is the dual cone of $\mathscr{S}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$, by Theorem 4.26. If $\mathcal{H}$ is not $\mathbb{C}$-separable, there exists a psd tensor $\mathcal{A}_1 \in \mathscr{P}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$ such that $\langle \mathcal{A}_1, \mathcal{H} \rangle < 0$. For $\epsilon > 0$, let $\mathcal{A}$ be the Hermitian tensor such that

$$\mathcal{A}(z, \overline{z}) = \mathcal{A}_1(z, \overline{z}) + \epsilon (z_1^* z_1) \cdots (z_m^* z_m).$$

If $\epsilon > 0$ is sufficiently small, $\langle \mathcal{A}, \mathcal{H} \rangle < 0$ and $\mathcal{A}$ is $\mathbb{C}$-positive definite. Write that $\mathcal{A} = \mathcal{A}^{\mathrm{Re}} + \sqrt{-1}\mathcal{A}^{\mathrm{Im}}$, where $\mathcal{A}^{\mathrm{Re}}, \mathcal{A}^{\mathrm{Im}}$ are both real tensors. Since $\mathcal{A}$ is positive definite, for the variable $x$, we have that

$$\mathcal{A}(x) := \langle \mathcal{A}, [x_1^{\mathrm{Re}} + \sqrt{-1}x_1^{\mathrm{Im}}, \ldots, x_m^{\mathrm{Re}} + \sqrt{-1}x_m^{\mathrm{Im}}]_{\otimes h} \rangle =$$

$$\sum_{I,J \in S} \mathcal{A}_{IJ}^{\mathrm{Re}} R_{IJ} + \sum_{I,J \in S} \mathcal{A}_{IJ}^{\mathrm{Im}} T_{IJ} > 0, \text{ for all } x \in K.$$

Select $\mathbf{f} = (f^{\mathrm{Re}}, f^{\mathrm{Im}})$ as follows

$$f_{IJ}^{\mathrm{Re}} = \begin{cases} -\mathcal{A}_{IJ}^{\mathrm{Re}} & \text{if } I = J \\ -2\mathcal{A}_{IJ}^{\mathrm{Re}} & \text{if } I < j \end{cases}, \; f_{IJ}^{\mathrm{Im}} = -2\mathcal{A}_{IJ}^{\mathrm{Im}} \text{ for } I < J.$$

Thus, $G(\mathbf{f}) = F(x) + \mathcal{A}(x)$. By Putinar's Positivstellensatz [124], we have $\mathcal{A}(x) \in I_{2k_0}(h) + Q_{k_0}(g)$ for some $k_0$. Since $F(x) \in \Sigma[x]_{2m}$, we have

$$F(x) + \tau\mathcal{A}(x) \; \in \; I_{2k_0}(h) + Q_{k_0}(g)$$

for all $\tau > 0$. This implies that $\tau\mathbf{f}$ is feasible for (4.52) for all $\tau > 0$. Moreover, for the above choice of $\mathbf{f}$, the objective value in (4.52) is such that

$$\sum_{I \leq J} \tau f_{IJ}^{\mathrm{Re}} \mathcal{H}_{IJ}^{\mathrm{Re}} + \sum_{I < J} \tau f_{IJ}^{\mathrm{Im}} \mathcal{H}_{IJ}^{\mathrm{Im}} = \tau \langle -\mathcal{A}, \mathcal{H} \rangle \to +\infty,$$

as $\tau \to +\infty$. Therefore, the dual problem (4.52) is unbounded from above and hence, by duality, the primal problem (4.51) must be infeasible for all $k \geq k_0$. □

Second, we prove the asymptotic convergence of Algorithm 4.31. For the minimizer $w^{\star,k}$, recall that the notation $w^{\star,k}|_{2m}$ denotes the subvector of entries $(w^{\star,k})_\alpha$ with $|\alpha| \leq 2m$. The $w^{\star,k}|_{2m}$ is called the truncation of $w^{\star,k}$ with degree $2m$. The asymptotic convergence for Algorithm 4.31 means that the truncated sequence $\{w^{\star,k}|_{2m}\}_{k=m}^{\infty}$ of minimizers is bounded and all its accumulation points are optimizers of the moment optimization (4.47).

**Theorem 4.33** ( [49]). *Let $\mathcal{H}$ be a separable Hermitian tensor and let $\mathcal{H}_{IJ}^{\mathrm{Re}}, \mathcal{H}_{IJ}^{\mathrm{Im}}$ be as in* (4.46). *If $F(x)$ is a generic polynomial in $\Sigma[x]_{2m}$, then we have the following properties:*

*(i) For all $k \geq m$, the semidefinite relaxation* (4.51) *has an optimizer $w^{\star,k}$.*

*(ii) The truncated sequence $\{w^{\star,k}|_{2m}\}_{k=m}^{\infty}$ is bounded and all its accumulation points are optimizers of the moment optimization problem* (4.47).

*Proof.* Since the Hermitian tensor $\mathcal{H}$ is separable (i.e., $\mathcal{H} \in \mathscr{S}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$), there is a satisfactory measure $\mu$ for (4.47), by Corollary 4.30. Hence the problem (4.49) is feasible.

(i) Since (4.49) is feasible, the problem (4.51) is feasible as well. The genericity of $F(x)$ implies that $F$ lies in the interior of $\Sigma[x]_{2m}$. Therefore, (4.51) is bounded from below and $(f^{\mathrm{Re}}, f^{\mathrm{Im}}) = (0,0)$ is an interior point of the dual optimization (4.52). Therefore, the strong duality holds and the semidefinite relaxation (4.51) must have an optimizer $w^{\star,k}$.

(ii) The set $K$ satisfies the ball condition

$$\|x\|^2 = \sum_{j=1}^{m} \|x_j^{\mathrm{Re}}\|^2 + \|x_j^{\mathrm{Im}}\|^2 \leq m,$$

so the archimedeanness holds for the constraining polynomials of $K$. The conclusion then follows from [105, Theorem 4.3(ii)]. □

Last, we study the finite convergence property of Algorithm 4.31, i.e., we investigate conditions for it to terminate within finitely many loops. The finite convergence can occur under some assumptions on the optimizer of (4.50).

**Assumption 4.34.** *Suppose $\boldsymbol{f}^*$ is a maximizer of the optimization* (4.50) *and the polynomial $F^* := G(\boldsymbol{f}^*)$ satisfies the conditions:*

*i) There exists a $k_1$ such that $F^* \in I_{2k_1}(h) + Q_{k_1}(g)$;*

*ii) The optimization problem*

$$\min\ F^*(x)\ s.t.\ h(x) = 0, g(x) \geq 0$$

*has finitely many KKT points $u$ for which $F^*(u) = 0$.*

We refer to [102] for the notion of KKT points. Assumption 4.34 holds if $F^*$ is a generic point on the boundary of $\mathscr{P}_{2m}(K)$ (see [103]). The following is the finite convergence result.

**Theorem 4.35** ( [49])**.** *Let $\mathcal{H} \in \mathscr{S}_{\mathbb{C}}^{[n_1,\ldots,n_m]}$ and $\mathcal{H}_{IJ}^{\mathrm{Re}}, \mathcal{H}_{IJ}^{\mathrm{Im}}$ be as in* (4.46)*. Suppose $F(x) \in \mathrm{int}(\Sigma[x]_{2m})$, Assumption 4.34 holds, and $w^{\star,k}$ is a minimizer of* (4.51) *for the relaxation order $k$. Then, for all $k > t$ sufficiently large, the rank condition* (4.53) *must be satisfied.*

*Proof.* The conclusion follows from Theorem 4.6 of [105]. □

We present examples for detecting separability of Hermitian tensors by Algorithm 4.31. The algorithm can be implemented in the software `GloptiPoly3` [66], which calls the SDP solver `SeDuMi` [67]. Since the semidefinite programs are solved numerically, we display only four decimal digits for the computational results. The computation is implemented in MATLAB R2019b, on an `Intel(R) Core(TM) i7-8550U CPU` with 3.79 GHz and 16 GB of RAM.

**Example 4.36.** *Consider the Hankel tensor $\mathcal{H} \in \mathbb{C}^{[2,2]}$ in [111] such that*

$$\mathcal{H}_{i_1 i_2 j_1 j_2} = i_1 + i_2 + j_1 + j_2$$

*for all $1 \leq i_1, i_2, j_1, j_2 \leq 2$. The tensor $\mathcal{H}$ is not separable, detected by Algorithm 4.31, since the semidefinite relaxation* (4.51) *is infeasible for $k = 2$. The computation took around* 0.8 *second.*

**Example 4.37.** *Consider the tensor $\mathcal{H} \in \mathbb{C}^{[3,3]}$ such that*

$$\mathcal{H}_{i_1 i_2 j_1 j_2} = i_1 j_1 + i_2 j_2$$

*for all $i_1, i_2, j_1, j_2$ in the range. It is separable, detected by Algorithm 4.31 for $k = 2$. We got the positive Hermitian decomposition $\mathcal{H} = \lambda_1 [u_1^1, u_1^2]_{\otimes h} + \lambda_2 [u_2^1, u_2^2]_{\otimes h}$, with weights $\lambda_1 = \lambda_2 = 42$ and*

$$u_1^1 = \begin{pmatrix} \sqrt{14}/14 \\ \sqrt{14}/7 \\ 3/\sqrt{14} \end{pmatrix}, u_1^2 = \begin{pmatrix} \sqrt{3}/3 \\ \sqrt{3}/3 \\ \sqrt{3}/3 \end{pmatrix}, u_2^1 = \begin{pmatrix} \sqrt{3}/3 \\ \sqrt{3}/3 \\ \sqrt{3}/3 \end{pmatrix}, u_2^2 = \begin{pmatrix} \sqrt{14}/14 \\ \sqrt{14}/7 \\ 3/\sqrt{14} \end{pmatrix}.$$

*The computation took around* 2.7 *seconds.*

**Example 4.38.** *Consider the Hermitian tensor $\mathcal{H} = \frac{1}{2}\psi_1 \otimes \overline{\psi_1} + \frac{1}{2}\psi_2 \otimes \overline{\psi_2}$, where*

$$\begin{aligned} \psi_1 &:= \tfrac{1}{\sqrt{3}}(e_1 \otimes e_1 + e_1 \otimes e_2 + \sqrt{-1} e_2 \otimes e_2), \\ \psi_2 &:= \tfrac{1}{3\sqrt{2}}(e_1 \otimes e_1 - e_1 \otimes e_2 + 4\sqrt{-1} e_2 \otimes e_1), \end{aligned}$$

*for $e_1 := (1, 0), e_2 := (0, 1)$. In terms of the eigenvalue decomposition of the Hermitian flattening matrix, it was shown in [101, Example 6.1] that this state is not separable. The semidefinite relaxation* (4.51) *is infeasible for $k = 2$, so we know $\mathcal{H} \notin \mathscr{S}_{\mathbb{C}}^{[2,2]}$ not separable. The computation took around* 0.8 *second.*

In what follows, we consider more general Hermitian tensors. For neatness, the weights $\lambda_i$ are set to be one by scaling the vectors $u_i^j$ accordingly. That is, we display the positive Hermitian decomposition as $\mathcal{H} = \sum_{i=1}^{r}[u_i^1, \ldots, u_i^m]_{\otimes h}$. Moreover, we use the notation $\mathrm{i} := \sqrt{-1}$. Note that a Hermitian tensor $\mathcal{H}$ can be equivalently represented by its Hermitian flattening matrix $\mathfrak{m}(\mathcal{H})$.

**Example 4.39.** *Consider $\mathcal{H} \in \mathscr{S}_{\mathbb{C}}^{[3,3]}$ whose flattening matrix $\mathfrak{m}(\mathcal{H})$ is*

$$\begin{pmatrix}
10 & -2-2\mathrm{i} & 1+1\mathrm{i} & 7-\mathrm{i} & -2-4\mathrm{i} & 2\mathrm{i} & -4-6\mathrm{i} & 0 & -2 \\
-2+2\mathrm{i} & 10 & -6+1\mathrm{i} & -2 & 5+3\mathrm{i} & -5+1\mathrm{i} & -4-4\mathrm{i} & -4+2\mathrm{i} & 3+1\mathrm{i} \\
1-\mathrm{i} & -6-\mathrm{i} & 12 & 2+4\mathrm{i} & -5-\mathrm{i} & 8+1\mathrm{i} & 4+6\mathrm{i} & -3-\mathrm{i} & -4-2\mathrm{i} \\
7+1\mathrm{i} & -2 & 2-4\mathrm{i} & 9 & -1-3\mathrm{i} & -1-\mathrm{i} & 1-7\mathrm{i} & -2 & -4+2\mathrm{i} \\
-2+4\mathrm{i} & 5-3\mathrm{i} & -5+1\mathrm{i} & -1+3\mathrm{i} & 8 & -5-\mathrm{i} & -2\mathrm{i} & 4 & 2\mathrm{i} \\
-2\mathrm{i} & -5-\mathrm{i} & 8-\mathrm{i} & -1+1\mathrm{i} & -5+1\mathrm{i} & 11 & 2+4\mathrm{i} & -2+4\mathrm{i} & 3-5\mathrm{i} \\
-4+6\mathrm{i} & -4+4\mathrm{i} & 4-6\mathrm{i} & 1+7\mathrm{i} & 2\mathrm{i} & 2-4\mathrm{i} & 20 & -3-\mathrm{i} & 2 \\
0 & -4-2\mathrm{i} & -3+1\mathrm{i} & -2 & 4 & -2-4\mathrm{i} & -3+1\mathrm{i} & 17 & -9+1\mathrm{i} \\
-2 & 3-\mathrm{i} & -4+2\mathrm{i} & -4-2\mathrm{i} & -2\mathrm{i} & 3+5\mathrm{i} & 2 & -9-\mathrm{i} & 22
\end{pmatrix}.$$

*By Algorithm 4.31 with relaxation order $k = 2$, we got the positive Hermitian decomposition $\mathcal{H} = \sum_{i=1}^{9}[u_i^1, u_i^2]_{\otimes h}$, where $U_1 := [u_1^1, \ldots, u_9^1]^T, U_2 := [u_1^2, \ldots, u_9^2]^T$ are given as follows*

$$U_1 = \begin{pmatrix}
0.0000 & -1.1632 - 0.5687\mathrm{i} & -0.2972 - 0.8659\mathrm{i} \\
-0.0000 & -1.1309 + 0.5564\mathrm{i} & -0.8436 - 0.2873\mathrm{i} \\
1.3161 & 0.6580 - 0.6580\mathrm{i} & -0.6580 - 0.6580\mathrm{i} \\
1.0000 & -1.0000 + 1.0000\mathrm{i} & 1.0000 - 1.0000\mathrm{i} \\
1.4142 & 0.7071 + 0.7071\mathrm{i} & -1.4142 + 0.0000\mathrm{i} \\
0.8691 & -0.4346 - 0.4345\mathrm{i} & 0.8691 - 0.0000\mathrm{i} \\
1.3375 & -0.6687 - 0.6687\mathrm{i} & 0.0000 + 1.3375\mathrm{i} \\
1.3375 & -0.6687 + 0.6687\mathrm{i} & 1.3375 + 0.0000\mathrm{i} \\
0.6921 & -0.3460 - 0.3461\mathrm{i} & 0.6921 + 0.0000\mathrm{i}
\end{pmatrix},$$

$$U_2 = \begin{pmatrix}
0.7928 & 0.0000 - 0.7928\mathrm{i} & -0.7928 - 0.7929\mathrm{i} \\
0.7718 & -0.0000 - 0.7718\mathrm{i} & -0.7718 - 0.7718\mathrm{i} \\
1.0746 & 0.0000 - 0.0000\mathrm{i} & -1.0746 + 1.0746\mathrm{i} \\
1.4142 & 0.7071 - 0.7071\mathrm{i} & -0.0000 - 1.4142\mathrm{i} \\
1.0000 & 1.0000 + 1.0000\mathrm{i} & -1.0000 + 1.0000\mathrm{i} \\
0.0000 & 0.0000 + 0.0000\mathrm{i} & -0.5884 + 1.2419\mathrm{i} \\
1.0574 & 1.0574 - 0.0000\mathrm{i} & 1.0574 + 1.0574\mathrm{i} \\
1.0574 & 1.0574 - 0.0000\mathrm{i} & -1.0574 + 1.0574\mathrm{i} \\
0.0000 & 0.0000 - 0.0000\mathrm{i} & 0.8382 + 0.7034\mathrm{i}
\end{pmatrix}.$$

*The computation took around 4.2 seconds. This Hermitian tensor is separable.*

**Example 4.40.** *Consider the tensor* $\mathcal{H} \in \mathscr{S}_{\mathbb{C}}^{[2,2,2]}$ *with* $\mathfrak{m}(\mathcal{H})$ *being the matrix*

$$\begin{pmatrix}
18 & -2+8\mathrm{i} & -16-8\mathrm{i} & 4-4\mathrm{i} & -2-4\mathrm{i} & 2+16\mathrm{i} & 2-2\mathrm{i} & -4-12\mathrm{i} \\
-2-8\mathrm{i} & 50 & 16\mathrm{i} & -34-26\mathrm{i} & -10+20\mathrm{i} & 2+32\mathrm{i} & 8-16\mathrm{i} & 30-30\mathrm{i} \\
-16+8\mathrm{i} & -16\mathrm{i} & 32 & -18+18\mathrm{i} & 10+6\mathrm{i} & -8-16\mathrm{i} & -2-6\mathrm{i} & 6+14\mathrm{i} \\
4+4\mathrm{i} & -34+26\mathrm{i} & -18-18\mathrm{i} & 78 & 4-28\mathrm{i} & 2+2\mathrm{i} & -14+30\mathrm{i} & -4+22\mathrm{i} \\
-2+4\mathrm{i} & -10-20\mathrm{i} & 10-6\mathrm{i} & 4+28\mathrm{i} & 22 & 12+6\mathrm{i} & -18-8\mathrm{i} & -12 \\
2-16\mathrm{i} & 2-32\mathrm{i} & -8+16\mathrm{i} & 2-2\mathrm{i} & 12-6\mathrm{i} & 70 & -16+12\mathrm{i} & -50-26\mathrm{i} \\
2+2\mathrm{i} & 8+16\mathrm{i} & -2+6\mathrm{i} & -14-30\mathrm{i} & -18+8\mathrm{i} & -16-12\mathrm{i} & 30 & 2+10\mathrm{i} \\
-4+12\mathrm{i} & 30+30\mathrm{i} & 6-14\mathrm{i} & -4-22\mathrm{i} & -12 & -50+26\mathrm{i} & 2-10\mathrm{i} & 86
\end{pmatrix}.$$

*By Algorithm 4.31 with relaxation order* $k = 3$*, we got the positive Hermitian decomposition* $\mathcal{H} = \sum_{i=1}^{6} [u_i^1, u_i^2, u_i^3]_{\otimes h}$*, where*

$$U_1 := [u_1^1, \ldots, u_6^1]^T,\ U_2 := [u_1^2, \ldots, u_6^2]^T,\ U_3 := [u_1^3, \ldots, u_6^3]^T$$

*are shown as follows*

$$U_1 = \begin{pmatrix}
1.0191 & 2.0381 - 0.0000\mathrm{i} \\
1.2222 & -1.2222 - 1.2222\mathrm{i} \\
-0.0000 & -1.2100 - 1.6470\mathrm{i} \\
-0.0001 & 1.2959 - 0.4964\mathrm{i} \\
1.4837 & -0.7419 - 0.7418\mathrm{i} \\
1.3077 & -1.3077 + 0.0000\mathrm{i}
\end{pmatrix}, \quad U_2 = \begin{pmatrix}
1.6113 & -1.6113 + 0.0000\mathrm{i} \\
0.9467 & -1.8935 + 0.0000\mathrm{i} \\
1.4451 & 0.0000 + 1.4451\mathrm{i} \\
0.9812 & 0.0000 + 0.9812\mathrm{i} \\
1.4836 & -0.7418 + 0.7418\mathrm{i} \\
0.8270 & 0.0000 + 1.6541\mathrm{i}
\end{pmatrix},$$

$$U_3 = \begin{pmatrix}
1.2181 & -0.6090 - 1.8270\mathrm{i} \\
1.7285 & -0.8642 + 0.8642\mathrm{i} \\
1.4451 & 0.8671 - 1.1561\mathrm{i} \\
0.9812 & 0.5888 - 0.7851\mathrm{i} \\
1.2848 & -0.0000 + 1.2850\mathrm{i} \\
1.3077 & 1.3077 - 0.0000\mathrm{i}
\end{pmatrix}.$$

*The computation took around* 5 *minutes. This Hermitian tensor is separable.*

**Acknowledgement.** The Sections 4.1-4.6 of the Chapter 4 are a reprint of the material as it appears in *SIAM Journal on Matrix Analysis and Applications* 2020 [110]. The dissertation author coauthored this paper with Nie, Jiawang. The Section 4.7 of the Chapter 4 is part of the publication that has been accepted for publication in *Linear and Multilinear Algebra* 2021 [49]. The dissertation author coauthored this paper with Dressler, Mareike and Nie, Jiawang.

# Chapter 5

# Learning Gaussian Mixture Models

## 5.1 Gaussian mixture models

A Gaussian mixture model consists of several component Gaussian distributions. For given samples of a Gaussian mixture model, people often need to estimate parameters for each component Gaussian distribution [58, 85]. Consider a Gaussian mixture model with $r$ components. For each $i \in [r] := \{1, \ldots, r\}$, let $\omega_i$ be the positive probability for the $i$th component Gaussian to appear in the mixture model. We have each $\omega_i > 0$ and $\sum_{i=1}^{r} \omega_i = 1$. Suppose the $i$th Gaussian distribution is $\mathcal{N}(\mu_i, \Sigma_i)$, where $\mu_i \in \mathbb{R}^d$ is the expectation (or mean) and $\Sigma_i \in \mathbb{R}^{d \times d}$ is the covariance matrix. Let $y \in \mathbb{R}^d$ be the random vector for the Gaussian mixture model and let $y_1, \ldots, y_N$ be identically independent distributed (i.i.d) samples from the mixture model. Each $y_j$ is sampled from one of the $r$ component Gaussian distributions, associated with a label $Z_j \in [r]$ indicating the component that it is sampled from. The probability that a sample comes from the $i$th component is $\omega_i$. When people observe only samples without labels, the $Z_j$'s are called latent variables. The density function for the random variable $y$ is

$$f(y) := \sum_{i=1}^{r} \omega_i \frac{1}{\sqrt{(2\pi)^d \det \Sigma_i}} \exp\Big\{ -\frac{1}{2}(y-\mu_i)^T \Sigma_i^{-1} (y-\mu_i) \Big\},$$

where $\mu_i$ is the mean and $\Sigma_i$ is the covariance matrix for the $i$th component.

Learning a Gaussian mixture model is to estimate the parameters $\omega_i, \mu_i, \Sigma_i$ for each $i \in [r]$, from given samples of $y$. The number of parameters in a covariance matrix grows quadratically with respect to the dimension. Due to the curse of dimensionality, the computation becomes very expensive for large $d$ [90]. Hence, diagonal covariance matrices

are preferable in applications. In this paper, we focus on learning Gaussian mixture models with diagonal covariance matrices, i.e.,

$$\Sigma_i \,=\, \mathrm{diag}\big(\sigma_{i1}^2,\ldots,\sigma_{id}^2\big),\quad i=1,\ldots,r.$$

A natural approach for recovering the unknown parameters $\omega_i,\mu_i,\Sigma_i$ is the method of moments. It estimates parameters by solving a system of multivariate polynomial equations, from moments of the random vector $y$. Directly solving polynomial systems may encounter non-existence or non-uniqueness of statistically meaningful solutions [143]. However, for diagonal Gaussians, the third order moment tensor can help us avoid these troubles.

Let $M_3 := \mathbb{E}(y\otimes y\otimes y)$ be the third order tensor of moments for $y$. One can write that $y=\eta(z)+\zeta(z)$, where $z$ is a discrete random variable such that $\mathrm{Prob}(z=i)=\omega_i$, $\eta(i)=\mu_i\in\mathbb{R}^d$ and $\zeta(i)$ is the random variable $\zeta_i$ obeying the Gaussian distribution $\mathcal{N}(0,\Sigma_i)$. Assume all $\Sigma_i$ are diagonal, then

$$M_3=\sum_{i=1}^r\omega_i\mathbb{E}[(\eta(i)+\zeta_i)^{\otimes 3}]=\sum_{i=1}^r\omega_i\Big(\mu_i\otimes\mu_i\otimes\mu_i+\mathbb{E}[\mu_i\otimes\zeta_i\otimes\zeta_i]+ \\ \mathbb{E}[\zeta_i\otimes\mu_i\otimes\zeta_i]+\mathbb{E}[\zeta_i\otimes\zeta_i\otimes\mu_i]\Big). \quad (5.1)$$

The second equality holds because $\zeta_i$ has zero mean and

$$\mathbb{E}[\zeta_i\otimes\zeta_i\otimes\zeta_i]=\mathbb{E}[\mu_i\otimes\mu_i\otimes\zeta_i]=\mathbb{E}[\zeta_i\otimes\mu_i\otimes\mu_i]=\mathbb{E}[\mu_i\otimes\zeta_i\otimes\mu_i]=0.$$

The random variable $\zeta_i$ has diagonal covariance matrix, so $\mathbb{E}[(\zeta_i)_j(\zeta_i)_l]=0$ for $j\neq l$. Therefore,

$$\sum_{i=1}^r\omega_i\mathbb{E}[\mu_i\otimes\zeta_i\otimes\zeta_i]=\sum_{i=1}^r\sum_{j=1}^d\omega_i\sigma_{ij}^2\mu_i\otimes e_j\otimes e_j=\sum_{j=1}^d a_j\otimes e_j\otimes e_j,$$

where the vectors $a_j$ are given by

$$a_j\coloneqq\sum_{i=1}^r\omega_i\sigma_{ij}^2\mu_i,\quad j=1,\ldots,d. \quad (5.2)$$

Similarly, we have

$$\sum_{i=1}^r\omega_i\mathbb{E}[\zeta_i\otimes\mu_i\otimes\zeta_i]=\sum_{j=1}^d e_j\otimes a_j\otimes e_j,\ \sum_{i=1}^r\omega_i\mathbb{E}[\zeta_i\otimes\zeta_i\otimes\mu_i]=\sum_{j=1}^d e_j\otimes e_j\otimes a_j.$$

Therefore, we can express $M_3$ in terms of $\omega_i, \mu_i, \Sigma_i$ as

$$M_3 = \sum_{i=1}^{r} \omega_i \mu_i \otimes \mu_i \otimes \mu_i + \sum_{j=1}^{d} \Big( a_j \otimes e_j \otimes e_j + e_j \otimes a_j \otimes e_j + e_j \otimes e_j \otimes a_j \Big). \tag{5.3}$$

We are particularly interested in the following third order symmetric tensor

$$\mathcal{F} := \sum_{i=1}^{r} \omega_i \mu_i \otimes \mu_i \otimes \mu_i. \tag{5.4}$$

When the labels $i_1, i_2, i_3$ are distinct from each other, we have

$$(M_3)_{i_1 i_2 i_3} \ = \ (\mathcal{F})_{i_1 i_2 i_3} \quad \text{for} \quad i_1 \neq i_2 \neq i_3 \neq i_1.$$

Denote the label set

$$\mathcal{I} \ = \{(i_1, i_2, i_3) : \ i_1 \neq i_2 \neq i_3 \neq i_1, \ i_1, i_2, i_3 \, \text{are labels for} \, M_3\}. \tag{5.5}$$

The tensor $M_3$ can be estimated from the samplings for $y$, so the entries $\mathcal{F}_{i_1 i_2 i_3}$ with $(i_1, i_2, i_3) \in \mathcal{I}$ can also be obtained from the estimation of $M_3$. To recover the parameters $\omega_i, \mu_i$, we first find the tensor decomposition for $\mathcal{F}$, from the partially given entries $\mathcal{F}_{i_1 i_2 i_3}$ with $(i_1, i_2, i_3) \in \mathcal{I}$. Once the parameters $\omega_i, \mu_i$ are known, we can determine $\Sigma_i$ from the expressions of $a_j$ as in (5.2).

The above observation leads to the incomplete tensor decomposition problem. For a third order symmetric tensor $\mathcal{F}$ whose partial entries $\mathcal{F}_{i_1 i_2 i_3}$ with $(i_1, i_2, i_3) \in \mathcal{I}$ are known, we are looking for vectors $p_1, \ldots, p_r$ such that

$$\mathcal{F}_{i_1 i_2 i_3} \ = \ \Big( p_1^{\otimes 3} + \cdots + p_r^{\otimes 3} \Big)_{i_1 i_2 i_3}, \quad \text{for all} \quad (i_1, i_2, i_3) \in \mathcal{I}. \tag{5.6}$$

The above is called an incomplete tensor decomposition for $\mathcal{F}$. To find such a tensor decomposition for $\mathcal{F}$, a straightforward approach is to do tensor completion: first find unknown tensor entries $\mathcal{F}_{i_1 i_2 i_3}$ with $(i_1, i_2, i_3) \notin \mathcal{I}$ such that the completed $\mathcal{F}$ has low rank, and then compute the tensor decomposition for $\mathcal{F}$. However, there are serious disadvantages for this approach. The theory for tensor completion or recovery, especially for symmetric tensors, is premature. Low rank tensor completion or recovery is typically not guaranteed by the currently existing methodology. Most methods for tensor completion are based on convex relaxations, e.g., the nuclear norm or trace minimization [55, 95, 107, 140, 145]. These convex relaxations may not produce low rank completions [129].

## 5.2 Incomplete tensor decompositions

This section discusses how to compute an incomplete tensor decomposition for a symmetric tensor $\mathcal{F} \in \mathrm{S}^3(\mathbb{C}^d)$ when only its subtensor $\mathcal{F}_\Omega$ is given, for the label set $\Omega$ in (5.5). For convenience of notation, the labels for $\mathcal{F}$ begin with zeros while a vector $u \in \mathbb{C}^d$ is still labelled as $u := (u_1, \ldots, u_d)$. We set

$$n := d - 1, \quad x = (x_1, \ldots, x_n), \quad x_0 := 1.$$

For a given rank $r$, denote the monomial sets

$$\mathscr{B}_0 \coloneqq \{x_1, \cdots, x_r\}, \quad \mathscr{B}_1 = \{x_i x_j : \, i \in [r], \; j \in [r+1, n]\}. \tag{5.7}$$

For a monomial power $\alpha \in \mathbb{N}^n$, by writing $\alpha \in \mathscr{B}_1$, we mean that $x^\alpha \in \mathscr{B}_1$. For each $\alpha \in \mathscr{B}_1$, one can write $\alpha = e_i + e_j$ with $i \in [r], \; j \in [r+1, n]$. Let $\mathbb{C}^{[r]\times\mathscr{B}_1}$ denote the space of matrices labelled by the pair $(k, \alpha) \in [r] \times \mathscr{B}_1$. For each $\alpha = e_i + e_j \in \mathscr{B}_1$ and $G \in \mathbb{C}^{[r]\times\mathscr{B}_1}$, denote the quadratic polynomial in $x$

$$\varphi_{ij}[G](x) \; \coloneqq \; \sum_{k=1}^{r} G(k, e_i + e_j) x_k - x_i x_j. \tag{5.8}$$

Suppose $r$ is the symmetric rank of $\mathcal{F}$. A matrix $G \in \mathbb{C}^{[r]\times\mathscr{B}_1}$ is called a *generating matrix* of $\mathcal{F}$ if each $\varphi_{ij}[G](x)$, with $\alpha = e_i + e_j \in \mathscr{B}_1$, is a generating polynomial of $\mathcal{F}$. Equivalently, $G$ is a generating matrix of $\mathcal{F}$ if and only if

$$\langle x_t \varphi_{ij}[G](x), \mathcal{F} \rangle = \sum_{k=1}^{r} G(k, e_i + e_j) \mathcal{F}_{0kt} - \mathcal{F}_{ijt} \; = \; 0, \quad t = 0, 1, \ldots, n, \tag{5.9}$$

for all $i \in [r], \; j \in [r+1, n]$. The notion *generating matrix* is motivated from that the entire tensor $\mathcal{F}$ can be recursively determined by $G$ and its first $r$ entries (see [109]). The existence and uniqueness of the generating matrix $G$ is shown as follows.

**Theorem 5.1** ( [59]). *Suppose $\mathcal{F}$ has the decomposition*

$$\mathcal{F} = \lambda_1 \begin{bmatrix} 1 \\ u_1 \end{bmatrix}^{\otimes 3} + \cdots + \lambda_r \begin{bmatrix} 1 \\ u_r \end{bmatrix}^{\otimes 3}, \tag{5.10}$$

*for vectors $u_i \in \mathbb{C}^n$ and scalars $0 \neq \lambda_i \in \mathbb{C}$. If the subvectors $(u_1)_{1:r}, \ldots, (u_r)_{1:r}$ are linearly independent, then there exists a unique generating matrix $G \in \mathbb{C}^{[r]\times\mathscr{B}_1}$ satisfying (5.9) for the tensor $\mathcal{F}$.*

*Proof.* We first prove the existence. For each $i = 1, \ldots, r$, denote the vectors $v_i = (u_i)_{1:r}$. Under the given assumption, $V := [v_1 \ \ldots \ v_r]$ is an invertible matrix. For each $l = r+1, \ldots, n$, let

$$N_l \ := \ V \cdot \mathrm{diag}\big((u_1)_l, \ldots, (u_r)_l\big) \cdot V^{-1}. \tag{5.11}$$

Then $N_l v_i = (u_i)_l v_i$ for $i = 1, \ldots, r$, i.e., $N_l$ has eigenvalues $(u_1)_l, \ldots, (u_r)_l$ with corresponding eigenvectors $(u_1)_{1:r}, \ldots, (u_r)_{1:r}$. We select $G \in \mathbb{C}^{[r] \times \mathscr{B}_1}$ to be the matrix such that

$$N_l = \begin{bmatrix} G(1, e_1 + e_l) & \cdots & G(r, e_1 + e_l) \\ \vdots & \ddots & \vdots \\ G(1, e_r + e_l) & \cdots & G(r, e_r + e_l) \end{bmatrix}, \ l = r+1, \ldots, n. \tag{5.12}$$

For each $s = 1, \ldots, r$ and $\alpha = e_i + e_j \in \mathbb{B}_1$ with $i \in [r], \ j \in [r+1, n]$,

$$\varphi_{ij}[G](u_s) = \sum_{k=1}^{r} G(k, e_i + e_j)(u_s)_k - (u_s)_i (u_s)_j \ = \ 0.$$

For each $t = 1, \ldots, n$, it holds that

$$\begin{aligned}
\langle x_t \varphi_{ij}[G](x), \mathcal{F} \rangle &= \left\langle \sum_{k=1}^{r} G(k, e_i + e_j) x_t x_k - x_t x_i x_j, \mathcal{F} \right\rangle \\
&= \left\langle \sum_{k=1}^{r} G(k, e_i + e_j) x_t x_k - x_t x_i x_j, \sum_{s=1}^{r} \lambda_s \begin{bmatrix} 1 \\ u_s \end{bmatrix}^{\otimes 3} \right\rangle \\
&= \sum_{k=1}^{r} G(k, e_i + e_j) \sum_{s=1}^{r} \lambda_s (u_s)_t (u_s)_k - \sum_{s=1}^{r} \lambda_s (u_s)_t (u_s)_i (u_s)_j \\
&= \sum_{s=1}^{r} \lambda_s (u_s)_t \left( \sum_{k=1}^{r} G(k, e_i + e_j)(u_s)_k - (u_s)_i (u_s)_j \right) \\
&= 0.
\end{aligned}$$

When $t = 0$, we can similarly get

$$\begin{aligned}
\langle \varphi_{ij}[G](x), \mathcal{F} \rangle &= \left\langle \sum_{k=1}^{r} G(k, e_i + e_j) x_k - x_i x_j, \mathcal{F} \right\rangle \\
&= \sum_{s=1}^{r} \lambda_s \left( \sum_{k=1}^{r} G(k, e_i + e_j)(u_s)_k - (u_s)_i (u_s)_j \right) \\
&= 0.
\end{aligned}$$

Therefore, the matrix $G$ satisfies (5.9) and it is a generating matrix for $\mathcal{F}$.

Second, we prove the uniqueness of such $G$. For each $\alpha = e_i + e_j \in \mathscr{B}_1$, let

$$F := \begin{bmatrix} \mathcal{F}_{011} & \cdots & \mathcal{F}_{0r1} \\ \vdots & \ddots & \vdots \\ \mathcal{F}_{01n} & \cdots & \mathcal{F}_{0rn} \end{bmatrix}, \; g_{ij} := \begin{bmatrix} \mathcal{F}_{1ij} \\ \vdots \\ \mathcal{F}_{nij} \end{bmatrix}.$$

Since $G$ satisfies (5.9), we have $F \cdot G(:, e_i + e_j) = g_{ij}$. The decomposition (5.10) implies that

$$F = \begin{bmatrix} u_1 & \cdots & u_r \end{bmatrix} \cdot \operatorname{diag}(\lambda_1, \ldots, \lambda_r) \cdot \begin{bmatrix} v_1 & \cdots & v_r \end{bmatrix}^T.$$

The sets $\{v_1, \ldots, v_r\}$ and $\{u_1, \ldots, u_r\}$ are both linearly independent. Since each $\lambda_i \neq 0$, the matrix $F$ has full column rank. Hence, the generating matrix $G$ satisfying $F \cdot G(:, e_i + e_j) = g_{ij}$ for all $i \in [r], j \in [r+1, n]$ is unique. □

The following is an example of generating matrices.

**Example 5.2.** *Consider the tensor $\mathcal{F} \in \mathbb{S}^3(\mathbb{C}^6)$ that is given as*

$$\mathcal{F} = 0.4 \cdot (1,1,1,1,1,1)^{\otimes 3} + 0.6 \cdot (1,-1,2,-1,2,3)^{\otimes 3}.$$

*The rank $r = 2$, $\mathscr{B}_0 = \{x_1, x_2\}$ and $\mathscr{B}_1 = \{x_1x_3, x_1x_4, x_1x_5, x_2x_3, x_2x_4, x_2x_5\}$. We have the vectors*

$$u_1 = (1,1,1,1,1), \quad u_2 = (-1,2,-1,2,3), \quad v_1 = (1,1), \quad v_2 = (-1,2).$$

*The matrices $N_3$, $N_4$, $N_5$ as in (5.11) are*

$$N_3 = \begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}^{-1} = \begin{bmatrix} 1/3 & 2/3 \\ 4/3 & -1/3 \end{bmatrix},$$
$$N_4 = \begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}^{-1} = \begin{bmatrix} 4/3 & -1/3 \\ -2/3 & 5/3 \end{bmatrix},$$
$$N_5 = \begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \begin{bmatrix} 1 & -1 \\ 1 & 2 \end{bmatrix}^{-1} = \begin{bmatrix} 5/3 & -2/3 \\ -4/3 & 7/3 \end{bmatrix}.$$

*The entries of the generating matrix $G$ are listed as below:*

| $k \backslash (i,j)$ | $(1,3)$ | $(1,4)$ | $(1,5)$ | $(2,3)$ | $(2,4)$ | $(2,5)$ |
|---|---|---|---|---|---|---|
| 1 | 1/3 | 4/3 | 5/3 | 4/3 | −2/3 | −4/3 |
| 2 | 2/3 | −1/3 | −2/3 | −1/3 | 5/3 | 7/3 |

(5.13)

*The generating polynomials in (5.8) are*

$$\varphi_{13}[G](x) = \frac{1}{3}x_1 + \frac{2}{3}x_2 - x_1x_3, \quad \varphi_{23}[G](x) = \frac{4}{3}x_1 - \frac{1}{3}x_2 - x_2x_3,$$
$$\varphi_{14}[G](x) = \frac{4}{3}x_1 - \frac{1}{3}x_2 - x_1x_4, \quad \varphi_{24}[G](x) = -\frac{2}{3}x_1 + \frac{5}{3}x_2 - x_2x_4,$$
$$\varphi_{15}[G](x) = \frac{5}{3}x_1 - \frac{2}{3}x_2 - x_1x_5, \quad \varphi_{25}[G](x) = -\frac{4}{3}x_1 + \frac{7}{3}x_2 - x_2x_5.$$

*Above generating polynomials can be written in the following form*

$$\begin{bmatrix}\varphi_{1j}[G](x)\\ \varphi_{2j}[G](x)\end{bmatrix} = N_j \begin{bmatrix}x_1\\ x_2\end{bmatrix} - x_j \begin{bmatrix}x_1\\ x_2\end{bmatrix}, \text{ for } j = 3, 4, 5.$$

*For $x$ to be a common zero of $\varphi_{1j}[G](x)$ and $\varphi_{2j}[G](x)$, it requires that $(x_1, x_2)$ is an eigenvector of $N_j$ with the corresponding eigenvalue $x_j$.*

We show how to find an incomplete tensor decomposition (5.10) for $\mathcal{F}$ when only its subtensor $\mathcal{F}_\Omega$ is given, where the label set $\Omega$ is as in (5.5). Suppose that there exists the decomposition (5.10) for $\mathcal{F}$, for vectors $u_i \in \mathbb{C}^n$ and nonzero scalars $\lambda_i \in \mathbb{C}$. Assume the subvectors $(u_1)_{1:r}, \ldots, (u_r)_{1:r}$ are linearly independent, so there is a unique generating matrix $G$ for $\mathcal{F}$, by Theorem 5.1.

For each $\alpha = e_i + e_j \in \mathscr{B}_1$ with $i \in [r], j \in [r+1, n]$ and for each

$$l = r+1, \ldots, j-1, j+1, \ldots, n,$$

the generating matrix $G$ satisfies the equations

$$\left\langle x_l \left( \sum_{k=1}^{r} G(k, e_i + e_j) x_k - x_i x_j \right), \mathcal{F} \right\rangle = \sum_{k=1}^{r} G(k, e_i + e_j)\mathcal{F}_{0kl} - \mathcal{F}_{ijl} = 0. \tag{5.14}$$

Let the matrix $A_{ij}[\mathcal{F}] \in \mathbb{C}^{(n-r-1)\times r}$ and the vector $b_{ij}[\mathcal{F}] \in \mathbb{C}^{n-r-1}$ be such that

$$A_{ij}[\mathcal{F}] := \begin{bmatrix} \mathcal{F}_{0,1,r+1} & \cdots & \mathcal{F}_{0,r,r+1} \\ \vdots & \ddots & \vdots \\ \mathcal{F}_{0,1,j-1} & \cdots & \mathcal{F}_{0,r,j-1} \\ \mathcal{F}_{0,1,j+1} & \cdots & \mathcal{F}_{0,r,j+1} \\ \vdots & \ddots & \vdots \\ \mathcal{F}_{0,1,n} & \cdots & \mathcal{F}_{0,r,n} \end{bmatrix}, \quad b_{ij}[\mathcal{F}] := \begin{bmatrix} \mathcal{F}_{i,j,r+1} \\ \vdots \\ \mathcal{F}_{i,j,j-1} \\ \mathcal{F}_{i,j,j+1} \\ \vdots \\ \mathcal{F}_{i,j,n} \end{bmatrix}. \tag{5.15}$$

To distinguish changes in the labels of tensor entries of $\mathcal{F}$, the commas are inserted to separate labeling numbers.

The equations in (5.14) can be equivalently written as

$$A_{ij}[\mathcal{F}] \cdot G(:, e_i + e_j) \;=\; b_{ij}[\mathcal{F}]. \tag{5.16}$$

If the rank $r \leq \frac{d}{2} - 1$, then $n - r - 1 = d - r - 2 \geq r$. Thus, the number of rows is not less than the number of columns for matrices $A_{ij}[\mathcal{F}]$. If $A_{ij}[\mathcal{F}]$ has linearly independent columns, then (5.16) uniquely determines $G(:, \alpha)$. For such a case, the matrix $G$ can be fully determined by the linear system (5.16). Let $N_{r+1}(G), \ldots, N_m(G) \in \mathbb{C}^{r \times r}$ be the matrices given as

$$N_l(G) = \begin{bmatrix} G(1, e_1 + e_l) & \cdots & G(r, e_1 + e_l) \\ \vdots & \ddots & \vdots \\ G(1, e_r + e_l) & \cdots & G(r, e_r + e_l) \end{bmatrix}, \; l = r+1, \ldots, n. \tag{5.17}$$

As in the proof of Theorem 5.1, one can see that

$$N_l(G) \begin{bmatrix} (u_i)_1 \\ \vdots \\ (u_i)_r \end{bmatrix} = (u_i)_l \cdot \begin{bmatrix} (u_i)_1 \\ \vdots \\ (u_i)_r \end{bmatrix}, \; l = r+1, \ldots, n. \tag{5.18}$$

The above is equivalent to the equations

$$N_l(G) v_i \;=\; (w_i)_{l-r} \cdot v_i, \quad l = r+1, \ldots, n,$$

for the vectors $(i = 1, \ldots, r)$

$$v_i \;:=\; (u_i)_{1:r}, \quad w_i \;:=\; (u_i)_{r+1:n}. \tag{5.19}$$

Each $v_i$ is a common eigenvector of the matrices $N_{r+1}(G), \ldots, N_n(G)$ and $(w_i)_{l-r}$ is the associated eigenvalue of $N_l(G)$. These matrices may or may not have repeated eigenvalues. Therefore, we select a generic vector $\xi \coloneqq (\xi_{r+1}, \cdots, \xi_n)$ and let

$$N(\xi) \;\coloneqq\; \xi_{r+1} N_{r+1} + \cdots + \xi_n N_n. \tag{5.20}$$

The eigenvalues of $N(\xi)$ are $\xi^T w_1, \ldots, \xi^T w_r$. When $w_1, \ldots, w_r$ are distinct from each other and $\xi$ is generic, the matrix $N(\xi)$ does not have a repeated eigenvalue and hence it has unique eigenvectors $v_1, \ldots, v_r$, up to scaling. Let $\tilde{v}_1, \ldots, \tilde{v}_r$ be unit length eigenvectors of $N(\xi)$. They are also common eigenvectors of $N_{r+1}(G)$, …, $N_n(G)$. For each $i = 1, \ldots, r$, let $\tilde{w}_i$ be the vector such that its $j$th entry $(\tilde{w}_i)_j$ is the eigenvalue of $N_{j+r}(G)$, associated to the eigenvector $\tilde{v}_i$, or equivalently,

$$\tilde{w}_i = (\tilde{v}_i^H N_{r+1}(G) \tilde{v}_i, \cdots, \tilde{v}_i^H N_n(G) \tilde{v}_i) \quad i = 1, \ldots, r. \tag{5.21}$$

Up to a permutation of $(\tilde{v}_1, \ldots, \tilde{v}_r)$, there exist scalars $\gamma_i$ such that

$$v_i = \gamma_i \tilde{v}_i, \quad w_i = \tilde{w}_i. \tag{5.22}$$

The tensor decomposition of $\mathcal{F}$ can also be written as

$$\mathcal{F} = \lambda_1 \begin{bmatrix} 1 \\ \gamma_1 \tilde{v}_1 \\ \tilde{w}_1 \end{bmatrix}^{\otimes 3} + \cdots + \lambda_r \begin{bmatrix} 1 \\ \gamma_r \tilde{v}_r \\ \tilde{w}_r \end{bmatrix}^{\otimes 3}.$$

The scalars $\lambda_1, \cdots, \lambda_r$ and $\gamma_1, \cdots, \gamma_r$ satisfy the linear equations

$$\begin{aligned} \lambda_1 \gamma_1 \tilde{v}_1 \otimes \tilde{w}_1 + \cdots + \lambda_r \gamma_r \tilde{v}_r \otimes \tilde{w}_r &= \mathcal{F}_{[0,1:r,r+1:n]}, \\ \lambda_1 \gamma_1^2 \tilde{v}_1 \otimes \tilde{v}_1 \otimes \tilde{w}_1 + \cdots + \lambda_r \gamma_r^2 \tilde{v}_r \otimes \tilde{v}_r \otimes \tilde{w}_r &= \mathcal{F}_{[1:r,1:r,r+1:n]}. \end{aligned}$$

Denote the label sets

$$\boxed{\begin{aligned} J_1 &\coloneqq \big\{(0, i_1, i_2) : \, i_1 \in [r], \, i_2 \in [r+1, n]\big\}, \\ J_2 &\coloneqq \big\{(i_1, i_2, i_3) : \, i_1 \neq i_2, \, i_1, i_2 \in [r], i_3 \in [r+1, n]\big\}. \end{aligned}} \tag{5.23}$$

To determine the scalars $\lambda_i, \gamma_i$, we can solve the linear least squares

$$\min_{(\beta_1, \ldots, \beta_r)} \left\| \mathcal{F}_{J_1} - \sum_{i=1}^{r} \beta_i \cdot \tilde{v}_i \otimes \tilde{w}_i \right\|^2, \tag{5.24}$$

$$\min_{(\theta_1, \ldots, \theta_r)} \left\| \mathcal{F}_{J_2} - \sum_{k=1}^{r} \theta_k \cdot (\tilde{v}_k \otimes \tilde{v}_k \otimes \tilde{w}_i)_{J_2} \right\|^2. \tag{5.25}$$

Let $(\beta_1^*, \ldots, \beta_r^*)$, $(\theta_1^*, \ldots, \theta_r^*)$ be minimizers of (5.24) and (5.25) respectively. Then, for each $i = 1, \ldots, r$, let

$$\lambda_i := (\beta_i^*)^2 / \theta_i^*, \quad \gamma_i := \theta_i^* / \beta_i^*. \tag{5.26}$$

For the vectors $(i = 1, \ldots, r)$

$$p_i := \sqrt[3]{\lambda_i}(1, \gamma_i \tilde{v}_i, \tilde{w}_i),$$

the sum $p_1^{\otimes 3} + \cdots + p_r^{\otimes 3}$ is a tensor decomposition for $\mathcal{F}$. This is justified in the following theorem.

**Theorem 5.3** ( [59]). *Suppose the tensor $\mathcal{F}$ has the decomposition as in (5.10). Assume that the vectors $v_1, \ldots, v_r$ are linearly independent and the vectors $w_1, \ldots, w_r$ are distinct from each other, where $v_1, \ldots, v_r, w_1, \ldots, w_r$ are defined as in (5.19). Let $\xi$ be a generically chosen coefficient vector and let $p_1, \ldots, p_r$ be the vectors produced as above. Then, the tensor decomposition $\mathcal{F} = p_1^{\otimes 3} + \cdots + p_r^{\otimes 3}$ is unique.*

*Proof.* Since $v_1, \ldots, v_r$ are linearly independent, the tensor decomposition (5.10) is unique, up to scalings and permutations. By Theorem 5.1, there is a unique generating matrix $G$ for $\mathcal{F}$ satisfying (5.9). Under the given assumptions, the equation (5.16) uniquely determines $G$. Note that $\xi^T w_1, \ldots, \xi^T w_r$ are the eigenvalues of $N(\xi)$ and $v_1, \ldots, v_r$ are the corresponding eigenvectors. When the vector $\xi$ is generically chosen, the values of $\xi^T w_1, \ldots, \xi^T w_r$ are distinct eigenvalues of $N(\xi)$. So $N(\xi)$ has unique eigenvalue decompositions, and hence (5.22) must hold, up to a permutation of $(v_1, \ldots, v_r)$. Since the coefficient matrices have full column ranks, the linear least squares problems have unique optimal solutions. Up to a permutation of $p_1, \ldots, p_r$, it holds that $p_i = \sqrt[3]{\lambda_i} \begin{bmatrix} 1 \\ u_i \end{bmatrix}$. Then, the conclusion follows readily. □

The following is the algorithm for computing an incomplete tensor decomposition for $\mathcal{F}$ when only its subtensor $\mathcal{F}_\Omega$ is given.

**Algorithm 5.4.** *(Incomplete symmetric tensor decompositions.)*

*Input: A third order symmetric subtensor $\mathcal{F}_\mathcal{I}$ and a rank $r = \text{rank}_S(\mathcal{F}) \leq \frac{d}{2} - 1$.*

1. *Determine the matrix $G$ by solving (5.16) for each $\alpha = e_i + e_j \in \mathbb{B}_1$.*
2. *Let $N(\xi)$ be the matrix as in (5.20), for a randomly selected vector $\xi$. Compute the unit length eigenvectors $\tilde{v}_1, \ldots, \tilde{v}_r$ of $N(\xi)$ and choose $\tilde{w}_i$ as in (5.21).*
3. *Solve the linear least squares (5.24) and (5.25) to get the coefficients $\lambda_i, \gamma_i$ as in (5.26).*
4. *For each $i = 1, \ldots, r$, let $p_i := \sqrt[3]{\lambda_i}(1, \gamma_i \tilde{v}_i, \tilde{w}_i)$.*

*Output: The tensor decomposition $\mathcal{F} = (p_1)^{\otimes 3} + \cdots + (p_r)^{\otimes 3}$.*

The following is an example of applying Algorithm 5.4.

**Example 5.5.** *Consider the same tensor $\mathcal{F}$ as in Example 5.2. The monomial sets $\mathscr{B}_0$, $\mathscr{B}_1$ are the same. The matrices $A_{ij}[\mathcal{F}]$ and vectors $b_{ij}[\mathcal{F}]$ are*

$$A_{13}[\mathcal{F}] = A_{23}[\mathcal{F}] = \begin{bmatrix} -0.8 & 2.8 \\ -1.4 & 4 \end{bmatrix}, \quad b_{13}[\mathcal{F}] = \begin{bmatrix} 1.6 \\ 2.2 \end{bmatrix}, b_{23}[\mathcal{F}] = \begin{bmatrix} -2 \\ -3.2 \end{bmatrix},$$

$$A_{14}[\mathcal{F}] = A_{24}[\mathcal{F}] = \begin{bmatrix} 1 & -0.8 \\ -1.4 & 4 \end{bmatrix}, \quad b_{14}[\mathcal{F}] = \begin{bmatrix} 1.6 \\ -3.2 \end{bmatrix}, b_{24}[\mathcal{F}] = \begin{bmatrix} -2 \\ 7.6 \end{bmatrix},$$

$$A_{15}[\mathcal{F}] = A_{25}[\mathcal{F}] = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 2.8 \end{bmatrix}, \quad b_{15}[\mathcal{F}] = \begin{bmatrix} 2.2 \\ -3.2 \end{bmatrix}, b_{25}[\mathcal{F}] = \begin{bmatrix} -3.2 \\ 7.6 \end{bmatrix}.$$

*Solve (5.16) to obtain $G$, which is same as in (5.13). The matrices $N_3(G), N_4(G), N_5(G)$ are*

$$N_3(G) = \begin{bmatrix} 1/3 & 2/3 \\ 4/3 & -1/3 \end{bmatrix}, \; N_4(G) = \begin{bmatrix} 4/3 & -1/3 \\ -2/3 & 5/3 \end{bmatrix}, \; N_5(G) = \begin{bmatrix} 5/3 & -2/3 \\ -4/3 & 7/3 \end{bmatrix}.$$

*Choose a generic $\xi$, say, $\xi = (3, 4, 5)$, then*

$$N(\xi) = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{5} \\ 1/\sqrt{2} & 2/\sqrt{5} \end{bmatrix} \begin{bmatrix} 12 & 0 \\ 0 & 20 \end{bmatrix} \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{5} \\ 1/\sqrt{2} & 2/\sqrt{5} \end{bmatrix}^{-1}.$$

*The unit length eigenvectors are*

$$\tilde{v}_1 = (1/\sqrt{2}, 1/\sqrt{2}), \quad \tilde{v}_2 = (-1/\sqrt{5}, 2/\sqrt{5}).$$

*As in (5.21), we get the vectors*

$$w_1 = (1, 1, 1), \; w_2 = (-1, 2, 3).$$

*Solving (5.24) and (5.25), we get the scalars*

$$\gamma_1 = \sqrt{2}, \quad \gamma_2 = \sqrt{5}, \quad \lambda_1 = 0.4, \quad \lambda_2 = 0.6.$$

*This produces the decomposition $\mathcal{F} = \lambda_1 u_1^{\otimes 3} + \lambda_2 u_2^{\otimes 3}$ for the vectors*

$$u_1 = (1, \gamma_1 v_1, w_1) = (1, 1, 1, 1, 1, 1), \quad u_2 = (1, \gamma_2 v_2, w_2) = (1, -1, 2, -1, 2, 3).$$

*Remark.* Algorithm 5.4 requires the value of $r$. This is generally a hard question. In computational practice, one can estimate the value of $r$ as follows. Let $\mathrm{Flat}(\mathcal{F}) \in \mathbb{C}^{(n+1)\times(n+1)^2}$ be the flattening matrix, labelled by $(i, (j, k))$ such that

$$\mathrm{Flat}(\mathcal{F})_{i,(j,k)} \; = \; \mathcal{F}_{ijk}$$

for all $i, j, k = 0, 1, \ldots, n$. The rank of $\mathrm{Flat}(\mathcal{F})$ equals the rank of $\mathcal{F}$ when the vectors $p_1, \ldots, p_r$ are linearly independent. The rank of $\mathrm{Flat}(\mathcal{F})$ is not available since only the subtensor $(\mathcal{F})_\Omega$ is known. However, we can calculate the ranks of submatrices of $(\mathcal{F})_\Omega$ whose entries are known. If the tensor $\mathcal{F}$ as in (5.10) is such that both the sets $\{v_1, \ldots, v_r\}$ and $\{w_1, \ldots, w_r\}$ are linearly independent, one can see that $\sum_{i=1}^r \lambda_i v_i w_i^T$ is a known submatrix

of Flat($\mathcal{F}$) whose rank is $r$. This is generally the case if $r \leq \frac{d}{2} - 1$, since $v_i$ has the length $r$ and $w_i$ has length $d - 1 - r \geq r$. Therefore, the known submatrices of Flat($\mathcal{F}$) are generally sufficient to estimate $\text{rank}_S(\mathcal{F})$. For instance, we consider the case $\mathcal{F} \in \text{S}^3(\mathbb{C}^7)$. The flattening matrix Flat($\mathcal{F}$) is

$$\begin{bmatrix} * & * & * & * & * & * & * \\ * & * & \mathcal{F}_{120} & \mathcal{F}_{130} & \mathcal{F}_{140} & \mathcal{F}_{150} & \mathcal{F}_{160} \\ * & \mathcal{F}_{210} & * & \mathcal{F}_{230} & \mathcal{F}_{240} & \mathcal{F}_{250} & \mathcal{F}_{260} \\ * & \mathcal{F}_{310} & \mathcal{F}_{320} & * & \mathcal{F}_{340} & \mathcal{F}_{350} & \mathcal{F}_{360} \\ * & \mathcal{F}_{410} & \mathcal{F}_{420} & \mathcal{F}_{430} & * & \mathcal{F}_{450} & \mathcal{F}_{460} \\ * & \mathcal{F}_{510} & \mathcal{F}_{520} & \mathcal{F}_{530} & \mathcal{F}_{540} & * & \mathcal{F}_{560} \\ * & \mathcal{F}_{610} & \mathcal{F}_{620} & \mathcal{F}_{630} & \mathcal{F}_{640} & \mathcal{F}_{650} & * \end{bmatrix}, \tag{5.27}$$

where each $*$ means that entry is not given. The largest submatrices with known entries are

$$\begin{bmatrix} \mathcal{F}_{410} & \mathcal{F}_{420} & \mathcal{F}_{430} \\ \mathcal{F}_{510} & \mathcal{F}_{520} & \mathcal{F}_{530} \\ \mathcal{F}_{610} & \mathcal{F}_{620} & \mathcal{F}_{630} \end{bmatrix}, \quad \begin{bmatrix} \mathcal{F}_{140} & \mathcal{F}_{150} & \mathcal{F}_{160} \\ \mathcal{F}_{240} & \mathcal{F}_{250} & \mathcal{F}_{260} \\ \mathcal{F}_{340} & \mathcal{F}_{350} & \mathcal{F}_{360} \end{bmatrix}.$$

The rank of above matrices generally equals $\text{rank}_S(\mathcal{F})$ if $r \leq \frac{d}{2} - 1 = 2.5$.

## 5.3 Tensor approximations

In some applications, we do not have the subtensor $\mathcal{F}_\Omega$ exactly but only have an approximation $\widehat{\mathcal{F}}_\Omega$ for it. The Algorithm 5.4 can still provide a good rank-$r$ approximation for $\mathcal{F}$ when it is applied to $\widehat{\mathcal{F}}_\Omega$. We define the matrix $A_{ij}[\widehat{\mathcal{F}}]$ and the vector $b_{ij}[\widehat{\mathcal{F}}]$ in the same way as in (5.15), for each $\alpha = e_i + e_j \in \mathscr{B}_1$. The generating matrix $G$ for $\mathcal{F}$ can be approximated by solving the linear least squares

$$\min_{g \in \mathbb{C}^r} \quad \|A_{ij}[\widehat{\mathcal{F}}] \cdot g - b_{ij}[\widehat{\mathcal{F}}]\|^2, \tag{5.28}$$

for each $\alpha = e_i + e_j \in \mathbb{B}_1$. Let $\widehat{G}(:, e_i + e_j)$ be the optimizer of the above and $\widehat{G}$ be the matrix consisting of all such $\widehat{G}(:, e_i + e_j)$. Then $\widehat{G}$ is an approximation for $G$. For each $l = r + 1, \ldots, n$, define the matrix $N_l(\widehat{G})$ similarly as in (5.17). Choose a generic vector $\xi = (\xi_{r+1}, \ldots, \xi_n)$ and let

$$\widehat{N}(\xi) \coloneqq \xi_{r+1} N_{r+1}(\widehat{G}) + \cdots + \xi_n N_n(\widehat{G}). \tag{5.29}$$

The matrix $\widehat{N}(\xi)$ is an approximation for $N(\xi)$. Let $\hat{v}_1, \ldots, \hat{v}_r$ be unit length eigenvectors of $\widehat{N}(\xi)$. For $k = 1, \ldots, r$, let

$$\hat{w}_k := \big((\hat{v}_k)^H N_{r+1}(\widehat{G})\hat{v}_k, \ldots, (\hat{v}_k)^H N_n(\widehat{G})\hat{v}_k\big). \tag{5.30}$$

For the label sets $J_1, J_2$ as in (5.23), the subtensors $\widehat{\mathcal{F}}_{J_1}, \widehat{\mathcal{F}}_{J_2}$ are similarly defined like $\mathcal{F}_{J_1}, \mathcal{F}_{J_2}$. Consider the following linear least square problems

$$\min_{(\beta_1,\ldots,\beta_r)} \left\| \widehat{\mathcal{F}}_{J_1} - \sum_{k=1}^{r} \beta_k \cdot \hat{v}_k \otimes \hat{w}_k \right\|^2, \tag{5.31}$$

$$\min_{(\theta_1,\ldots,\theta_r)} \left\| \widehat{\mathcal{F}}_{J_2} - \sum_{k=1}^{r} \theta_i \cdot (\hat{v}_k \otimes \hat{v}_k \otimes \hat{w}_k)_{J_2} \right\|^2. \tag{5.32}$$

Let $(\hat{\beta}_1, \ldots, \hat{\beta}_r)$ and $(\hat{\theta}_1, \ldots, \hat{\theta}_r)$ be their optimizers respectively. For each $k = 1, \ldots, r$, let

$$\hat{\lambda}_k := (\hat{\beta}_k)^2/\hat{\theta}_k, \quad \hat{\gamma}_k := \hat{\theta}_k/\hat{\beta}_k. \tag{5.33}$$

This results in the tensor approximation

$$\mathcal{F} \approx (\hat{p}_1)^{\otimes 3} + \cdots + (\hat{p}_r)^{\otimes 3},$$

for the vectors $\hat{p}_k := \sqrt[3]{\hat{\lambda}_k}(1, \hat{\gamma}_k\hat{v}_k, \hat{w}_k)$.

The above may not give an optimal tensor approximation. To get an improved one, we can use $\hat{p}_1, \ldots, \hat{p}_r$ as starting points to solve the following nonlinear optimization

$$\min_{(q_1,\ldots,q_r)} \left\| \left( \sum_{k=1}^{r} (q_k)^{\otimes 3} - \widehat{\mathcal{F}} \right)_{\mathcal{I}} \right\|^2. \tag{5.34}$$

The minimizer of the optimization (5.34) is denoted as $(p_1^*, \ldots, p_r^*)$.

Summarizing the above, we have the following algorithm for computing a tensor approximation.

**Algorithm 5.6.** *(Incomplete symmetric tensor approximations.)*

*Input: A third order symmetric subtensor $\widehat{\mathcal{F}}_{\mathcal{I}}$ and a rank $r \leq \frac{d}{2} - 1$.*

1. *Find the matrix $\widehat{G}$ by solving (5.28) for each $\alpha = e_i + e_j \in \mathbb{B}_1$.*
2. *Choose a generic vector and let $\widehat{N}(\xi)$ be the matrix as in (5.29). Compute unit length eigenvectors $\hat{v}_1, \ldots, \hat{v}_r$ for $\widehat{N}(\xi)$ and define $\hat{w}_i$ in (5.30).*

3. *Solve the linear least squares (5.31), (5.32) to get the coefficients* $\hat{\lambda}_i, \hat{\gamma}_i$.

4. *For each* $i = 1, \ldots, r$, *let* $\hat{p}_i := \sqrt[3]{\hat{\lambda}_i}(1, \hat{\gamma}_i \hat{v}_i, \hat{w}_i)$. *Then* $(\hat{p}_1)^{\otimes 3} + \cdots + (\hat{p}_r)^{\otimes 3}$ *is a tensor approximation for* $\widehat{\mathcal{F}}$.

5. *Use* $\hat{p}_1, \ldots, \hat{p}_r$ *as starting points to solve the nonlinear optimization (5.34) for an optimizer* $(p_1^*, \ldots, p_r^*)$.

*Output: The tensor approximation* $(p_1^*)^{\otimes 3} + \cdots + (p_r^*)^{\otimes 3}$ *for* $\widehat{\mathcal{F}}$.

When $\widehat{\mathcal{F}}$ is close to $\mathcal{F}$, Algorithm 5.6 also produces a good rank-$r$ tensor approximation for $\mathcal{F}$. This is shown in the following.

**Theorem 5.7** ( [59]). *Suppose the tensor* $\mathcal{F} = (p_1)^{\otimes 3} + \cdots + (p_r)^{\otimes 3}$, *with* $r \leq \frac{d}{2} - 1$, *satisfies the following conditions:*

*(i) The leading entry of each* $p_i$ *is nonzero;*

*(ii) the subvectors* $(p_1)_{2:r+1}, \ldots, (p_r)_{2:r+1}$ *are linearly independent;*

*(iii) the subvectors* $(p_1)_{[r+2:j,j+2:d]}, \ldots, (p_r)_{[r+2:j,j+2:d]}$ *are linearly independent for each* $j \in [r+1, n]$;

*(iv) the eigenvalues of the matrix* $N(\xi)$ *in (5.20) are distinct from each other.*

*Let* $\hat{p}_i, p_i^*$ *be the vectors produced by Algorithm 5.6. If the distance* $\epsilon := \|(\widehat{\mathcal{F}} - \mathcal{F})_{\mathcal{I}}\|$ *is small enough, then there exist scalars* $\hat{\tau}_i, \tau_i^*$ *such that*

$$(\hat{\tau}_i)^3 = (\tau_i^*)^3 = 1, \quad \|\hat{\tau}_i \hat{p}_i - p_i\| = O(\epsilon), \quad \|\tau_i^* p_i^* - p_i\| = O(\epsilon),$$

*up to a permutation of* $(p_1, \ldots, p_r)$, *where the constants inside* $O(\cdot)$ *only depend on* $\mathcal{F}$ *and the choice of* $\xi$ *in Algorithm 5.6.*

*Proof.* The conditions (i)-(ii), by Theorem 5.1, imply that there is a unique generating matrix $G$ for $\mathcal{F}$. The matrix $G$ can be approximated by solving the linear least square problems (5.28). Note that

$$\|A_{ij}[\widehat{\mathcal{F}}] - A_{ij}[\mathcal{F}]\| \leq \epsilon, \quad \|b_{ij}[\widehat{\mathcal{F}}] - b_{ij}[\mathcal{F}]\| \leq \epsilon,$$

for all $\alpha = e_i + e_j \in \mathscr{B}_1$. The matrix $A_{ij}[\mathcal{F}]$ can be written as

$$A_{ij}[\mathcal{F}] = [(p_1)_{[r+2:j,j+2:d]}, \ldots, (p_r)_{[r+2:j,j+2:d]}] \cdot [(p_1)_{2:r+1}, \ldots, (p_r)_{2:r+1}]^T.$$

By the conditions (ii)-(iii), the matrix $A_{ij}[\mathcal{F}]$ has full column rank for each $j \in [r+1, n]$ and hence the matrix $A_{ij}[\widehat{\mathcal{F}}]$ has full column rank when $\epsilon$ is small enough. Therefore, the linear least problems (5.28) have unique solutions and the solution $\widehat{G}$ satisfies that

$$\|\widehat{G} - G\| = O(\epsilon),$$

where $O(\epsilon)$ depends on $\mathcal{F}$ (see [40, Theorem 3.4]). For each $j = r+1, \ldots, n$, $N_j(\widehat{G})$ is part of the generating matrix $\widehat{G}$, so

$$\|N_j(\widehat{G}) - N_j(G)\| \leq \|\widehat{G} - G\| = O(\epsilon), \quad j = r+1, \ldots, n.$$

This implies that $\|\widehat{N}(\xi) - N(\xi)\| = O(\epsilon)$. When $\epsilon$ is small enough, the matrix $\widehat{N}(\xi)$ does not have repeated eigenvalues, due to the condition (iv). Thus, the matrix $N(\xi)$ has a set of unit length eigenvectors $\tilde{v}_1, \ldots, \tilde{v}_r$ with eigenvalues $\tilde{w}_1, \ldots, \tilde{w}_r$ respectively, such that

$$\|\hat{v}_i - \tilde{v}_i\| = O(\epsilon), \quad \|\hat{w}_i - \tilde{w}_i\| = O(\epsilon).$$

This follows from Proposition 4.2.1 in [22]. The constants inside the above $O(\cdot)$ depend only on $\mathcal{F}$ and $\xi$. The $\tilde{w}_1, \ldots, \tilde{w}_r$ are scalar multiples of linearly independent vectors $(p_1)_{r+2:d}, \ldots, (p_r)_{r+2:d}$ respectively, so $\tilde{w}_1, \ldots, \tilde{w}_r$ are linearly independent. When $\epsilon$ is small, $\hat{w}_1, \ldots, \hat{w}_r$ are linearly independent as well. The scalars $\hat{\lambda}_i\hat{\gamma}_i$ and $\hat{\lambda}_i(\hat{\gamma}_i)^2$ are optimizers for the linear least square problems (5.31) and (5.32). By Theorem 3.4 in [40], we have

$$\|\hat{\lambda}_i\hat{\gamma}_i - \lambda_i\gamma_i\| = O(\epsilon),\ \|\hat{\lambda}_i(\hat{\gamma}_i)^2 - \lambda_i\gamma_i^2\| = O(\epsilon).$$

The vector $p_i$ can be written as $p_i = \sqrt[3]{\lambda_i}(1, \gamma_i\tilde{v}_i, \tilde{w}_i)$, so we must have $\lambda_i, \gamma_i \neq 0$ due to the condition (ii). Thus, it holds that

$$\|\hat{\lambda}_i - \lambda_i\| = O(\epsilon),\ \|\hat{\gamma}_i - \gamma_i\| = O(\epsilon),$$

where constants inside $O(\cdot)$ depend only on $\mathcal{F}$ and $\xi$. For the vectors $\tilde{p}_i := \sqrt[3]{\lambda_i}(1, \gamma_i\tilde{v}_i, \tilde{w}_i)$, we have $\mathcal{F} = \sum_{i=1}^r \tilde{p}_i^{\otimes 3}$, by Theorem 5.3. Since $p_1, \ldots, p_r$ are linearly independent by the assumption, the rank decomposition of $\mathcal{F}$ is unique up to scaling and permutation. There exist scalars $\hat{\tau}_i$ such that $(\hat{\tau}_i)^3 = 1$ and $\hat{\tau}_i\tilde{p}_i = p_i$, up to a permutation of $p_1, \ldots, p_r$. For $\hat{p}_i = \sqrt[3]{\hat{\lambda}_i}(1, \hat{\gamma}_i\hat{v}_i, \hat{w}_i)$, we have $\|\hat{\tau}_i\hat{p}_i - p_i\| = O(\epsilon)$, where the constants in $O(\cdot)$ only depend on $\mathcal{F}$ and $\xi$.

Since $\|\hat{\tau}_i\hat{p}_i - p_i\| = O(\epsilon)$, we have $\|(\sum_{i=1}^r (\hat{p}_i)^{\otimes 3} - \mathcal{F})_{\mathcal{I}}\| = O(\epsilon)$. The $(p_1^*, \ldots, p_r^*)$ is a minimizer of (5.34), so

$$\left\| \left( \sum_{i=1}^r (p_i^*)^{\otimes 3} - \widehat{\mathcal{F}} \right)_{\mathcal{I}} \right\| \leq \left\| \left( \sum_{i=1}^r (\hat{p}_i)^{\otimes 3} - \widehat{\mathcal{F}} \right)_{\mathcal{I}} \right\| = O(\epsilon).$$

For the tensor $\mathcal{F}^* := \sum_{i=1}^r (p_i^*)^{\otimes 3}$, we get

$$\|(\mathcal{F}^* - \mathcal{F})_{\mathcal{I}}\| \leq \|(\mathcal{F}^* - \widehat{\mathcal{F}})_{\mathcal{I}}\| + \|(\widehat{\mathcal{F}} - \mathcal{F})_{\mathcal{I}}\| = O(\epsilon).$$

When Algorithm 5.6 is applied to $(\mathcal{F}^*)_\Omega$, the Step 4 will give the exact decomposition $\mathcal{F}^* = \sum_{i=1}^r (p_i^*)^{\otimes 3}$. By repeating the previous argument, we can similarly show that $\|p_i - \tau_i^* p_i^*\| = O(\epsilon)$ for some $\tau_i^*$ such that $(\tau_i^*)^3 = 1$, where the constants in $O(\cdot)$ only depend on $\mathcal{F}$ and $\xi$. □

*Remark.* For the special case that $\epsilon = 0$, Algorithm 5.6 is the same as Algorithm 5.4, which produces the exact rank decomposition for $\mathcal{F}$. The conditions in Theorem 5.7 are satisfied for generic vectors $p_1, \ldots, p_r$, since $r \leq \frac{d}{2} - 1$. The constant in $O(\cdot)$ is not explicitly given in the proof. It is related to the condition number $\kappa(\mathcal{F})$ for tensor decomposition. It was shown by Breiding and Vannieuwenhoven [17] that

$$\sqrt{\sum_{i=1}^r \|p_i^{\otimes 3} - \hat{p}_i^{\otimes 3}\|^2} \leq \kappa(\mathcal{F})\|\mathcal{F} - \hat{\mathcal{F}}\| + c\epsilon^2$$

for some constant $c$. The continuity of $\hat{G}$ in $\hat{\mathcal{F}}$ is implicitly impled by the proof. Eigenvalues and unit eigenvectors of $\widehat{N}(\xi)$ are continuous in $\hat{G}$. Furthermore, $\hat{\lambda}_i, \hat{\gamma}_i$ are continuous in the eigenvalues and unit eigenvectors. All these functions are locally Lipschitz continuous. The $\hat{p}_i$ is Lipschitz continuous with respect to $\hat{\mathcal{F}}$, in a neighborhood of $\mathcal{F}$, which also implies an error bound for $\hat{p}_i$. The tensors $(p_i^*)^{\otimes 3}$ are also locally Lipschitz continuous in $\widehat{\mathcal{F}}$ illustrated by [18]. This also gives error bounds for decomposing vectors $p_i^*$. We refer to [17, 18] for more details about condition numbers of tensor decompositions.

**Example 5.8.** *We consider the same tensor $\mathcal{F}$ as in Example 5.2. The subtensor $(\mathcal{F})_\Omega$ is perturbed to $(\widehat{\mathcal{F}})_\Omega$. The perturbation is randomly generated from the Gaussian distribution $\mathcal{N}(0, 0.01)$. For neatness of the paper, we do not display $(\widehat{\mathcal{F}})_\Omega$ here. We use Algorithm 5.6 to compute the incomplete tensor approximation. The matrices $A_{ij}[\widehat{\mathcal{F}}]$ and vectors $b_{ij}[\widehat{\mathcal{F}}]$ are*

*given as follows:*

$$A_{13}[\widehat{\mathcal{F}}] = A_{23}[\widehat{\mathcal{F}}] = \begin{bmatrix} -0.8135 & 2.7988 \\ -1.3697 & 4.0149 \end{bmatrix}, \qquad b_{13}[\widehat{\mathcal{F}}] = \begin{bmatrix} 1.5980 \\ 2.1879 \end{bmatrix}, \qquad b_{23}[\widehat{\mathcal{F}}] = \begin{bmatrix} -2.0047 \\ -3.2027 \end{bmatrix},$$

$$A_{14}[\widehat{\mathcal{F}}] = A_{24}[\widehat{\mathcal{F}}] = \begin{bmatrix} 1.0277 & -0.8020 \\ -1.3697 & 4.0149 \end{bmatrix}, \quad b_{14}[\widehat{\mathcal{F}}] = \begin{bmatrix} 1.5920 \\ -3.2013 \end{bmatrix}, \quad b_{24}[\widehat{\mathcal{F}}] = \begin{bmatrix} -2.0059 \\ 7.5915 \end{bmatrix},$$

$$A_{15}[\widehat{\mathcal{F}}] = A_{25}[\widehat{\mathcal{F}}] = \begin{bmatrix} 1.0277 & -0.8020 \\ -0.8135 & 2.7988 \end{bmatrix}, \quad b_{15}[\widehat{\mathcal{F}}] = \begin{bmatrix} 2.1993 \\ -3.2020 \end{bmatrix}, \quad b_{25}[\widehat{\mathcal{F}}] = \begin{bmatrix} -3.1917 \\ 7.6153 \end{bmatrix}.$$

*The linear least square problems (5.28) are solved to obtain* $\widehat{G}$ *and* $N_3(\widehat{G}), N_4(\widehat{G}), N_5(\widehat{G})$, *which are*

$$N_3(\widehat{G}) = \begin{bmatrix} 0.5156 & 0.7208 \\ 1.6132 & -0.2474 \end{bmatrix}, \; N_4(\widehat{G}) = \begin{bmatrix} 1.2631 & -0.3665 \\ -0.6489 & 1.6695 \end{bmatrix},$$

$$N_5(\widehat{G}) = \begin{bmatrix} 1.6131 & -0.6752 \\ -1.2704 & 2.3517 \end{bmatrix}.$$

*For* $\xi = (3, 4, 5)$, *the eigendecomposition of the matrix* $\widehat{N}(\xi)$ *in (5.29) is*

$$\widehat{N}(\xi) = \begin{bmatrix} -0.7078 & 0.4470 \\ -0.7064 & -0.8945 \end{bmatrix} \begin{bmatrix} 12.0343 & 0 \\ 0 & 20.0786 \end{bmatrix} \begin{bmatrix} -0.7524 & 0.4499 \\ -0.6588 & -0.8931 \end{bmatrix}^{-1}.$$

*It has eigenvectors* $\hat{v}_1 = (-0.7078, -0.7064), \hat{v}_2 = (0.4470, -0.8945)$. *The vectors* $\hat{w}_1, \hat{w}_2$ *obtained as in (5.30) are*

$$\hat{w}_1 = (1.2021, 0.9918, 0.9899), \; \hat{w}_2 = (-1.0389, 2.0145, 3.0016).$$

*By solving (5.31) and (5.32), we got the scalars*

$$\hat{\gamma}_1 = -1.1990, \; \hat{\gamma}_2 = -2.1458, \qquad \hat{\lambda}_1 = 0.4521, \; \hat{\lambda}_2 = 0.6232.$$

*Finally, we got the decomposition* $\hat{\lambda}_1 \hat{u}_1^{\otimes 3} + \hat{\lambda}_2 \hat{u}_2^{\otimes 3}$ *with*

$$\hat{u}_1 = (1, \hat{\gamma}_1 \hat{v}_1, \hat{w}_1) = (1, 0.8477, 0.8479, 1.2021, 0.9918, 0.9899),$$
$$\hat{u}_2 = (1, \hat{\gamma}_2 \hat{v}_2, \hat{w}_2) = (1, -0.9776, 1.9102, -1.0389, 2.0145, 3.0016).$$

*They are pretty close to the decomposition of* $\mathcal{F}$.

## 5.4 Learning diagonal GMMs

We use the incomplete tensor decomposition or approximation method to recover parameters for Gaussian mixture models. The Algorithms 5.4 and 5.6 can be applied to do that.

Let $y$ be the random variable of dimension $d$ for a Gaussian mixture model, with $r$ components of Gaussian distribution parameters $(\omega_i, \mu_i, \Sigma_i)$, $i = 1, \ldots, r$. We consider the case that $r \leq \frac{d}{2} - 1$. Let $y_1, \ldots, y_N$ be samples drawn from the Gaussian mixture model. The sample average

$$\widehat{M_1} := \frac{1}{N}(y_1 + \cdots + y_N)$$

is an estimation for the mean $M_1 := \mathbb{E}[y] = \omega_1\mu_1 + \cdots + \omega_r\mu_r$. The symmetric tensor

$$\widehat{M_3} := \frac{1}{N}(y_1^{\otimes 3} + \cdots + y_N^{\otimes 3})$$

is an estimation for the third order moment tensor $M_3 := \mathbb{E}[y^{\otimes 3}]$. Recall that $\mathcal{F} = \sum_{i=1}^{r} \omega_i \mu_i^{\otimes 3}$. When all the covariance matrices $\Sigma_i$ are diagonal, we have shown in (5.3) that

$$M_3 = \mathcal{F} + \sum_{j=1}^{d}(a_j \otimes e_j \otimes e_j + e_j \otimes a_j \otimes e_j + e_j \otimes e_j \otimes a_j).$$

If the labels $i_1, i_2, i_3$ are distinct from each other, $(M_3)_{i_1 i_2 i_3} = (\mathcal{F})_{i_1 i_2 i_3}$. Recall the label set $\mathcal{I}$ in (5.5). It holds that

$$(M_3)_\Omega = (\mathcal{F})_\Omega.$$

Note that $(\widehat{M_3})_\Omega$ is only an approximation for $(M_3)_\Omega$ and $(\mathcal{F})_\Omega$, due to sampling errors. If the rank $r \leq \frac{d}{2} - 1$, we can apply Algorithm 5.6 with the input $(\widehat{M_3})_\Omega$, to compute a rank-$r$ tensor approximation for $\mathcal{F}$. Suppose the tensor approximation produced by Algorithm 5.6 is

$$\mathcal{F} \approx (p_1^*)^{\otimes 3} + \cdots + (p_r^*)^{\otimes 3}.$$

The computed $p_1^*, \ldots, p_r^*$ may not be real vectors, even if $\mathcal{F}$ is real. When the error $\epsilon := \|(\mathcal{F} - \widehat{M_3})_\Omega\|$ is small, by Theorem 5.7, we know

$$\|\tau_i^* p_i^* - \sqrt[3]{\omega_i}\mu_i\| \;=\; O(\epsilon)$$

where $(\tau_i^*)^3 = 1$. In computation, we can choose $\tau_i^*$ such that $(\tau_i^*)^3 = 1$ and the imaginary part vector $\mathrm{Im}(\tau_i^* p_i^*)$ has the smallest norm. It can be done by checking the imaginary part

of $\tau_i^* p_i^*$ one by one for

$$\tau_i^* = 1,\ -\frac{1}{2} + \frac{\sqrt{-3}}{2},\ -\frac{1}{2} - \frac{\sqrt{-3}}{2}.$$

Then we get the real vector

$$\hat{q}_i \ := \ \mathrm{Re}(\tau_i^* p_i^*).$$

It is expected that $\hat{q}_i \approx \sqrt[3]{\omega_i}\mu_i$. Since

$$M_1 = \omega_1\mu_1 + \cdots + \omega_r\mu_r \approx \omega_1^{2/3}\hat{q}_1 + \cdots + \omega_r^{2/3}\hat{q}_r,$$

the scalars $\omega_1^{2/3}, \ldots, \omega_r^{2/3}$ can be obtained by solving the linear least squares

$$\min_{(\beta_1,\ldots,\beta_r)\in\mathbb{R}_+^r} \left\| \widehat{M}_1 - \sum_{i=1}^r \beta_i \hat{q}_i \right\|^2. \tag{5.35}$$

Let $(\beta_1^*, \ldots, \beta_r^*)$ be an optimizer for the above, then $\hat{\omega}_i := (\beta_i^*)^{3/2}$ is a good approximation for $\omega_i$ and the vector

$$\hat{\mu}_i \ := \ \hat{q}_i / \sqrt[3]{\hat{\omega}_i}$$

is a good approximation for $\mu_i$. We may use

$$\hat{\mu}_i, \quad \big(\sum_{j=1}^r \hat{\omega}_j\big)^{-1}\hat{\omega}_i, \quad i = 1, \ldots, r$$

as starting points to solve the nonlinear optimization

$$\begin{cases} \min\limits_{(\omega_1,\ldots,\omega_r,\mu_1,\ldots,\mu_r)} & \|\sum_{i=1}^r \omega_i\mu_i - \widehat{M}_1\|^2 + \|\sum_{i=1}^r \omega_i(\mu_i^{\otimes 3})_{\mathcal{I}} - (\widehat{M}_3)_{\mathcal{I}}\|^2 \\ \text{subject to} & \omega_1 + \cdots + \omega_r = 1,\ \omega_1, \ldots, \omega_r \geq 0, \end{cases} \tag{5.36}$$

for getting improved approximations. Suppose an optimizer of the above is

$$(\omega_1^*, \ldots, \omega_r^*, \mu_1^*, \ldots, \mu_r^*).$$

Now we discuss how to estimate the diagonal covariance matrices $\Sigma_i$. Let

$$\mathcal{A} := M_3 - \mathcal{F}, \quad \widehat{\mathcal{A}} := \widehat{M}_3 - (\hat{q}_1)^{\otimes 3} - \cdots - (\hat{q}_r)^{\otimes 3}. \tag{5.37}$$

By (5.3), we know that

$$\mathcal{A} = \sum_{j=1}^d (a_j \otimes e_j \otimes e_j + e_j \otimes a_j \otimes e_j + e_j \otimes e_j \otimes a_j), \tag{5.38}$$

where $a_j = \sum_{i=1}^{r} \omega_i \sigma_{ij}^2 \mu_i$ for $j = 1, \cdots, d$. The equation (5.38) implies that

$$(a_j)_j = \frac{1}{3}\mathcal{A}_{jjj}, \quad (a_j)_i = \mathcal{A}_{jij}, \tag{5.39}$$

for $i, j = 1, \cdots, d$ and $i \neq j$. So we choose vectors $\hat{a}_j \in \mathbb{R}^d$ such that

$$(\hat{a}_j)_j = \frac{1}{3}\widehat{\mathcal{A}}_{jjj}, \quad (\hat{a}_j)_i = \widehat{\mathcal{A}}_{jij} \quad \text{for} \quad i \neq j. \tag{5.40}$$

Since $\hat{a}_j \approx \sum_{i=1}^{r} \omega_i \sigma_{ij}^2 \mu_i$, the covariance matrices $\Sigma_i = \operatorname{diag}(\sigma_{i1}^2, \ldots, \sigma_{id}^2)$ can be estimated by solving the nonnegative linear least squares ($j = 1, \ldots, d$)

$$\begin{cases} \min\limits_{(\beta_{1j}, \ldots, \beta_{rj})} & \left\| \hat{a}_j - \sum\limits_{i=1}^{r} \omega_i^* \mu_i^* \beta_{ij} \right\|^2 \\ \text{subject to} & \beta_{1j} \geq 0, \ldots, \beta_{rj} \geq 0. \end{cases} \tag{5.41}$$

For each $j$, let $(\beta_{1j}^*, \ldots, \beta_{rj}^*)$ be the optimizer for the above. When $(\widehat{M_3})_\Omega$ is close to $(M_3)_\Omega$, it is expected that $\beta_{ij}^*$ is close to $(\sigma_{ij})^2$. Therefore, we can estimate the covariance matrices $\Sigma_i$ as follows

$$\Sigma_i^* := \operatorname{diag}(\beta_{i1}^*, \ldots, \beta_{id}^*), \quad (\sigma_{ij}^*)^2 := \beta_{ij}^*. \tag{5.42}$$

The following is the algorithm for learning Gaussian mixture models.

**Algorithm 5.9.** *(Learning diagonal Gaussian mixture models.)*

*Input: Samples $\{y_1, \ldots, y_N\} \subseteq \mathbb{R}^d$ drawn from a Gaussian mixture model and the number $r$ of component Gaussian distributions.*

*Step 1. Compute the sample averages $\widehat{M_1} := \frac{1}{N}\sum_{i=1}^{N} y_i$ and $\widehat{M_3} := \dfrac{1}{N}\sum\limits_{i=1}^{N} y_i^{\otimes 3}$.*

*Step 2. Apply Algorithm 5.6 to the subtensor $(\widehat{\mathcal{F}})_\Omega := (\widehat{M_3})_\Omega$. Let $(p_1^*)^{\otimes 3} + \cdots + (p_r^*)^{\otimes 3}$ be the obtained rank-$r$ tensor approximation for $\widehat{\mathcal{F}}$. For each $i = 1, \ldots, r$, let $\hat{q}_i := Re(\tau_i p_i^*)$ where $\tau_i$ is the cube root of 1 that minimizes the imaginary part vector norm $\|Im(\tau_i p_i^*)\|$.*

*Step 3. Solve (5.35) to get $\hat{\omega}_1, \ldots, \hat{\omega}_r$ and $\hat{\mu}_i = q_i / \sqrt[3]{\hat{\omega}_i}, i = 1, \ldots, r$.*

*Step 4. Use the above $\hat{\omega}_i$, $\hat{q}_i$ as initial points to solve the nonlinear optimization (5.36) for the optimal $\omega_i^*, \mu_i^*, i = 1, \ldots, r$.*

*Step 5. Get vectors $\hat{a}_1, \ldots, \hat{a}_d$ as in (5.40). Solve the optimization (5.41) to get optimizers $\beta_{ij}^*$ and then choose $\Sigma_i^*$ as in (5.42).*

*Output: Component Gaussian distribution parameters* $(\mu_i^*, \Sigma_i^*, \omega_i^*), i = 1, \ldots, r$.

The sample averages $\widehat{M}_1, \widehat{M}_3$ can typically be used as good estimates for the true moments $M_1, M_3$. When the value of $r$ is not known, it can be determined as in Remark 5.2. The performance of Algorithm 5.9 is analyzed as follows.

**Theorem 5.10** ( [59]). *Consider the d-dimensional diagonal Gaussian mixture model with parameters* $\{(\omega_i, \mu_i, \Sigma_i) : i \in [r]\}$ *and* $r \leq \frac{d}{2} - 1$. *Let* $\{(\omega_i^*, \mu_i^*, \Sigma_i^*) : i \in [r]\}$ *be produced by Algorithm 5.9. If the distance* $\epsilon := \max(\|M_3 - \widehat{M}_3\|, \|M_1 - \widehat{M}_1\|)$ *is small enough and the tensor* $\mathcal{F} = \sum_{i=1}^r \omega_i \mu_i^{\otimes 3}$ *satisfies conditions of Theorem 5.7, then*

$$\|\mu_i - \mu_i^*\| = O(\epsilon), \|\omega_i - \omega_i^*\| = O(\epsilon), \|\Sigma_i - \Sigma_i^*\| = O(\epsilon),$$

*where the above constants inside* $O(\cdot)$ *only depend on parameters* $\{(\omega_i, \mu_i, \Sigma_i) : i \in [r]\}$ *and the choice of* $\xi$ *in Algorithm 5.9.*

*Proof.* For the vectors $p_i := \sqrt[3]{\omega_i}\mu_i$, we have $\mathcal{F} = \sum_{i=1}^r p_i^{\otimes 3}$. Since

$$\|(\mathcal{F} - \widehat{\mathcal{F}})_{\mathcal{I}}\| = \|(M_3 - \widehat{M}_3)_{\mathcal{I}}\| \leq \epsilon$$

and $\mathcal{F}$ satisfies conditions of Theorem 5.7, we know $\|\tau_i^* p_i^* - p_i\| = O(\epsilon)$ for some $(\tau_i^*)^3 = 1$, by Theorem 5.7. The constants inside $O(\epsilon)$ depend on parameters of the Gaussian model and $\xi$. Then, we have $\|\mathrm{Im}(\tau_i^* p_i^*)\| = O(\epsilon)$ since the vectors $p_i$ are real. When $\epsilon$ is small enough, such $\tau_i^*$ is the $\tau$ in Step 2 of Algorithm 5.9 that minimizes $\|\mathrm{Im}(\tau_i p_i^*)\|$, so we have

$$\|\hat{q}_i - p_i\| \leq \|\tau_i p_i^* - p_i\| = O(\epsilon)$$

where $\hat{q}_i = \mathrm{Re}(\tau_i p_i^*)$ is from the Step 2. The vectors $\hat{q}_1, \ldots, \hat{q}_r$ are linearly independent when $\epsilon$ is small. Thus, the problem (5.35) has a unique solution and the weights $\hat{\omega}_i$ can be found by solving (5.35). Since $\|M_1 - \widehat{M}_1\| \leq \epsilon$ and $\|\hat{q}_i - p_i\| = O(\epsilon)$, we have $\|\omega_i - \hat{\omega}_i\| = O(\epsilon)$ (see [40, Theorem 3.4]). The mean vectors $\hat{\mu}_i$ are obtained by $\hat{\mu}_i = \hat{q}_i / \sqrt[3]{\hat{\omega}_i}$, so the approximation error is

$$\|\mu_i - \hat{\mu}_i\| = \|p_i / \sqrt[3]{\omega_i} - \hat{q}_i / \sqrt[3]{\hat{\omega}_i}\| = O(\epsilon).$$

The constants inside the above $O(\epsilon)$ depend on parameters of the Gaussian mixture model and $\xi$.

The problem (5.36) is solved to obtain $\omega_i^*$ and $\mu_i^*$, so

$$\left\| \widehat{M}_1 - \sum_{i=3}^r \omega_i^* \mu_i^* \right\| + \left\| \widehat{\mathcal{F}} - \sum_{i=1}^r \omega_i^* (\mu_i^*)^{\otimes 3} \right\| = O(\epsilon).$$

Let $\mathcal{F}^* := \sum_{i=1}^r \omega_i^* (\mu_i^*)^{\otimes 3} = \sum_{i=1}^r (\sqrt[3]{\omega_i^*}\mu_i^*)^{\otimes 3}$, then

$$\|\mathcal{F} - \mathcal{F}^*\| \leq \|\mathcal{F} - \hat{\mathcal{F}}\| + \|\hat{\mathcal{F}} - \mathcal{F}^*\| = O(\epsilon).$$

Theorem 5.7 implies $\|p_i - \sqrt[3]{\omega_i^*}\mu_i^*\| = O(\epsilon)$. In addition, we have

$$\left\|\widehat{M}_1 - \sum_{i=1}^r \omega_i^* \mu_i^*\right\| = \left\|\widehat{M}_1 - \sum_{i=1}^r (\omega_i^*)^{2/3} \sqrt[3]{\omega_i^*}\mu_i^*\right\| = O(\epsilon).$$

The first order moment is $M_1 = \sum_{i=1}^r (\omega_i)^{2/3} p_i$. Since $\|M_1 - \hat{M}_1\| = O(\epsilon)$ and $\|p_i - \sqrt[3]{\omega_i^*}\mu_i^*\| = O(\epsilon)$, it holds that $\|\omega_i^{2/3} - (\omega_i^*)^{2/3}\| = O(\epsilon)$ by [40, Theorem 3.4]. This implies that $\|\omega_i - \omega_i^*\| = O(\epsilon)$, so

$$\|\mu_i - \mu_i^*\| = \|p_i/\sqrt[3]{\omega_i} - (\sqrt[3]{\omega_i^*}\mu_i^*)/\sqrt[3]{\omega_i^*}\| = O(\epsilon).$$

The constants inside the above $O(\cdot)$ only depend on parameters $\{(\omega_i, \mu_i, \Sigma_i) : i \in [r]\}$ and $\xi$.

The covariance matrices $\Sigma_i$ are recovered by solving the linear least squares (5.41). In the least square problems, it holds that $\|\omega_i \mu_i - \omega_i^* \mu_i^*\| = O(\epsilon)$ and

$$\|\mathcal{A} - \widehat{\mathcal{A}}\| \leq \|M_3 - \widehat{M}_3\| + \|\mathcal{F} - \sum_{i=1}^r \hat{q}_i^{\otimes 3}\| = O(\epsilon),$$

where tensors $\mathcal{A}, \widehat{\mathcal{A}}$ are defined in (5.37). When the error $\epsilon$ is small, the vectors $\omega_i^*\mu_1^*, \ldots, \omega_i^*\mu_r^*$ are linearly independent and hence (5.41) has a unique solution for each $j$. According to [40, Theorem 3.4], we have

$$\|(\sigma_{ij})^2 - (\sigma_{ij}^*)^2\| = O(\epsilon).$$

It implies that $\|\Sigma_i - \Sigma_i^*\| = O(\epsilon)$, where the constants inside $O(\cdot)$ only depend on parameters $\{(\omega_i, \mu_i, \Sigma_i) : i \in [r]\}$ and $\xi$. □

## 5.5 Numerical examples

First, we show the performance of Algorithm 5.6 for computing incomplete symmetric tensor approximations. For a range of dimension $d$ and rank $r$, we get the tensor $\mathcal{F} = (p_1)^{\otimes 3} + \cdots + (p_r)^{\otimes 3}$, where each $p_i$ is randomly generated according to the Gaussian distribution in `MATLAB`. Then, we apply the perturbation $(\widehat{\mathcal{F}})_\Omega = (\mathcal{F})_\Omega + \mathcal{E}_\Omega$, where $\mathcal{E}$ is a randomly generated tensor, also according to the Gaussian distribution in `MATLAB`, with the norm $\|\mathcal{E}_\omega\|_\Omega = \epsilon$. After that, Algorithm 5.6 is applied to the subtensor $(\widehat{\mathcal{F}})_\Omega$ to find the rank-$r$

tensor approximation. The approximation quality is measured by the absolute error and the relative error

$$\text{abs-error} := \|(\mathcal{F}^* - \mathcal{F})_\Omega\|, \quad \text{rel-error} := \frac{\|(\mathcal{F}^* - \widehat{\mathcal{F}})_\Omega\|}{\|(\mathcal{F} - \widehat{\mathcal{F}})_\Omega\|},$$

where $\mathcal{F}^*$ is the output of Algorithm 5.6. For each case of $(d, r, \epsilon)$, we generate 100 random instances. The min, average, and max relative errors for each dimension $d$ and rank $r$ are reported in the Table 5.1. The results show that Algorithm 5.6 performs very well for computing tensor approximations.

Table 5.1: The performance of Algorithm 5.6

| | | | rel-error | | | abs-error | | | |
|---|---|---|---|---|---|---|---|---|---|
| $d$ | $r$ | $\epsilon$ | min | average | max | min | average | max | time |
| 20 | 3 | 0.1 | 0.9610 | 0.9731 | 0.9835 | 0.0141 | 0.0268 | 0.0556 | 0.2687 |
| | 5 | 0.01 | 0.9634 | 0.9700 | 0.9742 | 0.0019 | 0.0032 | 0.0068 | 0.2392 |
| | 7 | 0.001 | 0.9148 | 0.9373 | 0.9525 | $2.3 \cdot 10^{-4}$ | $3.8 \cdot 10^{-4}$ | $6.6 \cdot 10^{-4}$ | 0.2638 |
| 30 | 4 | 0.1 | 0.9816 | 0.9854 | 0.9890 | 0.0094 | 0.0174 | 0.0533 | 0.4386 |
| | 8 | 0.01 | 0.9634 | 0.9700 | 0.9742 | 0.0015 | 0.0024 | 0.0060 | 0.7957 |
| | 11 | 0.001 | 0.9501 | 0.9587 | 0.9667 | $1.8 \cdot 10^{-4}$ | $3.0 \cdot 10^{-4}$ | $5.7 \cdot 10^{-4}$ | 0.8954 |
| 40 | 6 | 0.1 | 0.9853 | 0.9877 | 0.9904 | 0.0099 | 0.0146 | 0.0359 | 1.7779 |
| | 10 | 0.01 | 0.9761 | 0.9795 | 0.9820 | 0.0013 | 0.0020 | 0.0045 | 2.6454 |
| | 15 | 0.001 | 0.9653 | 0.9690 | 0.9734 | $1.7 \cdot 10^{-4}$ | $2.6 \cdot 10^{-4}$ | $4.8 \cdot 10^{-4}$ | 3.6785 |
| 50 | 7 | 0.1 | 0.9887 | 0.9911 | 0.9925 | 0.0081 | 0.0128 | 0.0294 | 4.9774 |
| | 13 | 0.01 | 0.9812 | 0.9831 | 0.9854 | 0.0011 | 0.0018 | 0.0045 | 8.7655 |
| | 18 | 0.001 | 0.9739 | 0.9767 | 0.9792 | $1.5 \cdot 10^{-4}$ | $2.2 \cdot 10^{-4}$ | $4.1 \cdot 10^{-4}$ | 11.6248 |

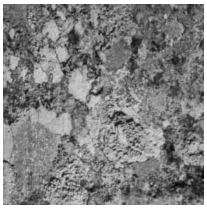
Second, we explore the performance of Algorithm 5.9 for learning diagonal Gaussian mixture models. We compare it with the classical EM algorithm, for which the MATLAB function `fitgmdist` is used (`MaxIter` is set to be 100 and `RegularizationValue` is set to be 0.0001). The dimensions $d = 20, 30, 40, 50, 60$ are tested. Three values of $r$ are tested for each case of $d$. We generate 100 random instances of $\{(\omega_i, \mu_i, \Sigma_i) : i = 1, \cdots, r\}$ for $d \in \{20, 30, 40\}$, and 20 random instances for $d \in \{50, 60\}$, because of the relatively more computational time for the latter case. For each instance, 10000 samples are generated. To generate the weights $\omega_1, \ldots, \omega_r$, we first use the MATLAB function `randi` to generate a

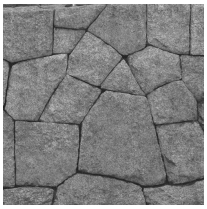
random 10000−dimensional integer vector of entries from $[r]$, then the occurring frequency of $i$ in $[r]$ is used as the weight $\omega_i$. For each diagonal covariance matrix $\Sigma_i$, its diagonal vector is set to be the square of a random vector generated by the MATLAB function `randn`. Each sample is generated from one of $r$ component Gaussian distributions, so they are naturally separated into $r$ groups. Algorithm 5.9 and the EM algorithm are applied to fit the Gaussian mixture model to the 10000 samples for each instance. For each sample, we calculate the likelihood of the sample to each component Gaussian distribution in the estimated Gaussian mixture model. A sample is classified to the $i$th group if its likelihood for the $i$th component is maximum. The classification accuracy is the rate that samples are classified to the correct group. In Table 5.2, for each pair $(d, r)$, we report the accuracy of Algorithm 5.9 in the first row and the accuracy of the EM algorithm in the second row. As one can see, Algorithm 5.9 performs better than EM algorithm, and its accuracy isn't affected when the dimensions and ranks increase. Indeed, as the difference between the dimension $d$ and the rank $r$ increases, Algorithm 5.9 becomes more and more accurate. This is opposite to the EM algorithm. The reason is that the difference between the number of rows and the number of columns of $A_{ij}[\mathcal{F}]$ in (5.15) increases as $d - r$ becomes bigger, which makes Algorithm 5.9 more robust.

Last, we apply Algorithm 5.9 to do texture classifications. We select 8 textured images of $512 \times 512$ pixels from the VisTex database, which are shown in Figure 5.1. We use the MATLAB function `rgb2gray` to convert them into grayscale version since we only need their structure and texture information. Each image is divided into subimages of $32 \times 32$ pixels. We perform the discrete cosine transformation(DCT) on each block of size $16 \times 16$ pixels with overlap of 8 pixels. Each component of 'Wavelet-like' DCT feature is the sum of the absolute value of the DCT coefficients in the corresponding sub-block. So the dimension $d$ of the feature vector extracted from each subimage is 13. We use blocks extracted from the first 160 subimages for training and those from the rest 96 subimages for testing. We refer to [123] for more details. For each image, we apply Algorithm 5.9 and the EM algorithm to fit a Gaussian mixture model to the image. We choose the number of components $r$ according to Remark 5.2. To classify the test data, we follow the Bayes decision rule that assigns each block to the texture which maximizes the posteriori probability, where we assume a uniform prior over all classes [46]. The classification accuracy is the rate that a subimage is correctly classified, which is shown in Table 5.3. Algorithm 5.9 outperforms the classical EM algorithm for the accuracy rates for six of the images.

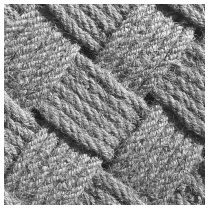
Table 5.2: Comparison between Algorithm 5.9 and EM for simulations

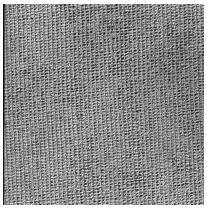
| | | accuracy | | time | |
|---|---|---|---|---|---|
| $d$ | $r$ | Algorithm 5.9 | EM | Algorithm 5.9 | EM |
| 20 | 3 | 0.9861 | 0.9763 | 0.8745 | 0.1649 |
| | 5 | 0.9740 | 0.9400 | 2.3476 | 0.3852 |
| | 7 | 0.9659 | 0.9252 | 3.4352 | 0.6777 |
| 30 | 4 | 0.9965 | 0.9684 | 4.5266 | 0.2959 |
| | 8 | 0.9923 | 0.9277 | 8.5494 | 0.8525 |
| | 11 | 0.9895 | 0.9219 | 17.2091 | 1.4106 |
| 40 | 6 | 0.9990 | 0.9117 | 18.9160 | 0.6273 |
| | 10 | 0.9981 | 0.8931 | 28.4161 | 1.2617 |
| | 15 | 0.9971 | 0.9111 | 69.8013 | 2.0627 |
| 50 | 7 | 0.9997 | 0.8997 | 40.6810 | 0.8314 |
| | 13 | 0.9995 | 0.9073 | 104.7927 | 1.7867 |
| | 18 | 0.9993 | 0.9038 | 163.2711 | 2.6862 |
| 60 | 8 | 0.9999 | 0.8874 | 93.9836 | 1.1266 |
| | 15 | 0.9998 | 0.8632 | 234.0331 | 2.6435 |
| | 22 | 0.9995 | 0.8929 | 497.9371 | 3.5527 |


Bark.0000

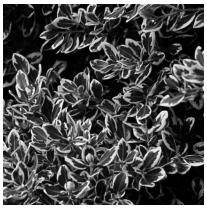

Bark.0009


Flowers.0001


Tile.0000


Paintings.11.0001


Grass.0001


Brick.0004

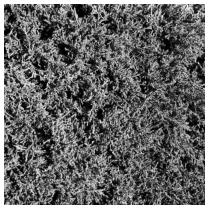

Fabric.0013

Figure 5.1: Textures from VisTex

Table 5.3: Classification results on 8 textures

| Accuracy | Algorithm 5.9 | EM |
|---|---|---|
| Bark.0000 | 0.5376 | 0.8413 |
| Bark.0009 | 0.5107 | 0.7150 |
| Flowers.0001 | 0.8137 | 0.6315 |
| Tile.0000 | 0.8219 | 0.7239 |
| Paintings.11.0001 | 0.8047 | 0.7350 |
| Grass.0001 | 0.9841 | 0.9068 |
| Brick.0004 | 0.9406 | 0.8854 |
| Fabric.0013 | 0.9220 | 0.9048 |

**Acknowledgement.** The Chapter 5, in full, has been accepted for publication in *Vietnam Journal of Mathematics* 2021 [59]. The dissertation author coauthored this paper with Guo, Bingni and Nie, Jiawang.

# Bibliography


[1] L. Ardila, M. Heyl and A. Eckardt, *Measuring the single-particle density matrix for fermions and hard-core bosons in an optical lattice*, Physical review letters, 121 (2018), no. 26, pp. 260–401.

[2] A. Ben-Tal and A. Nemirovski, *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*, MPS-SIAM Series on Optimization, SIAM, Philadelphia, 2001.

[3] A. Berman, M. Dür, and N. Shaked-Monderer. *Open problems in the theory of completely positive and copositive matrices. Electronic Journal of Linear Algebra* 29 (2015), 46-58.

[4] A. Bernardi, J. Brachat, P. Comon, and B. Mourrain, *General tensor decomposition, moment matrices and applications*, J. Symbolic Comput., 52 (2013), pp. 51–71.

[5] I. Bomze. *Linear-time detection of copositivity for tridiagonal matrices and extension to block-tridiagonality.* SIAM J. Matrix Anal. Appl. 21 (2000), 840-848.

[6] I. Bomze. *Copositive optimization - recent developments and applications.* European Journal of Operational Research 216 (2012), 509–520.

[7] I. Bomze and G. Eichfelder. *Copositivity detection by difference-of-convex decomposition and $\omega$-subdivision.* Math. Program., pages 1–36, 2013.

[8] D. Bertsekas, *Convex Optimization Theory*, Athena Scientific, 2009.

[9] D. Bertsekas, *Nonlinear programming, second edition*, Athena Scientific, 1995.

[10] D. Bertsekas, A. Nedić and A. Ozdaglar, *Convex Analysis and Optimization*, Athena Scientific, Belmont, 2003.

[11] G. Blekherman, P. Parrilo and R. Thomas (eds.), *Semidefinite optimization and convex algebraic geometry*, MOS-SIAM series on Optimization, SIAM, Philadelphia, PA, 2013.

[12] K. Blum, *Density matrix theory and applications*, Springer Science & Business Media, 2012.

[13] J. Bochnak, M. Coste and M-F. Roy, *Real Algebraic Geometry*, Springer, 1998.

[14] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2004.

[15] J. Brachat, P. Comon, B. Mourrain, and E. Tsigaridas, *Symmetric tensor decomposition*, Linear Algebra Appl., 433 (2010), no. 11–12, pp. 1851–1872.

[16] P. Breiding and N. Vannieuwenhoven, *A Riemannian trust region method for the canonical tensor rank approximation problem*, SIAM J. Optim., 28 (2018), no. 3, pp. 2435–2465.

[17] P. Breiding and N. Vannieuwenhoven. *The condition number of join decompositions.* SIAM Journal on Matrix Analysis and Applications, 39(1):287–309, 2018.

[18] P. Breiding and N. Vannieuwenhoven. *The condition number of Riemannian approximation problems.* SIAM Journal on Optimization, 31(1):1049–1077, 2021.

[19] S. Bundfuss and M. Dür. *Algorithmic copositivity detection by simplicial partition.* Linear Algebra and its Applications, 428(7):1511–1523, 2008.

[20] S. Burer. *On the copositive representation of binary and continuous nonconvex quadratic programs.* Math. Program., 120(2):479–495, 2009.

[21] L. Calderaro, G. Foletto, D. Dequal, P. Villoresi and G. Vallone, *Direct reconstruction of the quantum density matrix by strong measurements*, Physical review letters, 121 (2018), no. 23, pp. 230–501.

[22] F. Chatelin, *Eigenvalues of matrices: revised edition*, SIAM, 2012.

[23] M. Che, L. Qi, and Y. Wei. *Positive-definite tensors to nonlinear complementarity problems.* Journal of Optimization Theory and Applications, 168(2):475–487, 2016.

[24] H. Chen, Z. Huang, and L. Qi. *Copositive tensor detection and its applications in physics and hypergraphs.* arXiv preprint arXiv:1609.07919, 2016.

[25] H. Chen, Z. Huang and L. Qi. *Copositivity detection of tensors: theory and algorithm.* J. Optimization Theory and Applications 174(3): 746-761 (2017).

[26] Y. Chen, G. Lan, and Y. Ouyang. *Optimal primal-dual methods for a class of saddle point problems.* SIAM J. Optim. 24(2014), no. 4, 1779–1814.

[27] L. Chiantini, G. Ottaviani, and N. Vannieuwenhoven, *Effective criteria for specific identifiability of tensors and forms*, SIAM J. Matrix Anal. Appl., 38 (2017), pp. 656–681.

[28] P. Comon, G. Golub, L.-H. Lim, and B. Mourrain, *Symmetric tensors and symmetric tensor rank*, SIAM J. Matrix Anal. Appl., 30 (2008), no. 3, pp. 1254–1279.

[29] P. Comon, L.-H. Lim, Y. Qi and K. Ye, *Topology of tensor ranks*, Advances in Mathematics, vol. 367, pp. 107-128, 2020.

[30] B. Cox, A. Juditsky, and A. Nemirovski, *Decomposition techniques for bilinear saddle point problems and variational inequalities with affine monotone operators*, J. Optim. Theory Appl. 172(2017), no. 2, 402–435.

[31] C. Cui, Y. Dai and J. Nie. *All real eigenvalues of symmetric tensors.* SIAM J. Matrix Anal. Appl., 35 (2014), pp. 1582–1601.

[32] R. Curto and L. Fialkow. *Truncated K-moment problems in several variables.* J. Operator Theory, 54(2005), pp. 189-226.

[33] G. Dahl, J. M. Leinaas, J. Myrheim, and E. Ovrum, *A tensor product matrix approximation problem in quantum physics*, Linear Algebra and its Applications, 420 (2007), pp. 711–725.

[34] Y. Dauphin, R. Pascanu, C. Gülçehre, K. Cho, S. Ganguli and Y. Bengio, *Identifying and attacking the saddle point problem in high-dimensional non-convex optimization*, Advances in Neural Information Processing Systems 27 (NIPS 2014), 2933–2941, Curran Associates, Inc., 2014.

[35] E. de Klerk and D. V. Pasechnik. *A linear programming reformulation of the standard quadratic optimization problem* J. Global Optim., 37 (2007), 75–84

[36] E. de Klerk and D. Pasechnik. *Approximation of the stability number of a graph via copositive programming.* SIAM J. Optim., 12(4):875–892, 2002.

[37] L. De Lathauwer, B. De Moor, and J. Vandewalle, *Computation of the canonical decomposition by means of a simultaneous generalized Schur decomposition*, SIAM J. Matrix Anal. Appl., 26 (2004), no. 2, pp. 295–327.

[38] L. De Lathauwer, *A link between the canonical decomposition in multilinear algebra and simultaneous matrix diagonalization*, SIAM J. Matrix Anal. Appl., 28 (2006), no. 3, pp. 642–666.

[39] V. De Silva and L.-H. Lim, *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM. J. Matrix Anal. Appl., 30 (2008), no. 3, pp. 1084–1127.

[40] J. Demmel, *Applied Numerical Linear Algebra*, SIAM, 1997.

[41] J. Demmel, J.Nie and V. Powers, *Representations of positive polynomials on non-compact semialgebraic sets via KKT ideals*, J. Pure Appl. Algebra 209(2007), no. 1, pp. 189–200.

[42] H. Derksen, S. Friedland, L.-H. Lim and L. Wang, *Theoretical and computational aspects of entanglement*, `arXiv:1705.07160`, preprint, 2017.

[43] P. Diananda. *On non-negative forms in real variables some or all of which are non-negative.* Proceedings of the Cambridge Philosophical Society, 58 (1962), 17–25.

[44] P. Dickinson, M. Dür, L. Gijben and R. Hildebrand. *Scaling relationship between the copositive cone and Parrilo's first level approximation.* Optimization Letters, 7 (2013), 1669-1679.

[45] P. Dickinson and L. Gijben. *On the computational complexity of membership problems for the completely positive cone and its dual.* Comput. Optim. Appl., 57 (2014), 403–415.

[46] M. Dixit, N. Rasiwasia, and N. Vasconcelos, *Adapted Gaussian models for image classification*. CVPR 2011, pages 937–943, 2011.

[47] C. Dobre and J. Vera. *Exploiting symmetry in copositive programs via semidefinite hierarchies*. Math. Program., 151(2):659-680, 2015.

[48] I. Domanov, and L. De Lathauwer, *Generic uniqueness conditions for the canonical polyadic decomposition and INDSCAL*, SIAM J. Matrix Anal. Appl., 36 (2015), no. 4, pp. 1567–1589.

[49] M. Dressler, J. Nie, and Z. Yang, *Separability of Hermitian Tensors and PSD Decompositions*, Linear and Multilinear Linear Algebra, 2021.

[50] I. Dukanovic and . Rendl. *Copositive programming motivated bounds on the stability and the chromatic numbers*. Math. Program., 121(2):249–268, 2010.

[51] M. Dür. Copositive Programming - a survey. In: M. Diehl, F. Glineur, E. Jarlebring, W. Michiels (Eds.), *Recent Advances in Optimization and its Applications in Engineering*, Springer 2010, pp. 3-20.

[52] M. Dür and J. Hiriart-Urruty. *Testing copositivity with the help of difference-of-convex optimization* Math. Program., Vol. 140, No. 1, 31-43, 2013.

[53] F. Facchinei and J. Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

[54] L. Fialkow and J. Nie, *The truncated moment problem via homogenization and flat extensions*, J. Funct. Anal. 263 (6), 1682–1700, 2012.

[55] S. Friedland and L.-H. Lim, *Nuclear norm of higher-order tensors*, Mathematics of Computation, 87(311), 1255–1281, 2018.

[56] T. Fu, B. Jiang and Z. Li, *On decompositions and approximations of conjugate partial-symmetric complex tensors*, `arXiv:1802.09013`, preprint, 2018.

[57] F. Galuppi and M. Mella, *Identifiability of homogeneous polynomials and Cremona Transformations*, J. Reine Angew. Math., 757 (2019), pp. 279–308.

[58] F. Ge, Y. Ju, Z. Qi, and Y. Lin, *Parameter estimation of a gaussian mixture model for wind power forecast error by riemann l-bfgs optimization*. IEEE Access, 6:38892–38899, 2018.

[59] B. Guo, J. Nie, and Z. Yang, *Learning Diagonal Gaussian Mixture Models and Incomplete Tensor Decompositions*, Vietnam Journal of Mathematics, 2021.

[60] M. Hall and M. Newman. *Copositive and completely positive quadratic forms*. Proceedings of the Cambridge Philosophical Society, 59 (1963), 329–33.

[61] B. Halldórsson and R. Tütüncü, *An interior-point method for a class of saddle-point problems*, J. Optim. Theory Appl. 116(2003), no. 3, 559–590.

[62] J. Harris, *Algebraic Geometry, A First Course*, Springer Verlag, 1992.

[63] B. He and X. Yuan, *Convergence analysis of primal-dual algorithms for a saddle-point problem: from contraction perspective*, SIAM J. Imaging Sci. 5(2012), no. 1, 119–149.

[64] Y. He and R. Monteiro, *Accelerating block-decomposition first-order methods for solving composite saddle-point and two-player nash equilibrium problems*, SIAM J. Optim. 25(2015), no. 4, 2182–2211.

[65] J.W. Helon and J. Nie, A semidefinite approach for truncated K-moment problems, *§Found. Comput. Math.* 12 (6), 851–881, 2012.

[66] D. Henrion, J. Lasserre and J. Loefberg. *GloptiPoly 3: moments, optimization and semidefinite programming*. Optim. Methods Softw., 24 (2009), pp. 761–779.

[67] D. Henrion and J. B. Lasserre, *Detecting global optimality and extracting solutions in GloptiPoly*, Positive polynomials in control, Lect. Notes Control Inf. Sci., vol. 312, Springer, Berlin, 2005, pp. 293–310.

[68] R. Hildebrand. *The extreme rays of the* $5 \times 5$ *copositive cone*. Linear Algebra Appl., 437(7):1538–1547, 2012.

[69] J. Hiriart-Urruty and A. Seeger. *A variational approach to copositive matrices*. SIAM Review, 52 (2010), 593–629.

[70] A. Hoffman and F. Pereira. *On copositive matrices with* $-1, 0, 1$ *entries*. Journal of Combinatorial Theory, Series A, 14(3):302–309, 1973.

[71] K. Ikramov. *Linear-time algorithm for verifying the copositivity of an acyclic matrix*. Computational mathematics and mathematical physics, 42(12):1701–1703, 2002.

[72] D. Jacobson. *Extensions of linear-quadratic control, optimization and matrix theory*, volume 133. Academic press, 2000.

[73] B. Jiang, Z. Li, and S. Zhang, *Characterizing real-valued multivariate complex polynomials and their symmetric tensor representations*, SIAM J. Matrix Anal. Appl., 37 (2016), no. 1, pp. 381–408.

[74] K. Kannike. *Vacuum stability of a general scalar potential of a few fields*. The European Physical Journal C, 76(6):1–16, 2016.

[75] W. Kaplan. *A test for copositive matrices*. Linear Algebra Appl., 313 (2000), 203-206.

[76] T. Kolda and B. Bader, *Tensor decompositions and applications*, SIAM Rev., 51 (2009), no. 3, pp. 455–500.

[77] G.M. Korpelevič, *An extragradient method for finding saddle points and other problems*, Èkonom. i Mat. Metody 12(1976), no. 4, 747–756.

[78] J. Kruskal, *Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Lin. Alg. Appl., 18 (1977), no. 2, pp. 95–138.

[79] J. Landsberg, *Tensors: Geometry and Applications*, Grad. Stud. Math., Providence, 2012.

[80] R. Laraki and J. Lasserre, *Semidefinite programming for min-max problems and games*, Math. Program. 131(2012), 305–332.

[81] J. Lasserre. *Global optimization with polynomials and the problem of moments.* SIAM J. Optim., 11(3):796–817, 2001.

[82] J.B. Lasserre, *Introduction to Polynomial And Semi-Algebraic Optimization*, Cambridge University Press, Cambridge, 2015.

[83] M. Laurent, *Revisiting two theorems of Curto and Fialkow on moment matrices*, Proceedings of the AMS 133(2005), no. 10, 2965–2976.

[84] M. Laurent, *Sums of squares, moment matrices and optimization over polynomials*, Emerging Applications of Algebraic Geometry of IMA Volumes in Mathematics and its Applications, 149 (2009), pp. 157–270.

[85] D. -S. Lee, *Effective Gaussian mixture learning for video background subtraction*, IEEE transactions on pattern analysis and machine intelligence, 27(5):827–832, 2005.

[86] Z. Li, Y. Nakatsukasa, T. Soma and A. Uschmajew, *On orthogonal tensors and best rank-one approximation ratio*, SIAM J. Matrix Anal. Appl., 39 (2018), no. 1, pp. 400–425.

[87] Y. Li and G. Ni, *Separability discrimination and decomposition of m-partite quantum mixed states*, Phys. Rev. A **102** (2020), 012402.

[88] L.-H. Lim, *Tensors and hypermatrices, in: L. Hogben (Ed.)*, Handbook of linear algebra, 2nd Ed., CRC Press, Boca Raton, 2013.

[89] D. Maistroskii, *Gradient methods for finding saddle points*, Matekon 13 (1977), 3–22.

[90] M. Magdon-Ismail and J. T. Purnell, *Approximating the covariance matrix of gmms with low-rank perturbations*, International Conference on Intelligent Data Engineering and Automated Learning, pages 300–307, 2010.

[91] O. Mason and R. Shorten. *On linear copositive lyapunov functions and the stability of switched positive linear systems.* IEEE Transactions on Automatic Control, 52 (2007), 1346–1349.

[92] R. Monteiro and B. Svaiter, *Complexity of variants of Tseng's modified F-B splitting and Korpelevich's methods for hemivariational inequalities with applications to saddle-point and convex optimization problems*, SIAM J. Optim. 21(2011), no. 4, 1688–1720.

[93] T. Motzkin. *Copositive quadratic forms*. National Bureau of Standards Report, 1952:11–22, 1818.

[94] T. Motzkin and E. Straus. *Maxima for graphs and a new proof of a theorem of Turán*. Canadian J. Math., 17(1965): 533-540.

[95] C. Mu, B. Huang, J. Wright, and D. Goldfarb, *Square deal: lower bounds and improved relaxations for tensor recovery*, Proceeding of the International Conference on Machine Learning (PMLR), 32(2),73-81, 2014.

[96] K. Murty and S. Kabadi. *Some NP-complete problems in quadratic and nonlinear programming*. Math. Program., 39(2):117–129, 1987.

[97] M. Nakata. *A numerical evaluation of highly accurate multiple-precision arithmetic version of semidefinite programming solver:SDPA-GMP, -QD and -DD*. The proceedings of 2010 IEEE Multi-Conference on Systems and Control, 29-34, 2010. `http://sdpa.sourceforge.net/download.html#sdpa-gmp`

[98] A. Nedić and A. Ozdaglar, *Subgradient methods for saddle-point problems*, J. Optim. Theory Appl. 142(2009), no. 1, 205–228.

[99] A. Nemirovski, *Prox-method with rate of convergence $O(1/t)$ for variational inequalities with Lipschitz continuous monotone operators and smooth convex-concave saddle point problems*, SIAM J. Optim. 15 (2004), no. 1, 229–251.

[100] G. Ni, L. Qi, and M. Bai, *Geometric measure of entanglement and U-eigenvalues of tensors*, SIAM J. Matrix Anal. Appl., 35 (2014), no. 1, pp. 73–87.

[101] G. Ni, *Hermitian tensor and quantum mixed state*, `arXiv:1902.02640[quant-ph]`, preprint, 2019.

[102] J. Nie. *Certifying convergence of lasserre's hierarchy via flat truncation*. Math. Program., Ser. A, 142 (2013), no. 1-2, 485–510.

[103] J. Nie, *Optimality conditions and finite convergence of Lasserre's hierarchy*, Mathematical Programming **146** (2014), no. 1-2, Ser. A, 97–121.

[104] J. Nie, *The $\mathcal{A}$-truncated $K$-moment problem*, Foundations of Computational Mathematics **14** (2014), no. 6, 1243–1276.

[105] J. Nie, *Linear optimization with cones of moments and nonnegative polynomials*, Mathematical Programming **153** (2015), no. 1, 247–274.

[106] J. Nie. *Low rank symmetric tensor approximations*. SIAM J. Matrix Anal. Appl., 38(4):1517–1540, 2017.

[107] J. Nie, *Symmetric tensor nuclear norms*, SIAM J. Appl. Algebra Geometry, 1(1), 599–625, 2017.

[108] J. Nie. *Tight relaxations for polynomial optimization and lagrange multiplier expressions*. Mathematical Programming, 178(1), pp. 1–37, 2019.

[109] J. Nie, *Generating polynomials and symmetric tensor decompositions*, Found. Comput. Math., 17 (2017), no. 2, pp. 423–465.

[110] J. Nie and Z. Yang, *Hermitian Tensor Decompositions*, SIAM J. Matrix Anal. Appl. **41** (2020), no. 3, 1115-1144.

[111] J. Nie and K. Ye, *Hankel tensor decompositions and ranks*, SIAM J. Matrix Anal. Appl. **40** (2019), no. 2, 486–516.

[112] J. Nie and X. Zhang, *Positive maps and separable matrices*, SIAM J. Optim., 26 (2016), no. 2, pp. 1236–1256.

[113] J. Nie, *Linear optimization with cones of moments and nonnegative polynomials*, Math. Program., 153 (2015), pp. 247–274.

[114] J. Nie, *Polynomial optimization with real varieties*, SIAM J. Optim. 23(2013), no. 3, 1634–1646.

[115] J. Nie, *The $\mathcal{A}$-truncated $K$-moment problem*, Found. Comput. Math., 14 (2014), no. 6, pp. 1243–1276.

[116] J. Nie and K. Ranestad, *Algebraic degree of polynomial optimization*, SIAM J. Optim. 20(2009), no. 1, 485–502.

[117] J. Nie and K. Ye, *Hankel tensor decompositions and ranks*, SIAM J. Matrix Anal. Appl., 40 (2019), no. 2, pp. 486–516.

[118] J. Nie, Z. Yang, and G. Zhou, *The Saddle Point Problem of Polynomials*, Foundations of Computational Mathematics, pp. 1-37, 2021.

[119] J. Nie, Z. Yang, and X. Zhang, *A Complete Semidefinite Algorithm for Detecting Copositive Matrices and Tensors*, SIAM Journal on Optimization, Vol. 28, pp. 2902-2921, 2018.

[120] L. Oeding and G. Ottaviani, *Eigenvectors of tensors and algorithms for waring decomposition*, J. Symbolic Comput., 54 (2013), pp. 9–35.

[121] R. Pascanu, Y. Dauphin, S. Ganguli and Y. Bengio, *On the saddle point problem for non-convex optimization*, Preprint, 2014. `arXiv:1405.4604[cs.LG]`

[122] J. Peña, J. Vera, and L. Zuluaga. *Completely positive reformulations for polynomial optimization*. Math. Program., 151(2):405-431, 2014.

[123] H. Permuter, J. Francos, and I. Jermyn, *A study of Gaussian mixture models of color and texture features for image classification and segmentation*. Pattern Recognition, 39(4), 695–706, 2006.

[124] M. Putinar, *Positive polynomials on compact semi-algebraic sets*, Ind. Univ. Math. J., 42 (1993), pp. 203–206.

[125] L. Qi. *Symmetric nonnegative tensors and copositive tensors.* Linear Algebra and its Applications, 439(1):228–238, 2013.

[126] L. Qi and Z. Luo, *Tensor analysis: Spectral theory and special tensors*, SIAM, Philadelphia, 2017.

[127] L. Qi, G. Zhang, and G. Ni, *How entangled can a multi-party system possibly be?*, Physics Letters A, 382 (2018), no. 22, pp. 1465-1471.

[128] B. Reznick, *Some concrete aspects of Hilbert's* $17^{th}$ *problem*, Contemp. Math., 253 (2000), pp. 251–272.

[129] B. Romera-Paredes and M. Pontil, A New Convex Relaxation for Tensor Completion, *Advances in Neural Information Processing Systems 26*, 2967–2975, 2013.

[130] R. Schneider. *Convex bodies: the Brunn-Minkowski theory.* Encyclopedia of Mathematics and its Applications, Vol. 44. Cambridge University Press, Cambridge, 1993.

[131] M. Schweighofer. *Optimization of polynomials on compact semialgebraic sets.* SIAM J. Optim., 15(3), 805-825, 2005.

[132] P. Shah and P. Parrilo, *Polynomial stochastic games via sum of squares optimization*, Proceedings of the 46th IEEE Conference on Decision and Control, New Orleans, LA, USA, Dec. 12-14, 2007.

[133] N. Sidiropoulos and R. Bro, *On the uniqueness of multilinear decomposition of* $N$*-way arrays*, J. Chemometrics, 14 (2000), no, 3, pp. 229–239.

[134] Y. Song and L. Qi. *Necessary and sufficient conditions for copositive tensors.* Linear and Multilinear Algebra, 63 (2015), 120-131.

[135] Y. Song and L. Qi. *Properties of tensor complementarity problem and some classes of structured tensors.* arXiv preprint arXiv:1412.0113, 2014.

[136] Y. Song and L. Qi. *Tensor complementarity problem and semi-positive tensors.* J. Optim. Theory Appl., 169(3):1069–1078, 2016.

[137] L. Sorber, M. Van Barel, and L. De Lathauwer, *Optimization-based algorithms for tensor decompositions: canonical polyadic decomposition, decomposition in rank-*$(L_r, L_r, 1)$ *terms and a new generalization*, SIAM J. Optim., 23 (2013), no. 2, pp. 695–720.

[138] J. Sponsel, S. Bundfuss, and M. Dür. *An improved algorithm to test copositivity.* J. Global Optim., 52(3):537–551, 2012.

[139] J. Sturm. *SeDuMi 1.02: A MATLAB toolbox for optimization over symmetric cones*. Optim. Methods Softw., 11 & 12 (1999), pp. 625–653. `http://sedumi.ie.lehigh.edu`

[140] G. Tang and P. Shah, *Guaranteed tensor decomposition: a moment approach*, Proceedings of the 32nd International Conference on Machine Learning (ICML-15), pp. 1491-1500, 2015. *Journal of Machine Learning Research:* W&CP volume 37.

[141] M. Todd. *Semidefinite Optimization*. Acta Numerica, 10: 515–560, 2001.

[142] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, *Tensorlab 3.0*, March 2016, `http://www.tensorlab.net`.

[143] Y. Wu, P. Yang, *Optimal estimation of Gaussian mixtures via denoised method of moments*, Annals of Statistics, 48(4), pp. 1981–2007, 2020.

[144] K. Ye and L.-H. Lim, *Tensor network ranks*, `arXiv:1801.02662`, preprint, 2018

[145] M. Yuan and C.-H. Zhang, *On tensor completion via nuclear norm minimization*. Found. Comput. Math., 16(4), 1031–1068, 2016.

[146] I. Zabotin, *A subgradient method for finding a saddle point of a convex-concave function*, Issled. Prikl. Mat. 15(1988), 6–12.

[147] L. Zhu, T. Coleman and Y. Li, *Min-max robust CVaR robust mean-variance portfolios*, Journal of Risk 11(2009), no. 3, 55.