STRUCTURED TENSOR COMPUTATIONS: BLOCKING, SYMMETRIES AND KRONECKER FACTORIZATIONS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Stefan Ragnarsson

January 2012

STRUCTURED TENSOR COMPUTATIONS:

BLOCKING, SYMMETRIES AND KRONECKER FACTORIZATIONS

Stefan Ragnarsson, Ph.D.

Cornell University 2012

In this thesis we will explore the extensions of several ideas that have proven very successful in matrix computations to the rapidly maturing field of tensor computation. We will mainly focus on the use of blocking techniques, exploiting various different symmetries and developing new computational algorithms and factorizations.

In Chapter 2 we develop a novel method to embed a higher-order tensor in a larger, symmetric tensor. Such an embedding at the matrix level is well-known and has been used successfully to derive important matrix algorithms and is one of several ways of connecting the concepts of eigenvalues and singular values for matrices. Our method for higher-order tensors is a generalization of the matrix case, and we use it to derive a previously unknown connection between the concepts of tensor eigenvalues and tensor singular values. We also show how this symmetric tensor embedding can be used to generalize algorithms originally developed for symmetric tensors to arbitrary tensors while preserving their convergence properties.

Block tensors are becoming increasingly important within the field of numerical multilinear algebra. Accordingly, it is appropriate to develop an infrastructure that supports reasoning about block tensor computation. In Chapter 3 we establish concise notation that is suitable for the analysis and development of block tensor algorithms, prove several useful block tensor identities, and make precise the notion

of a block tensor unfolding.

In Chapter 4 we define a new block-based tensor operation that generalizes the matrix Kronecker product. Using this operation, we introduce a new tensor decomposition which extends the Kronecker Product SVD and has many attractive properties. The block unfoldings introduced in Chapter 3 play a pivotal role in the development of these tensor Kronecker methods.

Chapter 5 covers two special topics related to the overall theme of this thesis. First, a tensor decomposition based on the matrix QR decomposition with partial pivoting is introduced and its potential for low-rank approximation is explored. Then a power method that efficiently exploits the structure of partially symmetric tensors is proposed, and we investigate the singular value and singular vector properties of such tensors.

Overall, these results show how many ideas from matrix computations can be successfully extended to tensors. Just as matrix algorithms have increasingly been tuned to exploit structure such as blocking and symmetries, the same can be done in the tensor setting.

BIOGRAPHICAL SKETCH

Stefan Thor Ragnarsson-Torbergsen was born on September 27th, 1982 in Oslo, Norway to an Icelandic mother, Thorunn Sigurdardottir, and a Norwegian father, Ragnar Torbergsen. Stefan moved with his mother to Iceland in 1984 and lived in the west part of the quiet city of Reykjavik since 1986.

Stefan attended Menntaskolinn i Reykjavik, Iceland's oldest and most prestigious high school, from 1998 to 2002 and graduated with highest honors from the math and physics program. He then studied at the University of Iceland, graduating in 2005 with highest honors with a bachelor's degree in mathematics.

Stefan started graduate studies in 2005 at Cornell's Applied Mathematics program. He has worked with Professor Charles Van Loan since 2006, completing his A-exam in 2008.

Dedicated to my grandmother, Thorunn Gyda Arnadottir.

ACKNOWLEDGEMENTS

I would like to start by thanking Charlie Van Loan for being a fantastic thesis advisor. His support, guidance and insight have been invaluable during my time at Cornell University. He has been everything I could have wished for in an advisor and mentor.

I thank Éva Tardos and Lars Wahlbin for agreeing to be on my thesis committee and for the excellent courses they have taught me.

Tammy Kolda at Sandia National Laboratories deserves great praise for her joint work with Brett Bader on the Tensor Toolbox and for her illuminating research publications. I also thank her for several helpful hints and discussions and for giving me access to a beta version the latest iteration of the Toolbox before its release.

The administrative heroes of the Center for Applied Mathematics, Dolores Pendell and Selene Cammer, have my gratitude for helping me navigate the intricate bureaucracies of Cornell and expertly solving any problems along the way.

My friends and roommates have my deepest thanks for making my years in graduate school fun as well as productive. I would like to single out Chris Scheper, Steve Moseley and Anthony Sabelli for their support and friendship. You guys are awesome.

Last, but certainly not least, I thank my mother for her boundless support, generosity and love. Without her, none of this would have been possible.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

The primary theme of this thesis is exploitation of structure in tensor computations and the extension of successful methods from matrix computations to tensors. We develop new methods for block tensor computations, establish a new result connecting tensor eigenvalues and tensor singular values, and explore its algorithmic consequences, and create a new type of block-based tensor factorization that generalizes the Kronecker Product Singular Value Decomposition (KSVD).

In the rest of this introduction we survey the existing literature, introduce important tensor notation and decompositions, and give short previews of later chapters.

1.1 Overview of Tensor Computations

High-dimensional modeling is becoming more and more ubiquitous across the sciences and engineering because of advances in such areas as sensor technology and computer storage.

There is a large ongoing effort to broaden and generalize the successful tools provided by the numerical linear algebra community to higher-order tensors [20, 23, 41, 49, 72]. Tensors have been around in varying levels of abstraction for over a century and a half and play important roles in physics, engineering and mathematics. For example, Einstein's theory of relativity was written in tensor format.

The manipulation of tensors (multi-way arrays) involves *multilinear algebra*. As

a typical example of how tensors can arise is applications, the discretization of a continuous multivariate function on a grid yields a tensor. The element $\mathcal{A}(i,j,k,\ell)$ might be the value of $f(x,y,z,w)$ at gridpoint $(x,y,z,w) = (x_i, y_j, z_k, w_\ell)$. In information science applications, tensors can capture multi-way interaction, e.g. $\mathcal{A}(i,j,k,\ell)$ is some value that represents a 4-way interaction between variables.

The fields of chemometrics and psychometrics have developed infrastructure for tensor-based computations in the last four decades, independent of the numerical linear algebra community [41, 43, 65]. It was in these settings that many of the important tensor decompositions, such as CANDECOMP/PARAFAC (CP) and Tucker, were first proposed and used to analyze data [35, 69]. More recently, the HOSVD has received attention for its easy computation and attractive theoretical properties [20]. What these decompositions have in common is that they are all designed to extend one or more properties of the matrix SVD, such as orthogonality, optimal low-rank approximation and computational tractability. Unfortunately, it is impossible for a single tensor decomposition to capture all of the useful qualities of the SVD [20, 22]. Moreover, many of these decompositions are NP-hard to compute, and there are few efficient algorithms in use [31].

The important notion of rank and low-rank approximations for tensors is very different from the matrix case. There are several different definitions of rank for tensors, most notably the *outer-product rank* - the minimal number of rank-1 tensors needed to represent a given tensor. The outer product rank is strongly associated with the CP decomposition [44]. The notion of "best low-rank approximation" in this setting is ill-posed, quite unlike the case for matrices [22]. Often the best possible strategy is to use an iterative optimization algorithm to get an ap-

proximation for the best low-rank approximation. These algorithms have met with some success, but are often slow to converge and are susceptible to getting stuck in local optima. This can be somewhat mitigated by using cleverly constructed initial values [19, 36].

Up until now, tensor computations in the numerical linear algebra community have not focused much on structured tensors except for symmetric (sometimes called *supersymmetric*) tenors, although researchers in such fields as quantum chemistry have investigated other types of structured tensors. Symmetric tensors arise in many applications, such as higher-order statistics [50]. Several algorithms have been derived specifically for symmetric tensors, reducing computational cost and sometimes improving convergence, although they are also prone to converge to locally optimal solutions [37, 42]. There are also several concepts, such as tensor eigenvalues, which only make sense for symmetric tensors [61, 62].

A central challenge at the heart of tensor computations is the *curse of dimensionality*. It refers to the fact that as we add more dimensions to a tensor model, the total number of elements increases exponentially; even if each mode (i.e. dimension) is small, if there are enough of them the resulting tensor is huge. Such situations arise frequently in computational quantum chemistry [29, 77]. A promising development in this area is the concept of a *tensor network*, in which a high-order tensor is built by contracting many low-order tensors. These models have already proven very useful in many areas [6]. More such algorithms that scale with the number of dimensions are needed.

It is clear that as the field of tensor computation matures, exploiting structure and creating new kinds of factorizations and representations will be crucial.

1.2 Notation

An order-d tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is a real d-dimensional array $\mathcal{A}(1{:}n_1, \ldots, 1{:}n_d)$ where the index range in the k-th mode is from 1 to n_k.

Throughout this thesis we make use of the convenient notation defined in the Matlab Tensor Toolbox [40]. This toolbox supports fast prototyping of tensor algorithms and is extremely useful. It adds several new data types, but the ones we shall make most use of are the `tensor` and `tenmat` classes. The `tensor` class is basically a wrapper that contains a multidimensional array and allows for intuitive representation and provides many important functions. For example,

```
arr = rand(2,2,2);
```

is an order-3 multidimensional array with randomly assigned entries between 0 and 1. We can represent it as a tensor with the command

```
A = tensor(arr);
```

We can now do things with `A` such as contract along certain modes, compute norms and take inner products, which is not possible with multidimensional Matlab arrays.

The `tenmat` class can create matrix unfoldings of `tensor` objects, see §1.2.3 for more information. For more information on the Tensor Toolbox, see Appendix A or [39].

1.2.1 Index vectors, Permutations and Transposes

Since vectors of subscripts are prominent in the presentation, we elevate their notational status with boldface font, e.g., $\mathbf{p} = [\,4\;1\;2\;3\,]$. We let $\mathbf{1}$ denote the vector of ones and assume that dimension is clear from context. More generally, if N is an integer, then $\mathbf{N}$ is the vector of all N's. Finally, if $\mathbf{i}$ and $\mathbf{j}$ have equal length, then $\mathbf{i} \leq \mathbf{j}$ means that $i_k \leq j_k$ for all k.

If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $\mathbf{i} = (i_1, \ldots, i_d)$, then $\mathcal{A}(\mathbf{i})$ denotes component $(i_1, \ldots, i_d)$ of tensor $\mathcal{A}$. We use calligraphic characters to designate tensors, uppercase characters for matrices, lowercase characters for vectors and bold lower case characters to denote vectors of integers. For $\mathcal{A}(\mathbf{i})$ to make sense we must have $1 \leq i_k \leq n_k$ for $k = 1, \ldots, d$, i.e., $\mathbf{1} \leq \mathbf{i} \leq \mathbf{n}$. We refer to integer vectors of this form as an *index range vector* or simply *index vectors*.

If $\mathbf{p}$ is a permutation of $[\,1\;2\;\cdots\;d\,]$ and $\mathbf{i}$ is an index vector, then we define the permuted vector

$$\mathbf{i}(\mathbf{p}) = [\,i_{p_1}\;i_{p_2}\;\cdots\;i_{p_n}\,].$$

If $a < b$ and $c > 0$, then as in MATLAB we let $a\!:\!c\!:\!b$ denote the row vector $[a, a+c, a+2c, \cdots, a+mc]$ where $m = \lfloor (b-a)/c \rfloor$, i.e. the largest integer less than or equal to $(b-a)/c$. For example,

$$1\!:\!0.3\!:\!2 = [\,1\;\;1.3\;\;1.6\;\;1.9\,],$$

$$10\!:\!-2\!:\!1 = [\,10\;\;8\;\;6\;\;4\;\;2\,].$$

It is also handy to have a multi-index summation notation. If $\mathbf{n}$ is a length-d index vector, then

$$\sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \equiv \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d}. \tag{1.1}$$

For convenience we will sometimes use the functions `length` and `prod`. If $v \in \mathbb{R}^n$ then we define

$$\texttt{length}(v) \equiv n \quad \text{and} \quad \texttt{prod}(v) \equiv v_1 v_2 \cdots v_n. \tag{1.2}$$

For matrices, there is only one meaningful way of forming a transpose, i.e. by switching the rows and columns. However, for tensors of order d there are in general $(d-1)!$ different transposes. If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $\mathbf{p}$ is a permutation of $1{:}d$, then $\mathcal{A}^{<\mathbf{p}>} \in \mathbb{R}^{n_{p_1} \times \cdots \times n_{p_d}}$ denotes the $\mathbf{p}$-*transpose* of $\mathcal{A}$ and is defined by

$$\mathcal{A}^{<\mathbf{p}>}(i_{p_1}, \ldots, i_{p_d}) = \mathcal{A}(i_1, \ldots, i_d) \qquad \mathbf{1} \le \mathbf{i} \le \mathbf{n}, \tag{1.3}$$

i.e., $\mathcal{A}^{<\mathbf{p}>}(\mathbf{i}(\mathbf{p})) = \mathcal{A}(\mathbf{i})$.

The order-d tensor $\mathcal{A} \in \mathbb{R}^{n \times \cdots \times n}$ is said to be *symmetric* if $\mathcal{A} = \mathcal{A}^{<\mathbf{p}>}$ for any permutation $\mathbf{p}$ of $1{:}d$. Other types of tensor symmetries are considered in Chapters 4 and 5.

Example: If $\mathcal{A}$ is a matrix, then in this tensor notation we have $\mathcal{A}^T = \mathcal{A}^{<[2\ 1]>}$.

Example: If $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$, then there are $6 = 3!$ possible transpositions of $\mathcal{A}$:

$$\mathcal{B} = \left\{ \begin{array}{l} \mathcal{A}^{<[1\ 2\ 3]>} \\ \mathcal{A}^{<[1\ 3\ 2]>} \\ \mathcal{A}^{<[2\ 1\ 3]>} \\ \mathcal{A}^{<[2\ 3\ 1]>} \\ \mathcal{A}^{<[3\ 1\ 2]>} \\ \mathcal{A}^{<[3\ 2\ 1]>} \end{array} \right\} \quad \Longrightarrow \quad \left\{ \begin{array}{l} \mathcal{B}(i,j,k) \\ \mathcal{B}(i,k,j) \\ \mathcal{B}(j,i,k) \\ \mathcal{B}(j,k,i) \\ \mathcal{B}(k,i,j) \\ \mathcal{B}(k,j,i) \end{array} \right\} = \mathcal{A}(i,j,k) \tag{1.4}$$

for $i = 1, \ldots, n_1,\ j = 1, \ldots, n_2$ and $k = 1, \ldots, n_3$.

In MATLAB, if `A` is a `tensor` object and `p` is a permutation of 1:d then we can form the `p`-transpose with the command `permute(A,p)`. For example, the commands

```
A = tensor(rand(3,3,3,3));
p = [4 2 1 3];
A_perm = permute(A,p);
```

create a random $3 \times 3 \times 3 \times 3$ tensor and form its [4 2 1 3]-transpose.

The *outer product* $\mathcal{A} = \mathcal{B} \circ \mathcal{C}$ of tensors $\mathcal{B} \in \mathbb{R}^{j_1 \times \cdots \times j_d}$ and $\mathcal{C} \in \mathbb{R}^{k_1 \times \cdots \times k_e}$ is a tensor $\mathcal{A} \in \mathbb{R}^{j_1 \times \cdots \times j_d \times k_1 \times \cdots \times k_e}$ defined by

$$\mathcal{A}(\mathbf{i}) = \mathcal{B}(\,\mathbf{i}(1{:}d)\,) \cdot \mathcal{C}(\,\mathbf{i}(d{+}1{:}d{+}e)\,) \qquad \mathbf{1} \leq \mathbf{i} \leq [\,\mathbf{j}\ \mathbf{k}\,]. \tag{1.5}$$

The order of $\mathcal{B} \circ \mathcal{C}$ is the order of $\mathcal{B}$ plus the order of $\mathcal{C}$.

In the Tensor Toolbox, the tensor outer product of tensors $\mathcal{B}$ and $\mathcal{C}$, stored as `tensor` objects, can be formed with the `ttt` (**t**ensor **t**imes **t**ensor) function with the command

```
A = ttt(B,C);
```

where `A` will also be a `tensor` object.

A tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is said to be *rank-1* if for $k = 1, \ldots, d$ there exist

vectors $a_i \in \mathbb{R}^{n_i}$ such that for all $\mathbf{i} = \mathbf{1}, \ldots, \mathbf{n}$

$$\mathcal{A}(\mathbf{i}) = a_1(i_1)a_2(i_2)\cdots a_d(i_d) \tag{1.6}$$

and we denote this as $\mathcal{A} = a_1 \circ a_2 \circ \cdots a_d$.

Example: If $A \in \mathbb{R}^{n_1 \times n_2}$ and $A = a_1 \circ a_2$ for $a_1 \in \mathbb{R}^{n_1}, a_2 \in \mathbb{R}^{n_2}$ then $A = a_1 a_2^T$. For matrices, the tensor outer product is the same as the vector outer product.

1.2.2 Orderings

In numerical multilinear algebra it is frequently necessary to reshape a given tensor into a vector or a matrix and vice versa. In this section we collect results that make these maneuvers precise.

If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $N = n_1 \cdots n_d$, then $\text{vec}(\mathcal{A}) \in \mathbb{R}^N$ is a column vector defined recursively by

$$\text{vec}(\mathcal{A}) = \begin{bmatrix} \text{vec}(\mathcal{A}^{(1)}) \\ \vdots \\ \text{vec}(\mathcal{A}^{(n_d)}) \end{bmatrix} \tag{1.7}$$

where $\mathcal{A}^{(k)}$ is the order-$(d-1)$ tensor

$$\mathcal{A}^{(k)}(i_1, \ldots, i_{d-1}) = \mathcal{A}(i_1, \ldots, i_{d-1}, k) \qquad 1 \le k \le n_d. \tag{1.8}$$

It is assumed that $\mathbf{1} \le \mathbf{i}(1{:}d-1) \le \mathbf{n}(1{:}d-1)$. If $d = 1$, then $\mathcal{A}$ is a column vector and $\text{vec}(\mathcal{A}) = \mathcal{A}$. If $d = 2$, then $\mathcal{A}$ is a matrix and $\text{vec}(\mathcal{A})$ stacks its columns.

Each entry in tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ corresponds to a component of $\text{vec}(\mathcal{A})$. This implicitly defines an index mapping function $ivec(\cdot, \mathbf{n})$:

$$ivec(\mathbf{i}, \mathbf{n}) \;=\; i_1 + (i_2 - 1)n_1 + (i_3 - 1)n_1 n_2 + \cdots + (i_d - 1)n_1 \cdots n_{d-1}. \tag{1.9}$$

It is easy to show that if $v = \text{vec}(\mathcal{A})$, then

$$v_{ivec(\mathbf{i},\mathbf{n})} \;=\; \mathcal{A}(\mathbf{i}) \tag{1.10}$$

for all $\mathbf{i}$ that satisfy $\mathbf{1} \leq \mathbf{i} \leq \mathbf{n}$.

We also observe that if $w_k \in \mathbb{R}^{s_k}$ for $k = 1, \ldots, e$ then

$$w = w_e \otimes \cdots \otimes w_1 \qquad \Leftrightarrow \qquad w(ivec(\mathbf{i}, \mathbf{s})) = w_1(i_1) \cdots w_e(i_e). \tag{1.11}$$

1.2.3 Unfoldings

In order to *unfold* a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ into a matrix (also referred to in the literature as *flattening* or *matricizing* a tensor), it is necessary to choose (a) an integer e that satisfies $1 \leq e \leq d-1$ and (b) a permutation $\mathbf{p}$ of $1{:}d$. If

$$\mathbf{r} \;=\; \mathbf{p}(1{:}e) \tag{1.12}$$

$$\mathbf{c} \;=\; \mathbf{p}(e{+}1{:}d) \tag{1.13}$$

then the $\mathbf{r} \times \mathbf{c}$ *unfolding* of $\mathcal{A}$ is the matrix $\mathcal{A}_{\mathbf{r}\times\mathbf{c}}$ whose (α, β) entry is given by

$$\mathcal{A}_{\mathbf{r}\times\mathbf{c}}(\alpha, \beta) \;=\; \mathcal{A}^{<\mathbf{p}>}(i_1, \ldots, i_e, j_1, \ldots, j_{d-e}) \tag{1.14}$$

where

$$\alpha \;=\; ivec(\mathbf{i}, \mathbf{n}(\mathbf{r})) \qquad \mathbf{1} \leq \mathbf{i} \leq \mathbf{n}(\mathbf{r}) \tag{1.15}$$

$$\beta \;=\; ivec(\mathbf{j}, \mathbf{n}(\mathbf{c})) \qquad \mathbf{1} \leq \mathbf{j} \leq \mathbf{n}(\mathbf{c}). \tag{1.16}$$

Note that $\mathcal{A}_{\mathbf{r}\times\mathbf{c}}$ has $n_{p_1} \cdots n_{p_e}$ rows and $n_{p_{e+1}} \cdots n_{p_d}$ columns.

Example: Suppose $\mathcal{A} \in \mathbb{R}^{2\times3\times2\times2\times3}$. Then the $[\,1\ 2\ 3\,] \times [\,4\ 5\,]$ unfolding of $\mathcal{A}$ is

$$\mathcal{A}_{[1\ 2\ 3]\times[4\ 5]} \;=\; \begin{array}{c} \left[\begin{array}{cccccc} a_{11111} & a_{11121} & a_{11112} & a_{11122} & a_{11113} & a_{11123} \\ a_{21111} & a_{21121} & a_{21112} & a_{21122} & a_{21113} & a_{21123} \\ a_{12111} & a_{12121} & a_{12112} & a_{12122} & a_{12113} & a_{12123} \\ a_{22111} & a_{22121} & a_{22112} & a_{22122} & a_{22113} & a_{22123} \\ a_{13111} & a_{13121} & a_{13112} & a_{13122} & a_{13113} & a_{13123} \\ a_{23111} & a_{23121} & a_{23112} & a_{23122} & a_{23113} & a_{23123} \\ a_{11211} & a_{11221} & a_{11212} & a_{11222} & a_{11213} & a_{11223} \\ a_{21211} & a_{21221} & a_{21212} & a_{21222} & a_{21213} & a_{21223} \\ a_{12211} & a_{12221} & a_{12212} & a_{12222} & a_{12213} & a_{12223} \\ a_{22211} & a_{22221} & a_{22212} & a_{22222} & a_{22213} & a_{22223} \\ a_{13211} & a_{13221} & a_{13212} & a_{13222} & a_{13213} & a_{13223} \\ a_{23211} & a_{23221} & a_{23212} & a_{23222} & a_{23213} & a_{23223} \end{array}\right] \\ \begin{array}{cccccc} (1,1) & (2,1) & (1,2) & (2,2) & (1,3) & (2,3) \end{array} \end{array} \begin{array}{l} (1,1,1) \\ (2,1,1) \\ (1,2,1) \\ (2,2,1) \\ (1,3,1) \\ (2,3,1) \\ (1,1,2) \\ (2,1,2) \\ (1,2,2) \\ (2,2,2) \\ (1,3,2) \\ (2,3,2) \end{array}$$

where we have also highlighted the corresponding $\mathbf{i}$ and $\mathbf{j}$ indices from (1.15)-(1.16) and by (1.14) we have

$$\mathcal{A}(i_1, i_2, i_3, j_1, j_2) \;=\; \mathcal{A}_{[1\ 2\ 3]\times[4\ 5]}(i_1 + (i_2-1)\cdot 2 + (i_3-1)\cdot 2\cdot 3,\ j_1 + (j_2-1)\cdot 2).$$

Each row and column of $\mathcal{A}_{\mathbf{r}\times\mathbf{c}}$ is the vec of a reduced-order subtensor. In particular, for all $\mathbf{i}$ and $\mathbf{j}$ that satisfy $\mathbf{1} \le \mathbf{i} \le \mathbf{n}(\mathbf{r})$ and $\mathbf{1} \le \mathbf{j} \le \mathbf{n}(\mathbf{c})$, we have

$$\mathcal{A}_{\mathbf{r}\times\mathbf{c}}(\, ivec(\mathbf{i}, \mathbf{n}(\mathbf{r})),\, :\,) \;=\; \mathrm{vec}(\,\mathcal{R}^{(\mathbf{i})})^T \tag{1.17}$$

$$\mathcal{A}_{\mathbf{r}\times\mathbf{c}}(\, :\,, ivec(\mathbf{j}, \mathbf{n}(\mathbf{c}))\,) \;=\; \mathrm{vec}(\,\mathcal{C}^{(\mathbf{j})}) \tag{1.18}$$

where the tensors $\mathcal{R}^{(\mathbf{i})}$ and $\mathcal{C}^{(\mathbf{j})}$ are defined by

$$R^{(\mathbf{i})}(\mathbf{j}) = \mathcal{A}^{<\mathbf{p}>}(i_1, \ldots, i_e, j_1, \ldots, j_{d-e}) \tag{1.19}$$

$$C^{(\mathbf{j})}(\mathbf{i}) = \mathcal{A}^{<\mathbf{p}>}(i_1, \ldots, i_e, j_1, \ldots, j_{d-e}). \tag{1.20}$$

In the MATLAB Tensor Toolbox, the unfolding $\mathcal{A}_{\mathbf{r}\times\mathbf{c}}$ can be formed with the `tenmat` class, with the command

```
A_unfold = tenmat(A,r,c);
```

For example, the code

```
arr = reshape(1:8, [2 2 2]);
A = tensor(arr);
A_unfold = tenmat(A,[2 3],1);
```

forms the $2 \times 2 \times 2$ tensor where

$$\mathcal{A}(:,:,1) = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \quad \text{and} \quad \mathcal{A}(:,:,2) = \begin{bmatrix} 5 & 7 \\ 6 & 8 \end{bmatrix}$$

and then produces the unfolding matrix

$$\mathcal{A}_{[2\ 3]\times[1]} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \\ 7 & 8 \end{bmatrix}.$$

See Appendix A for more details on the `tenmat` class.

Particularly important are the so-called *modal unfoldings*. If we define the permutation $\mathbf{p} = [\,k, 1{:}k{-}1, k{+}1{:}d\,]$ and $e = 1$, then $\mathcal{A}_{\mathbf{r}\times\mathbf{c}}$ is the *mode-k unfolding*

of $\mathcal{A}$. The columns of this matrix are referred to as *mode-k fibers* of $\mathcal{A}$. Note that if we define the vectors

$$\tilde{\mathbf{n}} = [\, \mathbf{n}(1{:}k{-}1) \quad \mathbf{n}(k{+}1{:}d)\,], \tag{1.21}$$

$$\tilde{\mathbf{i}} = [\, \mathbf{i}(1{:}k{-}1) \quad \mathbf{i}(k{+}1{:}d)\,], \tag{1.22}$$

then

$$\mathcal{A}_{(k)}(i_k, ivec(\tilde{\mathbf{i}}, \tilde{\mathbf{n}})) = \mathcal{A}(\mathbf{i}) \qquad \mathbf{1} \le \mathbf{i} \le \mathbf{n}. \tag{1.23}$$

This matrix has n_k rows and $n_1 \cdots n_{k-1} n_{k+1} \cdots n_d$ columns. A third-order instance of this important concept is displayed in equation (1.24).

Example: Suppose $n_1 = 4$, $n_2 = 3$, and $n_3 = 2$. then the modal unfoldings $\mathcal{A}_{(1)}$, $\mathcal{A}_{(2)}$, and $\mathcal{A}_{(3)}$ are

$$\begin{aligned}
\mathcal{A}_{(1)} &= \begin{bmatrix} a_{111} & a_{121} & a_{131} & a_{112} & a_{122} & a_{132} \\ a_{211} & a_{221} & a_{231} & a_{212} & a_{222} & a_{232} \\ a_{311} & a_{321} & a_{331} & a_{312} & a_{322} & a_{332} \\ a_{411} & a_{421} & a_{431} & a_{412} & a_{422} & a_{432} \end{bmatrix} \\
\mathcal{A}_{(2)} &= \begin{bmatrix} a_{111} & a_{211} & a_{311} & a_{411} & a_{112} & a_{212} & a_{312} & a_{412} \\ a_{121} & a_{221} & a_{321} & a_{421} & a_{122} & a_{222} & a_{322} & a_{422} \\ a_{131} & a_{231} & a_{331} & a_{431} & a_{132} & a_{232} & a_{332} & a_{432} \end{bmatrix} \\
\mathcal{A}_{(3)} &= \begin{bmatrix} a_{111} & a_{211} & a_{311} & a_{411} & a_{121} & a_{221} & a_{321} & a_{421} & a_{131} & a_{231} & a_{331} & a_{431} \\ a_{112} & a_{212} & a_{312} & a_{412} & a_{122} & a_{222} & a_{322} & a_{422} & a_{132} & a_{232} & a_{332} & a_{432} \end{bmatrix}.
\end{aligned} \tag{1.24}$$

The columns of these matrices are the modal fibers. A fiber of a tensor is obtained by fixing all but one of the indices. For example, the third column of the unfolding

$$\mathcal{A}_{(1)} = \begin{bmatrix} \mathcal{A}(:,1,1) & \mathcal{A}(:,2,1) & \mathcal{A}(:,3,1) & \mathcal{A}(:,1,2) & \mathcal{A}(:,2,2) & \mathcal{A}(:,3,2) \end{bmatrix}$$

is the fiber

$$\mathcal{A}(:,3,1) = \begin{bmatrix} \mathcal{A}(1,3,1) \\ \mathcal{A}(2,3,1) \\ \mathcal{A}(3,3,1) \\ \mathcal{A}(4,3,1) \end{bmatrix}$$

obtained by fixing the 2-mode index at 3 and the 3-mode index at 1.

The mode-k unfolding can be formed in MATLAB with the command

```
tenmat(A,k)
```

For example, the code

```
arr = reshape(3*(0:17),[3 3 2])
A = tensor(arr);
A_3 = tenmat(A,3);
```

creates the $3 \times 3 \times 2$ tensor $\mathcal{A}$ where

$$\mathcal{A}(:,:,1) = \begin{bmatrix} 0 & 9 & 18 \\ 3 & 12 & 21 \\ 6 & 15 & 24 \end{bmatrix} \quad \text{and} \quad \mathcal{A}(:,:,2) = \begin{bmatrix} 27 & 36 & 45 \\ 30 & 39 & 48 \\ 33 & 42 & 51 \end{bmatrix}$$

and then creates the unfolding

$$\mathcal{A}_{(3)} = \mathcal{A}_{[3]\times[1\ 2]} = \begin{bmatrix} 0 & 3 & 6 & 9 & 12 & 15 & 18 & 21 & 24 \\ 27 & 30 & 33 & 36 & 39 & 42 & 45 & 48 & 51 \end{bmatrix}.$$

1.2.4 The Multilinear Product

The most fundamental operations in matrix computations are matrix-vector and matrix-matrix multiplication. The higher-order analogue of this operation is the *multilinear product* (also known as the *multilinear transform*). Suppose $\mathcal{F} \in \mathbb{R}^{s_1 \times \cdots \times s_d}$ and $B_k \in \mathbb{R}^{s_k \times t_k}$ for $k = 1, \ldots, d$. The tensor $\mathcal{T} \in \mathbb{R}^{t_1 \times \cdots \times t_d}$ defined by

$$\mathcal{T}(\mathbf{i}) \;=\; \sum_{\mathbf{j=1}}^{\mathbf{s}} \mathcal{F}(\mathbf{j}) B_1(j_1, i_1) B_2(j_2, i_2) \cdots B_k(j_k, i_k) \tag{1.25}$$

is the multilinear product of tensor $\mathcal{F}$ by the matrices $B_1, \ldots, B_d$ and is denoted by

$$\mathcal{T} \;=\; \mathcal{F} \cdot (B_1, B_2, \ldots, B_d). \tag{1.26}$$

See [22] for details. We also define

$$(B_1, B_2, \ldots, B_d) \cdot \mathcal{F} \;\;\equiv\;\; \mathcal{F} \cdot (B_1^T, B_2^T, \ldots, B_d^T). \tag{1.27}$$

Some of the key summations and vectors in this thesis can be expressed neatly through this transformation.

Note that if for some $1 \leq k \leq d$ we have $t_k = 1$, i.e B_k is a column vector, then $\mathcal{T}$ is in fact an order-$(d-1)$ tensor. If all of the B_k are vectors, i.e. $t_k = 1$ for all $k = 1, \ldots, d$, then $\mathcal{T}$ is a scalar.

Another important special case is where for some k the matrix B_k is the $s_k \times s_k$ identity matrix I_{s_k}. For example, if $B_1 = I_{s_1}$ then

$$\begin{aligned}\mathcal{T}(\mathbf{j}) &= \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{s}} \mathcal{F}(\mathbf{i}) B_1(i_1, j_1) B_2(i_2, j_2) \cdots B_d(i_d, j_d) \\ &= \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{s}} \mathcal{F}(\mathbf{i}) I_{s_1}(i_1, j_1) B_2(i_2, j_2) \cdots B_d(i_d, j_d) \\ &= \sum_{\mathbf{i}(2:d)=\mathbf{1}}^{\mathbf{s}(2:d)} \mathcal{F}(j_1, \mathbf{i}(2:d)) B_2(i_2, j_2) \cdots B_d(i_d, j_d).\end{aligned}$$

That is to say, an identity matrix in mode k corresponds to not contracting over that mode at all. An important special case is where all the B_k matrices are column vectors except the jth one which is an identity matrix, e.g. if $u_k \in \mathbb{R}^{s_k}$ for $k \in \{1, \ldots, j-1, j+1, \ldots, d\}$ then

$$v \;=\; \mathcal{F} \cdot (u_1, \ldots, u_{j-1}, I_{s_j}, u_{j+1}, \ldots, u_d) \in \mathbb{R}^{s_j}$$

is a column vector.

Example: If $A \in \mathbb{R}^{n_1 \times n_2}, U \in \mathbb{R}^{n_1 \times m_1}, V \in \mathbb{R}^{n_2 \times m_2}$ and $x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}$ then

$$\begin{aligned} A \cdot (U, V) &= U^T A V & \qquad A \cdot (x, y) &= x^T A y \\ A \cdot (I_{n_1}, V) &= AV & \qquad A \cdot (I_{n_1}, y) &= Ay \\ A \cdot (U, I_{n_2}) &= U^T A & \qquad A \cdot (x, I_{n_2}) &= x^T A \end{aligned}$$

Example: If $\mathcal{F} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $u_1 \in \mathbb{R}^{n_1}, u_3 \in \mathbb{R}^{n_3}$ then $v = \mathcal{F}(u_1, I_{n_2}, u_3) \in \mathbb{R}^{n_2}$ is given by

$$v(j) \;=\; \sum_{i_1=1}^{n_1} \sum_{i_3=1}^{n_3} \mathcal{F}(i_1, j, i_3) u_1(i_1) u_3(i_3)$$

for all $j = 1, \ldots, n_2$.

The Tensor Toolbox provides functionality to compute the multilinear product throughout the functions `ttm` (**t**ensor **t**imes **m**atrix) and `ttv` (**t**ensor **t**imes **v**ector). For example, if $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$ and $U_k \in \mathbb{R}^{m_k \times n_k}$ for $k = 1, 2, 3$ then the product

$$(U_1, U_2, U_3) \cdot \mathcal{A}$$

can be computed with

```
ttm(A,{U1,U2,U3})
```

and if $u_k \in \mathbb{R}^{n_k}$ for $k = 1, 2, 3$ then $(u_1, u_2, u_3) \cdot \mathcal{A}$ is computed with

```
ttv(A,{u1,u2,u3})
```

1.2.5 Norms and Inner Products

The *inner product* of two tensors $\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is defined by

$$\langle \mathcal{X}, \mathcal{Y} \rangle \;=\; \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \mathcal{X}(\mathbf{i})\mathcal{Y}(\mathbf{i}) \tag{1.28}$$

and the *norm* of a tensor $\mathcal{X}$ is

$$\|\mathcal{X}\| \;=\; \sqrt{\langle \mathcal{X}, \mathcal{X} \rangle}. \tag{1.29}$$

Note that for order-2 tensors (1.29) reduces to the Frobenius matrix norm.

1.2.6 Two Special Matrix Products

We will occasionally use some of the following useful matrix operations.

If F and G are m-by-n matrices then the *Hadamard product* of F and G, denoted $F * G$, is the m-by-n matrix whose elements are given by

$$(F * G)(i,j) \equiv F(i,j) \cdot G(i,j). \tag{1.30}$$

The Hadamard product is sometimes also called the "entry-wise product" for obvious reasons.

If $X \in \mathbb{R}^{p \times s}$ and $Y \in \mathbb{R}^{q \times s}$ then their *Khatri-Rao product* is defined as

$$X \odot Y \equiv [\, x_1 \otimes y_1 \;\; \cdots \;\; x_k \otimes y_k \,] \in \mathbb{R}^{pq \times s}. \tag{1.31}$$

The Khatri-Rao product is thus a columnwise Kronecker product. The following lemma lists two important properties of the Khatri-Rao and Hadamard products. See [65] for a more detailed discussion.

Lemma 1.1. *Let $A \in \mathbb{R}^{p \times s}, B \in \mathbb{R}^{q \times s}$ and $C \in \mathbb{R}^{r \times s}$. Then*

(a) $(A \odot B)^T (A \odot B) = (A^T A) * (B^T B)$, *and*

(b) $(A \odot B)^\dagger = ((A^T A) * (B^T B))^\dagger (A \odot B)^T$

where $M^\dagger$ is the Moore-Penrose pseudo-inverse of a matrix M.

1.3 Tensor Rank

There are many different ways of extending the notion of rank to higher order tensors and are appropriate for different situations [22, 41].

The *multilinear rank* of $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is the d-tuple

$$\operatorname{rank}_{\boxplus}(\mathcal{A}) \equiv (r_1(\mathcal{A}), r_2(\mathcal{A}), \ldots, r_d(\mathcal{A}))$$

where $r_i(\mathcal{A}) = \text{rank}(\mathcal{A}_{(i)})$. The multilinear tensor rank is important because if $\text{rank}_{\boxplus}(\mathcal{A}) = [\, m_1 \ \cdots \ m_d \,]$ then it is possible to decompose $\mathcal{A}$ as

$$\mathcal{A} \ = \ (X_1, \ldots, X_d) \cdot \mathcal{B}$$

where $\mathcal{B} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ and $X_i \in \mathbb{R}^{n_i \times m_i}$ have orthogonal columns for $i = 1, \ldots, d$. The multilinear rank is the smallest size of the tensor $\mathcal{B}$ for which this decomposition is possible [22].

The *outer product rank* of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is the smallest number R of rank-1 tensors needed to represent $\mathcal{A}$ as a sum of the form

$$\mathcal{A} \ = \ \sum_{i=1}^{R} a_1^{(i)} \circ \cdots \circ a_d^{(i)} \tag{1.32}$$

where $a_j^{(i)} \in \mathbb{R}^{n_j}$ for all $i = 1, \ldots, R$ and $j = 1, \ldots, d$. We denote the outer product rank as

$$R \ = \ \text{rank}_{\circledcirc}(A). \tag{1.33}$$

Unfortunately, determining the outer product rank of a higher-order tensor is NP-hard and is often an ill-posed problem [22, 31].

If $\mathcal{A} \in \mathbb{R}^{n \times \cdots \times n}$ is a symmetric tensor then the *symmetric rank* of $\mathcal{A}$ is the minimal number r of symmetric rank-1 tensors needed to represent it in the form

$$\mathcal{A} \ = \ \sum_{i=1}^{r} v^{(i)} \circ \cdots \circ v^{(i)}. \tag{1.34}$$

See [12] for a detailed discussion. The symmetric rank is often denoted as $\text{rank}_{\text{S}}(\mathcal{A})$. It is obvious that $\text{rank}_{\circledcirc}(A) \leq \text{rank}_{\text{S}}(\mathcal{A})$, but it is an open question if equality holds in equal. This is in sharp contrast with the case for symmetric matrices, for which the outer product rank and symmetric rank are easily shown to be the same.

1.4 Standard Tensor Decompositions

We present some of the more common tensor decompositions that will figure heavily in later chapters. See [41] for a more detailed overview of tensor decompositions and related computation.

1.4.1 The CP Decomposition

An order-d tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ can always be decomposed into a sum of rank-1 tensors of the form

$$\mathcal{A} \;=\; \sum_{i=1}^{R} a_1^{(i)} \circ \cdots \circ a_d^{(i)} \tag{1.35}$$

where $a_j^{(i)} \in \mathbb{R}^{n_j}$ for all $i = 1, \ldots, R$ and $j = 1, \ldots, d$. When $R = \text{rank}_{\circledcirc}(\mathcal{A})$ then (1.35) is called the *CP decomposition* [4, 28, 35]. Note that this decomposition is also referred to as the *Canonical Decomposition* (CANDECOMP) and the *Parallel Factors* (PARAFAC) decomposition in the literature.

The CP-decomposition is often written as in normalized form as

$$\mathcal{A} \;=\; \sum_{i=1}^{R} \lambda_i b_1^{(i)} \circ \cdots \circ b_d^{(i)} \tag{1.36}$$

where the b-vectors have unit length.

Unfortunately there is no straightforward way to determine the rank of a specific tensor. For example, there is a $9 \times 9 \times 9$ tensor whose rank is only known to be bounded between 18 and 23 [44]. There is no result about the maximum outer product rank of a general tensor although there are some special cases that have been investigated [33, 45]. Furthermore, the least squares problem

$$\min_{\text{rank}_{\circledcirc}(\mathcal{B}) \leq k} \| \mathcal{A} - \mathcal{B} \|$$

of approximating a given tensor $\mathcal{A}$ optimally with a rank-k tensor $\mathcal{B}$ is ill-posed [22], i.e. there are large classes of tensors that have the property that $\text{rank}_{\circledcirc}(\mathcal{A}) = R$ but there exists a sequence $\{\mathcal{A}_i\}$ of tensors where $\text{rank}_{\circledcirc}(\mathcal{A}_i) < R$ but $\lim \mathcal{A}_i = \mathcal{A}$. For a detailed discussion of the outer product rank, see [22].

1.4.2 Tucker Decomposition

The Tucker model is a form of higher-order principal component analysis. It decomposes a tensor into a *core tensor* multiplied by a matrix in each mode [69]. The form of the decomposition of $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ in terms of the multilinear product is

$$\mathcal{A} \;=\; (X_1, X_2, \ldots, X_d) \cdot \mathcal{C} \tag{1.37}$$

where $\mathcal{C} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ is the core tensor and the matrices $X_i \in \mathbb{R}^{n_i \times m_i}$ are the *factor matrices*, which are often chosen to be orthogonal (although this not necessary to be considered a Tucker decomposition). Note that $m_i \leq n_i$ for $i = 1, \ldots, d$. The factor matrices can be thought of as the principal components in each mode and the entries in the core tensor as the level of interaction between the different components. If $m_i < n_i$ for some i then $\mathcal{C}$ can be considered a compressed version of the tensor $\mathcal{A}$.

Interestingly, the CP decomposition can be viewed as a special case of Tucker where the core $\mathcal{C}$ is *diagonal*, i.e. $\mathcal{C}(i_1, \ldots, i_d) = 0$ unless $i_1 = i_2 = \cdots = i_d$, and $m_i = R$ for all $i = 1, \ldots, d$.

1.4.3 The Higher Order SVD

The *Higher-Order Singular Value Decomposition* (HOSVD) of de Lathauwer et al. [20] is an important tensor decomposition that can be written using the multilinear product. It is a special case of the Tucker decomposition (1.37). Since it will come up frequently in this thesis, we will summarize some of its properties derived in [20]. If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ then the HOSVD of $\mathcal{A}$ is

$$\mathcal{A} \;=\; (U_1, \ldots, U_d) \cdot \mathcal{S}, \tag{1.38}$$

where $\mathcal{S} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is the core tensor and $U_i \in \mathbb{R}^{n_i \times n_i}$ are orthogonal. The core tensor $\mathcal{S}$ has the special property of being "all-orthogonal". The factor matrices U_k in (1.38) are the matrices of the right singular vectors of the modal unfoldings $\mathcal{A}_{(k)}$. Thus, we have for all $k = 1, \ldots, d$,

$$\mathcal{A}_{(k)} \;=\; U_k \Sigma_k V_k^T \tag{1.39}$$

and from (1.38) and (1.39) we can show that for all k,

$$\mathcal{S}_{(k)} \;=\; \Sigma_k V_k^T (U_d \otimes U_{d-1} \otimes \cdots \otimes U_{k+1} \otimes U_{k-1} \otimes \cdots \otimes U_1). \tag{1.40}$$

If $\mathcal{T}$ is an order-d tensor, we denote by $\mathcal{T}_{i_k=\gamma}$ the order-$(d-1)$ subtensor of $\mathcal{T}$ obtained by fixing the kth index to be γ, i.e.

$$(\mathcal{T}_{i_k=\gamma})(j_1, \ldots, j_{d-1}) \;=\; \mathcal{T}(j_1, \ldots, j_{k-1}, \gamma, j_k, \ldots, j_{d-1}).$$

From equation (1.40) we can see that the HOSVD has several interesting properties. For example, the inner product of the subtensors of $\mathcal{S}$ when any index i_k, $1 \leq k \leq d$, is fixed to different values is zero, i.e.

$$\langle \mathcal{S}_{i_k=\alpha}, \mathcal{S}_{i_k=\beta} \rangle \;=\; \delta_{\alpha\beta}.$$

This is the *all-orthogonality* property and easily follows from the orthogonality of the U and V matrices.

The following subtensor norm grading relationship also holds:

$$\| \mathcal{S}_{i_k=1} \| \;=\; \sigma_1^{(k)} \;\geq\; \| \mathcal{S}_{i_k=2} \| \;=\; \sigma_2^{(k)} \;\geq\; \cdots \;\geq\; \| \mathcal{S}_{i_k=n_k} \| \;=\; \sigma_{n_k}^{(k)},$$

which means that the norms of the rows of the the modal unfolding $\mathcal{S}_{(k)}$ are monotonically decreasing. Furthermore, for any $k = 1, \ldots, d$ we have that $\| \mathcal{A} \|^2 = \| \mathcal{S} \|^2 = \sum_{i_k=1}^{R_k} \left(\sigma_{i_k}^{(k)} \right)^2$, where R_k is the rank of $\mathcal{A}_{(k)}$.

If we truncate the HOSVD to $\hat{\mathcal{A}} = (U_1', U_2', \ldots, U_d') \cdot \mathcal{S}'$ where $\mathcal{S}' \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ is equal to $\mathcal{S}(1{:}m_1, \ldots, 1{:}m_d)$ and $U_k' = U_k(:, 1{:}m_k)$ where $m_k \leq n_k$ for $k = 1, \ldots, d$, then the approximation error is bounded by the inequality

$$\| \mathcal{A} - \hat{\mathcal{A}} \| \;\leq\; \sum_{k=1}^{d} \sum_{i_k=m_k+1}^{R_k} \left(\sigma_{i_k}^{(k)} \right)^2. \tag{1.41}$$

Note that in general $\hat{\mathcal{A}}$ is *not* the optimal tensor approximation of $\mathcal{A}$ of size $m_1 \times \cdots \times m_d$ but is often a reasonable approximation and a good initial guess for algorithms such as the higher-order power method (HOPM) and higher-order orthogonal iteration (HOOI) [19, 21].

Example: If $\mathcal{A}$ is a matrix, i.e. $d = 2$, the equation $\mathcal{A} = (U_1, U_2) \cdot \mathcal{S}$ reduces to $A = U_1 S U_2^T$, where

$$\begin{aligned} A_{(1)} \;&=\; A \;=\; U \Sigma V^T \;=\; U_1 \Sigma_1 V_1^T, \\ A_{(2)} \;&=\; A^T \;=\; V \Sigma^T U^T \;=\; U_2 \Sigma_2 V_2^T, \end{aligned}$$

which implies $U_1 = V_2 \equiv U$ and $U_2 = V_1 \equiv V$. We also get

$$\begin{aligned} S_{(1)} &= S = \Sigma_1 V_1^T U_2 = \Sigma V^T V = \Sigma, \\ S_{(2)} &= S^T = \Sigma_2 V_2^T U_1 = \Sigma^T U^T U = \Sigma^T. \end{aligned}$$

Thus the HOSVD reduces to the matrix SVD when $d = 2$.

1.5 Preview Of Chapters 2-5

In Chapter 2 we extend the following well-known trick from matrix computations to tensors. If $A \in \mathbb{R}^{n_1 \times n_2}$, then there are well-known connections between its singular value decomposition (SVD) and the eigenvalue and eigenvector properties of the symmetric matrix

$$\text{sym}(A) = \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)}. \tag{1.42}$$

Specifically, the eigenvalues of $\text{sym}(A)$ are the singular values of A and their negatives and the associated eigenvectors are the column stacking of the left and right singular vectors. This has been used to analyze and derive important matrix algorithms in the past [25].

There is also an important connection between A and $\text{sym}(A)$ through their Rayleigh quotients and their critical values, which are closely related to eigenvalues and singular values.

We discuss all of these notions as they apply to higher-order tensors. We develop a new way to embed a tensor into a symmetric tensor of larger size. In analyzing this embedding we discover a new result concerning the connection be-

tween tensor singular values and tensor eigenvalues and show how this can be used to develop new algorithms.

The tensor rank properties of the symmetric embedding are also investigated. For matrices we have that $\text{rank}(\text{sym}(A)) = 2 \cdot \text{rank}(A)$. We prove a partial generalization of this result for higher-order tensors and discuss its similarity to an unsolved conjecture on tensor rank.

Chapter 3 develops a robust computational framework for block tensor computations. The field of matrix computations has matured to the point that it is not necessary to provide scalar-level verifications of basic block-level operations. For example, if

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^T \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix},$$

then without "ijk proof "it is <u>understood</u> that $C_{12} = A_{11}^T B_{12} + A_{21}^T B_{22}$ provided A and B are partitioned conformally. "Understandings" like this contribute to the culture of block matrix computations, enabling researchers to think at a high level when they are developing new algorithms and proofs.

It is our contention that the emerging field of tensor computations needs to develop a similar infrastructure that gracefully supports block tensor operations. By a *block tensor* we mean a tensor whose entries are themselves tensors. As with matrices, the act of blocking a tensor is the act of partitioning the index range vectors associated with each dimension. Thus, if $\mathcal{A} \in \mathbb{R}^{9\times 5\times 8}$ and

$$\begin{aligned} 1\!:\!9 &= \left[\, 1\!:\!2 \,\middle|\, 3\!:\!5 \,\middle|\, 6\!:\!9 \,\right] \\ 1\!:\!5 &= \left[\, 1\!:\!3 \,\middle|\, 4\!:\!5 \,\right] \\ 1\!:\!8 &= \left[\, 1\!:\!2 \,\middle|\, 3\!:\!4 \,\middle|\, 5\!:\!6 \,\middle|\, 7\!:\!8 \,\right], \end{aligned} \tag{1.43}$$

then we are choosing to regard $\mathcal{A}$ as a 3-by-2-by-4 block tensor with block dimensions that are determined by the indicated partitionings of 1:9, 1:5, and 1:8. The colon notation can be used to specify the blocks. For example, the (2,1,3) block $\mathcal{A}_{213}$, is prescribed by $\mathcal{A}$(3:5, 1:3, 5:6).

Block tensors are increasingly important [16, 17, 18, 32, 56] for the same reasons that block matrices are increasingly important:

1. *Structure.* Block-level sparsity is a common pattern because of nearest-neighbor coupling and other reasons.
2. *Generalization.* Block versions of point algorithms frequently have attractive features [56].
3. *Performance.* Blocking is the key to minimizing communication [2].

A common paradigm for tensor computation involves unfolding. In this framework, computations on a tensor $\mathcal{A}$ reduce to matrix computations on one or more of its unfoldings. For example, the higher-order singular value decomposition of a tensor involves computing the SVD of each modal unfolding [20].

Given all the advantages that result when a matrix computation is organized at the block level, it makes sense for an unfolding of a block tensor $\mathcal{A}$ to have a related block structure of its own. In particular, $\mathcal{A}$'s blocks should map to contiguous blocks in the unfolding. This is not the case when a typical "vec-oriented" unfolding is invoked [41]. Chapter 3 shows how to permute the rows and columns of a vec-oriented unfolding so that its blocks are unfoldings of the the tensor blocks and how tensor computations can be done with these block unfoldings in a natural way.

Armed with our framework for block tensor computations, the next step is to explore block-based tensor approximations and decompositions. The matrix *Kronecker Product SVD* (KSVD) [58] is a way of approximating a matrix A as a Kronecker product of two smaller matrices, i.e.

$$\min_{B,C} \| A - B \otimes C \|_F$$

and computing a minimal exact decomposition

$$A = \sum_{i=1}^{R} B_i \otimes C_i.$$

The B and C matrices can even be chosen to inherit some of A's structure, such as symmetry or non-negativity. In Chapter 4 we show how the block unfoldings developed in Chapter 3 allow us to generalize the KSVD to tensors and investigate how this *tensor KSVD* captures the more complicated symmetries possible for higher-order tensors as well as so-called multilevel tensors that have block-level structure. We define a new operation called the *Tensor Kronecker Product*, whose properties we will explore in some detail. Together, the TKSVD and Tensor Kronecker Product give us an easily computed, data-sparse tensor approximation that allows for efficient implementations of many common tensor operations.

A few special topics are considered in Chapter 5. We introduce a new Tucker decomposition reminiscent of the HOSVD but based on computing a QR decomposition with partial pivoting on the modal unfoldings. The resulting decomposition is called the Higher Order QR Decomposition (HOQRD) and is of the form

$$\mathcal{A} = (Q_1, \ldots, Q_d) \cdot \mathcal{R}.$$

Due to the partial pivoting strategy used to obtain the $\mathcal{R}$ tensor and $Q_1, \ldots, Q_d$ matrices there is a "norm grading" in the core tensor $\mathcal{R}$: entries tend to get smaller the further they are from the $(1, \ldots, 1)$ position. Due to this effect a

truncated HOQRD approximation can be a good initial guess for iterative low-rank approximation algorithms such as the HOPM and HOOI.

The last topic we explore are the tensor singular value properties of partially symmetric tensors. A partially symmetric tensor is invariant with respect to some permutation of its indices but is not necessarily fully symmetric. An example is a 3rd order tensor $\mathcal{A}$ such that $\mathcal{A}(i,j,k) = \mathcal{A}(j,i,k)$ for all i,j,k. We theorize that the singular vectors of such tensors can be chosen to reflect this partial symmetry. A power method that computes singular values and singular vectors of partially symmetric tensors that exploits their structure to reduce computation is proposed.

CHAPTER 2

BLOCK TENSORS AND SYMMETRIC EMBEDDINGS

This chapter introduces a novel way of embedding a tensor in a larger symmetric tensor, extending the classical matrix case, and investigates its tensor singular value and eigenvalue connections, algorithmic implications and rank properties.

The idea of embedding a general tensor into a larger symmetric tensor having the same order is developed in §2.2. This requires having a facility with block tensors. Fundamental orderings, unfoldings, and multilinear summations are discussed in §2.3 and used to characterize various multilinear Rayleigh quotients and their stationary values and vectors. This builds on the variational approach to tensor singular values developed in [48]. In §2.4 we provide a symmetric embedding analysis of several higher-order power methods for tensors that have recently been proposed [19, 21, 36, 37, 42]. Results that relate the multilinear and outer product ranks of a tensor to the corresponding ranks of its symmetric embedding are presented in §2.5. A brief conclusion section follows.

2.1 Motivations

Before proceeding with the discussion of the general tensor symmetric embedding we present two motivational examples. First, we discuss the properties of the matrix symmetric embedding, which has a long history in matrix computations. We then explore the symmetric embedding of an order-3 tensor, which allows for an intuitive visualization and simplified notation.

2.1.1 The Matrix Case

If $A \in \mathbb{R}^{m\times n}$ we define

$$\mathbf{sym}(A) \;=\; \begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \in \mathbb{R}^{(n_1+n_2)\times(n_1+n_2)} \tag{2.1}$$

which is a symmetric matrix and has the property that if $A = U\Sigma V^T$ is the SVD of A, then for $k = 1, 2, \ldots, \text{rank}(A)$

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix} \begin{bmatrix} u_k \\ \pm v_k \end{bmatrix} \;=\; \pm\sigma_k \begin{bmatrix} u_k \\ \pm v_k \end{bmatrix} \tag{2.2}$$

where $u_k = U(:,k)$, $v_k = V(:,k)$, and $\sigma_k = \Sigma(k,k)$. In other words, the eigenvalues of $\mathbf{sym}(A)$ are the singular values of A and their negatives and the corresponding eigenvectors are the column stacking of the singular vectors of A. This fact has been used to good effect in matrix computations.

The Lanczos tridiagonalization is an algorithm that computes a factorization of a symmetric matrix C in the form $C = UTV^T$ where U and V are orthogonal and T is tridiagonal [46]. It is an important component in many other algorithms, such as the computation of the symmetric Schur decomposition [26]. The Golub-Kahan bidiagonalization [54] is a way of decomposing a matrix M into the product UBV^T where B is bidiagonal. It was an important observation that bidiagonalization of A is essentially equivalent to applying Lanczos bidiagonalization of $\mathbf{sym}(A)$. This proved crucial in the development of the Golub-Kahan SVD algorithm [25].

While not as groundbreaking, we will show in this chapter how a tensor equivalent of (1.42) can be used to show new connections between tensor eigenvalues and singular values similar to (2.2) and derive new algorithms.

Another way to connect A and $\mathbf{sym}(A)$ is through the *Rayleigh quotients*

$$\phi_A(u,v) \;=\; \frac{u^T A v}{\| u \|_2 \| v \|_2} \;=\; \left(\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} A(i_1,i_2) u(i_1) v(i_2) \right) \Big/ \left(\| u \|_2 \| v \|_2 \right) \quad (2.3)$$

and

$$\phi_A^{(sym)}(x) \;=\; \frac{1}{2} \frac{x^T C x}{x^T x} \;=\; \frac{1}{2} \left(\sum_{i_1=1}^{N} \sum_{i_2=1}^{N} C(i_1,i_2) x(i_1) x(i_2) \right) \Big/ \| x \|_2^2 \qquad (2.4)$$

where $u \in \mathbb{R}^{n_1}$, $v \in \mathbb{R}^{n_2}$, $N = n_1 + n_2$, $x \in \mathbb{R}^N$, and $C = \mathbf{sym}(A)$. If x is a stationary vector for $\phi_A^{(sym)}$, then $u = x(1{:}n_1)$ and $v = x(n_1+1{:}n_1+n_2)$ render a stationary value for ϕ_A. See [26, p.448]. Rayleigh quotients are important because they are closely related to eigenvalues and singular values. We will explore Rayleigh quotients in detail and introduce higher-order tensor generalizations in §2.3.

2.1.2 A Third Order Example

Before we proceed with the rest of this chapter, we use the case of third-order tensors to preview some of the main ideas and to establish notation. The starting point is to define the trilinear Rayleigh quotient

$$\phi_{\mathcal{A}}(u,v,w) \;=\; \left(\sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \sum_{i_3=1}^{n_3} \mathcal{A}(i_1,i_2,i_3) u(i_1) v(i_2) w(i_3) \right) \Big/ \left(\| u \|_2 \, \| v \|_2 \, \| w \|_2 \right) \qquad (2.5)$$

where $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}$,$u \in \mathbb{R}^{n_1}$, $v \in \mathbb{R}^{n_2}$, and $w \in \mathbb{R}^{n_3}$.

The singular values and vectors of $\mathcal{A}$ are the critical values and vectors of $\phi_{\mathcal{A}}$ as formulated in [48]. A simple expression for the gradient $\nabla \phi_{\mathcal{A}}$ is made possible by unfolding $\mathcal{A} = (a_{ijk})$ in each of its three modes and aggregating the u, v, and w vectors with the Kronecker product. It is necessary to specify the order in which the fibers appear in a modal unfolding. The choice exhibited in (1.24) has the

property that

$$\sum_{i_1=1}^{n_1}\sum_{i_2=1}^{n_2}\sum_{i_3=1}^{n_3}\mathcal{A}(i_1,i_2,i_3)u(i_1)v(i_2)w(i_3) \;=\; \begin{cases} u^T\,\mathcal{A}_{(1)}\;w\otimes v \\ v^T\,\mathcal{A}_{(2)}\;w\otimes u \\ w^T\mathcal{A}_{(3)}\;v\otimes u \end{cases} \tag{2.6}$$

which makes it easy to specify the stationary vectors of $\phi_{\mathcal{A}}$. If u, v, and w are unit vectors, then the gradient of $\phi_{\mathcal{A}}$ is given by

$$\nabla\phi_{\mathcal{A}}(u,v,w) \;=\; \begin{bmatrix} \mathcal{A}_{(1)}\;w\otimes v \\ \mathcal{A}_{(2)}\;w\otimes u \\ \mathcal{A}_{(3)}\;v\otimes u \end{bmatrix} \;-\; \phi_{\mathcal{A}}(u,v,w)\begin{bmatrix} u \\ v \\ w \end{bmatrix}. \tag{2.7}$$

We remark that if $\mathcal{A}$ is an order-2 tensor, then (2.7) collapses to the familiar matrix-SVD equations $Av=\sigma u$ and $A^Tu=\sigma v$.

A central contribution of this chapter revolves around the tensor version of the **sym** matrix (1.42) and the associated Rayleigh quotient $\phi_{\mathcal{A}}^{(sym)}$ that is defined in (2.4). Just as **sym**-of-a-matrix sets up a symmetric block matrix whose entries are either zero or matrix transpositions, **sym**-of-a-tensor sets up a symmetric block tensor whose entries are either zero or a tensor transposition.

The symmetric embedding of a 3rd-order tensor results in a 3-by-3-by-3 block tensor, a kind of Rubik's cube built from 27 (possibly non-cubical) boxes. If $\mathcal{A}\in\mathbb{R}^{n_1\times n_2\times n_3}$ and $N=n_1+n_2+n_3$, then $\mathbf{sym}(\mathcal{A}) \;=\; C\in\mathbb{R}^{N\times N\times N}$ is the 3-by-3-by-3 block tensor whose ijk block is specified by

$$\mathcal{C}_{[i\,j\,k]} \;=\; \begin{cases} \mathcal{A}^{<[i\,j\,k]>} & \text{if } [i\;j\;k] \text{ is a permutation of } [1\;2\;3] \\ \\ 0\in\mathbb{R}^{n_i\times n_j\times n_k} & \text{otherwise.} \end{cases} \tag{2.8}$$

See Figure 2.1. The blocks in a block tensor such as $\mathcal{C}$ can be specified using the colon notation. For example, if $n_1=4$, $n_2=3$ and $n_3=2$, then

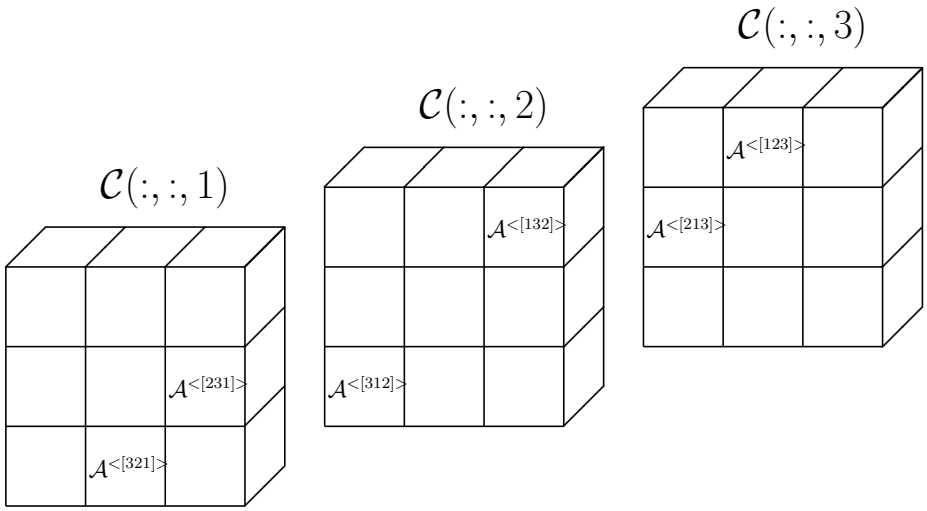


Figure 2.1: The Symmetric Embedding of an Order-3 Tensor

$$\begin{array}{rcccl}
\mathcal{C}_{[1\,2\,3]} & = & \mathcal{C}(1{:}4,5{:}7,8{:}9) & = & \mathcal{A}^{<[1\,2\,3]>} \in \mathbb{R}^{n_1\times n_2\times n_3} \\
\mathcal{C}_{[1\,3\,2]} & = & \mathcal{C}(1{:}4,8{:}9,5{:}7) & = & \mathcal{A}^{<[1\,3\,2]>} \in \mathbb{R}^{n_1\times n_3\times n_2} \\
\mathcal{C}_{[2\,1\,3]} & = & \mathcal{C}(5{:}7,1{:}4,8{:}9) & = & \mathcal{A}^{<[2\,1\,3]>} \in \mathbb{R}^{n_2\times n_1\times n_3} \\
\mathcal{C}_{[2\,3\,1]} & = & \mathcal{C}(5{:}7,8{:}9,1{:}4) & = & \mathcal{A}^{<[2\,3\,1]>} \in \mathbb{R}^{n_2\times n_3\times n_1} \\
\mathcal{C}_{[3\,1\,2]} & = & \mathcal{C}(8{:}9,1{:}4,5{:}7) & = & \mathcal{A}^{<[3\,1\,2]>} \in \mathbb{R}^{n_3\times n_1\times n_2} \\
\mathcal{C}_{[3\,2\,1]} & = & \mathcal{C}(8{:}9,5{:}7,1{:}4) & = & \mathcal{A}^{<[3\,2\,1]>} \in \mathbb{R}^{n_3\times n_2\times n_1}
\end{array} . \tag{2.9}$$

We will prove in §2.2.3 that the tensor $\mathcal{C}$ is in fact symmetric.

The last topic to cover in our order-3 preview is the generalization of the Rayleigh quotient $\phi_{\mathcal{A}}^{(sym)}$ defined in (2.5). If $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3}$, $\mathcal{C} = \mathbf{sym}(\mathcal{A})$, $N = n_1 + n_2 + n_3$, and $x \in \mathbb{R}^N$, then $\phi_{\mathcal{A}}^{(sym)}$ is defined by

$$\phi_{\mathcal{A}}^{(sym)}(x) \;=\; \frac{1}{3!}\left(\sum_{i_1=1}^{N}\sum_{i_2=1}^{N}\sum_{i_3=1}^{N}\mathcal{C}(i_1,i_2,i_3)x(i_1)x(i_2)x(i_3)\right)\Bigg/\,\|\,x\,\|_2^3 \tag{2.10}$$

It will be shown in §2.3.3 that if

$$x \;=\; \begin{bmatrix} u \\ v \\ w \end{bmatrix} \begin{matrix} \}n_1 \\ \}n_2 \\ \}n_3 \end{matrix}$$

satisfies $\nabla\phi_{\mathcal{A}}^{(sym)}(x) = 0$, then

$$\nabla_u \phi_{\mathcal{A}}(u,v,w) \;=\; 0 \qquad \nabla_v \phi_{\mathcal{A}}(u,v,w) \;=\; 0 \qquad \nabla_w \phi_{\mathcal{A}}(u,v,w) \;=\; 0$$

where ∇_z refers to the gradient with respect to the components in vector z. Moreover, it will be shown that

$$x_{+-} = \begin{bmatrix} u \\ v \\ -w \end{bmatrix} \qquad x_{-+} = \begin{bmatrix} u \\ -v \\ w \end{bmatrix} \qquad x_{--} = \begin{bmatrix} u \\ -v \\ -w \end{bmatrix}$$

are also stationary vectors for $\phi_{\mathcal{A}}^{(sym)}$ and

$$\phi_{\mathcal{A}}(u,v,w) \;=\; \phi_{\mathcal{A}}^{(sym)}(x) \;=\; \phi_{\mathcal{A}}^{(sym)}(x_{--}) \;=\; -\phi_{\mathcal{A}}^{(sym)}(x_{-+}) \;=\; -\phi_{\mathcal{A}}^{(sym)}(x_{+-}).$$

2.2 The General Case

Block matrix manipulation is such a fixture in numerical linear algebra that we take for granted the correctness of facts like

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}^T = \begin{bmatrix} A_{11}^T & A_{21}^T \\ A_{12}^T & A_{22}^T \end{bmatrix}. \tag{2.11}$$

Formal verification requires showing that the (i,j) entries on both sides of the equation are equal for all valid ij pairs.

The symmetric embedding of a tensor involves generalizations of both transposition and blocking so this section begins by discussing these notions and establishing the tensor analog of (2.11).

2.2.1 Blocking

The act of blocking an n_1-by-n_2 matrix C is the act defining the *blocking vectors* $\mathbf{m}^{(1)}$ and $\mathbf{m}^{(2)}$:

$$\mathbf{m}^{(1)} \;=\; \begin{bmatrix} m_1^{(1)} & \cdots & m_{b_1}^{(1)} \end{bmatrix}, \qquad \mathbf{m}^{(2)} \;=\; \begin{bmatrix} m_1^{(2)} & \cdots & m_{b_2}^{(2)} \end{bmatrix} \tag{2.12}$$

of positive integers that sum to n_1 and n_2, respectively.

Given (2.12), we are able to regard C as a $b_1 \times b_2$ block matrix (C_{i_1,i_2}) where block C_{i_1,i_2} has $m_{i_1}^{(1)}$ rows and $m_{i_2}^{(2)}$ columns. It is easy (although messy) to "locate" a particular entry of a particular block. Indeed,

$$C_{i_1,i_2}(j_1,j_2) \;=\; C(\,\rho_{i_1}^{(1)} + j_1\,,\; \rho_{i_2}^{(2)} + j_2\,)$$

where

$$\rho_{i_k}^{(k)} \;=\; m_1^{(k)} + m_2^{(k)} + \cdots + m_{i_k-1}^{(k)} \tag{2.13}$$

for $k = 1, 2$.

To block an order-d tensor $\mathcal{C} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ we proceed analogously. We define the blocking vectors

$$\mathbf{m}^{(k)} \;=\; \begin{bmatrix} m_1^{(k)} & \cdots & m_{b_k}^{(k)} \end{bmatrix} \qquad k = 1, 2, \ldots, d \tag{2.14}$$

and this permits us to regard $\mathcal{C}$ as a $b_1 \times \cdots \times b_d$ block tensor. If $\mathbf{i} = [i_1, \ldots, i_d]$, then the $\mathbf{i}$-th block is the subtensor

$$\mathcal{C}_{\mathbf{i}} \;=\; \mathcal{C}_{i_1,\ldots,i_d} \;=\; \mathcal{C}(\rho_{i_1}^{(1)}{+}1{:}\rho_{i_1}^{(1)}{+}m_{i_1}^{(1)}, \ldots, \rho_{i_d}^{(d)}{+}1{:}\rho_{i_d}^{(d)} + m_{i_d}^{(d)}).$$

If $\mathbf{j} = [j_1, \ldots, j_d]$, then the $\mathbf{j}$-th entry of this subtensor is given by

$$\mathcal{C}_{\mathbf{i}}(\mathbf{j}) \;=\; \mathcal{C}(\, \rho_{i_1}^{(1)} + j_1 \,, \ldots, \, \rho_{i_d}^{(d)} + j_d) \in \mathbb{R} \tag{2.15}$$

where $\rho_{i_k}^{(k)}$ is specified by (2.13) for $k = 1, 2, \ldots, d$.

To illustrate equations (2.13)-(2.15), if $\mathcal{C} \in \mathbb{R}^{9\times7\times5\times6}$ and

$$\begin{array}{rcll} m_1 &=& [\,4 \;\; 2 \;\; 3\,] & (b_1 = 3) \\ m_2 &=& [\,5 \;\; 2\,] & (b_2 = 2) \\ m_3 &=& [\,4 \;\; 1\,] & (b_3 = 2) \\ m_4 &=& [\,2 \;\; 2 \;\; 2\,] & (b_4 = 3) \end{array} \;,$$

then we are choosing to regard $\mathcal{C}$ as a $3\times2\times2\times3$ block tensor. Thus, if $\mathbf{i} = [3\ 1\ 2\ 1]$ then $\mathcal{C}_{\mathbf{i}} = \mathcal{C}(7{:}9, 1{:}5, 5{:}5, 1{:}2)$ and

$$\mathcal{C}_{\mathbf{i}}(\mathbf{j}) \;=\; \mathcal{C}(6 + j_1 \,,\; j_2 \,,\; 4 + j_3 \,,\; j_4)$$

where $\mathbf{1} \le \mathbf{j} \le [3\ 5\ 1\ 2]$.

Note that we will treat tensor blockings in more detail in Chapter 3.

2.2.2 Block Transposition

It is also easy to verify from the tensor transpose definition (1.3) that if $\mathbf{f}$ and $\mathbf{g}$ are both permutations of $1{:}d$, then

$$(\mathcal{A}^{<\mathbf{f}>})^{<\mathbf{g}>} \;=\; \mathcal{A}^{<\mathbf{f}(\mathbf{g})>}. \tag{2.16}$$

A transposition of a block tensor renders another block tensor. The following lemma makes this precise and generalizes (2.11).

Lemma 2.1. *Suppose* $\mathcal{C} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ *is a* $b_1 \times \cdots \times b_d$ *block tensor with block dimensions defined by the partitioning* (2.14). *Let* $\mathcal{C}_{\mathbf{i}}$ *denote its* $\mathbf{i}$*-th block where* $\mathbf{i} = [i_1, \ldots, i_d]$. *If* $\mathbf{p} = [\, p_1, \ldots, p_d \,]$ *is a permutation of* 1:d *and* $\mathcal{B} = \mathcal{C}^{<\mathbf{p}>}$, *then the tensor* $\mathcal{B} \in \mathbb{R}^{m_{p_1} \times \cdots \times m_{p_d}}$ *is a* $b_{p_1} \times \cdots \times b_{p_d}$ *block tensor where each block* $\mathcal{B}_{\mathbf{i}(\mathbf{p})}$ *is defined by* $\mathcal{B}_{\mathbf{i}(\mathbf{p})} = \mathcal{C}_{\mathbf{i}}^{<\mathbf{p}>}$.

Proof. If $1 \le j_k \le m_{i_k}^{(k)}$ for $k = 1, 2, \ldots, d$, then from (2.14) and (2.15) we have

$$\mathcal{C}_{\mathbf{i}}^{<\mathbf{p}>}(j_{p_1}, \ldots, j_{p_d}) \;=\; \mathcal{C}_{\mathbf{i}}(j_1, \ldots, j_p) \;=\; \mathcal{C}(\rho_{i_1}^{(1)} + j_1, \ldots, \rho_{i_d}^{(d)} + j_d)$$

On the other hand, $\mathcal{B} = \mathcal{C}^{<\mathbf{p}>}$ and so

$$\mathcal{C}(\rho_{i_1}^{(1)} + j_1, \ldots, \rho_{i_d}^{(d)} + j_d) \;=\; \mathcal{B}(\rho_{i_{p_1}}^{(p_1)} + j_{p_1}, \ldots, \rho_{i_{p_d}}^{(p_d)} + j_{p_d}) \;=\; \mathcal{B}_{\mathbf{i}(\mathbf{p})}(j_{p_1}, \ldots, j_{p_d}).$$

Thus, $\mathcal{B}_{\mathbf{i}(\mathbf{p})}(\mathbf{j}(\mathbf{p})) \;=\; \mathcal{C}_{\mathbf{i}}^{<\mathbf{p}>}(\mathbf{j}(\mathbf{p}))$ for all $\mathbf{j}$, i.e., $\mathcal{B}_{\mathbf{i}(\mathbf{p})} = \mathcal{C}_{\mathbf{i}}^{<\mathbf{p}>}$. □

2.2.3 The sym(·) Operation

The tensor analog of (1.42) involves constructing an order-d symmetric tensor $\mathbf{sym}(\mathcal{A})$ whose blocks are either zero or carefully chosen transposes of $\mathcal{A}$. In particular, if $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, then

$$\mathbf{sym}(\mathcal{A}) \in \mathbb{R}^{N \times \cdots \times N} \qquad N = n_1 + \cdots n_d$$

is a block tensor defined by the blocking vectors

$$\mathbf{m}^{(k)} = [\, n_1 \;\; n_2 \;\; \cdots \;\; n_d \,] \qquad k = 1, 2, \ldots, d. \tag{2.17}$$

The $\mathbf{i}$-th block of $\mathcal{C} = \mathbf{sym}(\mathcal{A})$ is given by

$$\mathcal{C}_{\mathbf{i}} \;=\; \begin{cases} A^{<\mathbf{i}>} & \text{if } \mathbf{i} \text{ is a permutation of } 1{:}d \\ 0 & \text{otherwise} \end{cases}$$

for all $\mathbf{i}$ that satisfy $\mathbf{1} \leq \mathbf{i} \leq \mathbf{d}$. Note that $\mathcal{C}_{\mathbf{i}}$ is $n_{i_1} \times n_{i_2} \times \cdots \times n_{i_d}$. We confirm that $\mathbf{sym}(\mathcal{A})$ is symmetric.

Lemma 2.2. *If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $\mathcal{C} = \mathbf{sym}(\mathcal{A})$, then $\mathcal{C}$ is symmetric.*

Proof. Let $\mathbf{p}$ be an arbitrary permutation of $1{:}d$. We must show that if $\mathcal{B} = C^{<\mathbf{p}>}$ then $\mathcal{B} = \mathcal{C}$. Since $\mathcal{C}$ as a block tensor is $d \times d \times \cdots \times d$, it follows from Lemma 2.1 that $\mathcal{B}$ has the same block structure and

$$\mathcal{B}_{\mathbf{i}(\mathbf{p})} \;=\; \mathcal{C}_{\mathbf{i}}^{<\mathbf{p}>}$$

for all $\mathbf{i}$ that satisfy $\mathbf{1} \leq \mathbf{i} \leq \mathbf{d}$. If $\mathbf{i}$ is a permutation of $1{:}d$, then $\mathcal{C}_{\mathbf{i}} = A^{<\mathbf{i}>}$ and by using (2.16) we conclude that

$$\mathcal{B}_{\mathbf{i}(\mathbf{p})} \;=\; (\mathcal{A}^{<\mathbf{i}>})^{<\mathbf{p}>} \;=\; \mathcal{A}^{<\mathbf{i}(\mathbf{p})>} \;=\; \mathcal{C}_{\mathbf{i}(\mathbf{p})}.$$

If $\mathbf{i}$ is *not* a permutation of $1{:}d$, then both $\mathcal{C}_{\mathbf{i}}$ and $\mathcal{C}_{\mathbf{i}(\mathbf{p})}$ are zero and so

$$\mathcal{B}_{\mathbf{i}(\mathbf{p})} = \mathcal{C}_{\mathbf{i}}^{<\mathbf{p}>} = 0 = \mathcal{C}_{\mathbf{i}(\mathbf{p})}.$$

Since $\mathcal{B}$ and $\mathcal{C}$ agree block-by-block, they are the same. □

Note that by using the Tensor Toolbox [39] with the block tensor extensions that are described in Chapter 3 and Appendix A, one can form the symmetric embedding of the `tensor` object `A` with the following commands.

```
n=size(A); d=length(n);
N = sum(n)*ones(1,d);
M = cell(d,1);
for i=1:d, M{i} = n; end
C = bltensor(zeros(N),M);
P = perms(1:d);
for i = 1:size(P,1)
  C(P(i,:)) = permute(A,P(i,:));
end
```

2.2.4 Notation and Facts

The summation that defines the multilinear Rayleigh quotient (2.5) can be written succinctly in matrix-vector terms.

Lemma 2.3. *If* $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ *and* $u_k \in \mathbb{R}^{n_k}$ *for* $k = 1, 2, \ldots, d$, *then*

$$\mathcal{A} \cdot (u_1, \ldots, u_d) \;=\; \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathcal{A}(\mathbf{i}) u_1(i_1) \cdots u_d(i_d) \;=\; \mathrm{vec}(\mathcal{A})^T u_d \otimes \cdots \otimes u_1. \tag{2.18}$$

Moreover, for $k = 1, 2, \ldots, d$ *we have*

$$\mathcal{A} \cdot (u_1, \ldots, u_d) \;=\; u_k^T \, \mathcal{A}_{(k)} \, \tilde{u}_k \tag{2.19}$$

where

$$\tilde{u}_k \;=\; (u_d \otimes \cdots \otimes u_{k+1} \otimes u_{k-1} \otimes \cdots \otimes u_1). \tag{2.20}$$

Proof. If $a = \mathrm{vec}(\mathcal{A})$ and $b = u_d \otimes \cdots \otimes u_1$, then using the definition of vec and equations (1.11) and (1.23), we have

$$\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathcal{A}(\mathbf{i}) u_1(i_1) \cdots u_d(i_d) \;=\; \sum_{\mathbf{i=1}}^{\mathbf{n}} a(\mathit{ivec}(\mathbf{i}, \mathbf{n})) \cdot b(\mathit{ivec}(\mathbf{i}, \mathbf{n})) \;=\; \sum_{\mathbf{i=1}}^{\mathbf{n}} a(\mathbf{i}) b(\mathbf{i}) \;=\; a^T b.$$

This proves (2.19). Using the modal subtensor interpretation of $\mathcal{A}_{(k)}$ that are discussed in §1.2.3 and definitions (1.21) and (1.22), we have

$$\begin{aligned}\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathcal{A}(\mathbf{i}) u_1(i_1)\cdots u_d(i_d) &= \sum_{i_k=1}^{n_k} u_k(i_k) \left(\sum_{\tilde{\mathbf{i}}=\mathbf{1}}^{\tilde{\mathbf{n}}} \mathcal{B}^{(i_k)}(\tilde{\mathbf{i}}) \tilde{u}(\tilde{\mathbf{i}}) \right) \\ &= \sum_{i_k=1}^{n_k} u_k(i_k) \left(\mathcal{A}_{(k)}(i_k,:) \tilde{u}_k \right) \;=\; u_k^T \mathcal{A}_{(k)} \tilde{u}_k\end{aligned}$$

which establishes (2.18). □

Notation: We will alternate between using the multilinear product notation, e.g. $\mathcal{A} \cdot (u_1, \ldots, u_d)$, and the equivalent modal unfolding notation, e.g. $u_k^T \mathcal{A}_{(k)} \tilde{u}_k$, as we deem appropriate in terms of clarity and conciseness.

Summations that involve symmetric tensors are important in later sections. The following notation for the multiple Kronecker product of a single vector is handy:

$$x^{\otimes d} \;=\; \underbrace{x \;\otimes\; \cdots \;\otimes\; x}_{d \text{ times}}.$$

Note that if $x \in \mathbb{R}^N$, then $x^{\otimes d} \in \mathbb{R}^{N^d}$.

Lemma 2.4. *If $\mathcal{C} \in \mathbb{R}^{N\times\cdots\times N}$ is a symmetric order-d tensor and $x \in \mathbb{R}^N$, then*

$$\mathcal{C} \cdot (x, \ldots, x) \;\equiv\; \sum_{\mathbf{i=1}}^{\mathbf{N}} \mathcal{C}(\mathbf{i}) x(i_1)\cdots x(i_d) \;=\; x^T \mathcal{C}_{(1)}\, x^{\otimes(d-1)} \tag{2.21}$$

Proof. This follows from Lemma 2.3 by setting $n_k = N$ and $u_k = x$ for all $k = 1, 2, \ldots, d$. Note that because $\mathcal{C}$ is symmetric, $\mathcal{C}_{(1)} = \cdots = \mathcal{C}_{(d)}$. □

The summation (2.21) has a special characterization if $\mathcal{C} = \mathbf{sym}(\mathcal{A})$. To pursue this we will have to navigate $\mathcal{C}$'s block structure and to that end we define the index

vectors $\mathbf{L}$ and $\mathbf{R}$ as follows:

$$\mathbf{L} = \begin{bmatrix} 1 \\ n_1+1 \\ \vdots \\ n_1+\cdots+n_{d-1}+1 \end{bmatrix} \qquad \mathbf{R} = \begin{bmatrix} n_1 \\ n_1+n_2 \\ \vdots \\ n_1+\cdots+n_d \end{bmatrix}. \tag{2.22}$$

Note that if $\mathbf{1} \le \mathbf{p} \le \mathbf{d}$, then

$$\mathcal{C}_{\mathbf{p}} = \mathcal{C}(\mathbf{L}(p_1):\mathbf{R}(p_1), \ldots, \mathbf{L}(p_d):\mathbf{R}(p_d))$$

is $\mathcal{C}$'s $\mathbf{p}$-th block.

Lemma 2.5. *Suppose $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, $\mathcal{C} = \mathbf{sym}(\mathcal{A})$, and $N = n_1 + \cdots + n_d$. If $x \in \mathbb{R}^N$ is partitioned as follows*

$$x = \begin{bmatrix} u_1 \\ \vdots \\ u_d \end{bmatrix} \qquad u_k \in \mathbb{R}^{n_k},$$

and $\tilde{u}_1, \ldots, \tilde{u}_d$ are defined by (2.20), then

$$\mathcal{C} \cdot (I_N, x, \ldots, x) = \mathcal{C}_{(1)} x^{\otimes(d-1)} = (d-1)! \begin{bmatrix} \mathcal{A}_{(1)}\tilde{u}_1 \\ \vdots \\ \mathcal{A}_{(d)}\tilde{u}_d \end{bmatrix} \tag{2.23}$$

and

$$\sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{N}} \mathcal{C}(\mathbf{j}) x(j_1)\cdots x(j_d) = d! \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \mathcal{A}(\mathbf{i}) u_1(i_1)\cdots u_d(i_d) \tag{2.24}$$

i.e.

$$\mathcal{C} \cdot (x, \ldots, x) = d!\, \mathcal{A} \cdot (u_1, \ldots, u_d). \tag{2.25}$$

Proof. If $v = \mathcal{C}_{(1)} x^{\otimes(d-1)}$ and

$$e_j = I_N(:, j) = \begin{bmatrix} w_1 \\ \vdots \\ w_d \end{bmatrix}, \qquad w_i \in \mathbb{R}^{n_i}, \tag{2.26}$$

is partitioned conformally with x, then for $j = 1, 2, \ldots, N$ we have

$$\begin{aligned} v(j) &= \sum_{\mathbf{i}(2:d)=\mathbf{1}}^{\mathbf{N}} \mathcal{C}(j, i_2, \ldots, i_d) x(i_2) \cdots x(i_d) \\ &= \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{N}} \mathcal{C}(\mathbf{i}) e_j(i_1) x(i_2) \cdots x(i_d) \\ &= \sum_{\mathbf{p}=\mathbf{1}}^{\mathbf{d}} \sum_{\mathbf{i}=\mathbf{L}(\mathbf{p})}^{\mathbf{R}(\mathbf{p})} \mathcal{C}(\mathbf{i}) e_j(i_1) x(i_2) \cdots x(i_d) \\ &= \sum_{\mathbf{p}=\mathbf{1}}^{\mathbf{d}} \left(\sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{n}(\mathbf{p})} \mathcal{C}_{\mathbf{p}}(\mathbf{k}) w_{p_1}(k_1) u_{p_2}(k_2) \cdots u_{p_d}(k_d) \right) \end{aligned}$$

Now suppose that $\mathbf{L}(q) \le j \le \mathbf{R}(q)$, $j = \mathbf{L}(q) + r - 1$. From (2.23) we must show that v_j is the rth component of $\mathcal{A}_{(q)} \tilde{u}_q$.

To that end observe that $\mathcal{C}_{\mathbf{p}}(\mathbf{k}) w_{p_1}(k_1)$ is necessarily zero unless $p_1 = q$, $k_1 = r$, and $\mathbf{p}$ is a permutation of $1{:}d$. Assuming this to be the case and defining the vectors $v_1, \ldots, v_d$ by

$$v_i = \begin{cases} u_i & \text{if } i \neq q \\ w_q & \text{otherwise} \end{cases},$$

we see using (2.19) that

$$\begin{aligned} \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{n}(\mathbf{p})} \mathcal{C}_{\mathbf{p}}(\mathbf{k}) w_{p_1}(k_1) u_{p_2}(k_2) \cdots u_{p_d}(k_d) &= \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{n}(\mathbf{p})} \mathcal{A}^{<p>}(\mathbf{k}) v_{p_1}(k_1) v_{p_2}(k_2) \cdots v_{p_d}(k_d) \\ &= \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{n}} \mathcal{A}(\mathbf{k}) v_1(k_1) v_2(k_2) \cdots v_d(k_d) \\ &= v_q^T \mathcal{A}_{(q)} v_d \otimes \cdots \otimes v_{q+1} \otimes v_{q-1} \otimes \cdots \otimes v_1 \\ &= w_q^T \mathcal{A}_{(q)} u_d \otimes \cdots \otimes u_{q+1} \otimes u_{q-1} \otimes \cdots \otimes u_1 \\ &= w_q^T \mathcal{A}_{(q)} \tilde{u}_q. \end{aligned}$$

Observe that the number of $\mathbf{p}$ that satisfy $\mathbf{1} \leq \mathbf{p} \leq \mathbf{d}$ subject to the constraint $p_1 = q$ is $(d-1)!$ and conclude from (2.26) that $w_q = I_{n_q}(:, r)$. It follows that

$$v(j) \;=\; \sum_{\mathbf{p}\,=\,\mathbf{1}}^{\mathbf{d}} w_q^T \mathcal{A}_{(q)} \tilde{u}_q \;=\; (d-1)! \left[\mathcal{A}_{(q)} \tilde{u}_q\right]_r .$$

This establishes (2.23). Equation (2.24) follows from

$$x^T \mathcal{C}_{(1)} x^{\otimes(d-1)} \;=\; \sum_{k=1}^{d} (d-1)! u_k^T \mathcal{A}_{(k)} \tilde{u}_k$$

and Lemmas 2.3 and 2.4. □

2.2.5 A Note on Uniqueness

For any tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ the function

$$f_{\mathcal{A}}(x) \;=\; f_{\mathcal{A}}(u_1, \ldots, u_d) \;=\; \mathcal{A} \cdot (u_1, \ldots, u_d) \;=\; \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \mathcal{A}(\mathbf{i}) u_1(i_1) \cdots u_d(i_d) \quad (2.27)$$

is a homogeneous polynomial of degree d in the coefficients of $x = [u_1^T \;\cdots\; u_d^T]^T \in \mathbb{R}^N$ where $N = n_1 + \cdots + n_d$. By the identification of homogeneous polynomials and symmetric tensors [14] there exists a unique symmetric tensor $\mathcal{B} \in \mathbb{R}^{N \times \cdots \times N}$ such that

$$\mathcal{B} \cdot (x, \ldots, x) \;=\; f_{\mathcal{A}}(x)$$

for all vectors $u_i \in \mathbb{R}^{n_i}$, $i = 1, \ldots, d$. From Lemma 2.5 that know that

$$\mathcal{B} \;=\; \frac{\mathbf{sym}(\mathcal{A})}{d!}$$

satisfies this relation. Indeed, we could have derived **sym** from the *polar form* of $f_{\mathcal{A}}$, which is the polynomial

$$F_{\mathcal{A}}(x^{(1)}, \ldots, x^{(d)})$$

where $x^{(i)} \in \mathbb{R}^N$, $F_{\mathcal{A}}$ is linear separately in each $x^{(i)}$, symmetric in the $x^{(i)}$ and satisfies

$$F_{\mathcal{A}}(x, \ldots, x) \;=\; f_{\mathcal{A}}(x).$$

In other words,

$$F_{\mathcal{A}}(x^{(1)}, \ldots, x^{(d)}) \;=\; \mathcal{B} \cdot (x^{(1)}, \ldots, x^{(d)}).$$

The polar form of $f_{\mathcal{A}}$ is given [76, 59] by the construction

$$F_{\mathcal{A}}(x^{(1)}, \ldots, x^{(d)}) \;=\; \frac{1}{d!} \frac{\partial}{\partial \lambda_1} \cdots \frac{\partial}{\partial \lambda_d} f_{\mathcal{A}}(\lambda_1 x^{(1)} + \cdots \lambda_d x^{(d)}) \Big|_{\boldsymbol{\lambda}=\mathbf{0}}.$$

Combining this with our previous results, we get the following theorem.

Theorem 2.6. *If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $x^{(1)}, \ldots, x^{(d)} \in \mathbb{R}^N$ then*

$$\mathbf{sym}(\mathcal{A}) \cdot (x^{(1)}, \ldots, x^{(d)}) \;=\; \frac{\partial}{\partial \lambda_1} \cdots \frac{\partial}{\partial \lambda_d} f_{\mathcal{A}}(\lambda_1 x^{(1)} + \cdots \lambda_d x^{(d)}) \Big|_{\boldsymbol{\lambda}=\mathbf{0}}. \tag{2.28}$$

Moreover, $\mathbf{sym}(\mathcal{A})/d!$ is the unique N-dimensional order-d symmetric tensor that satisfies (2.25).

Informally we might say that "$\mathbf{sym}(\mathcal{A})/d!$ is the polarization of $\mathcal{A}$."

Example: Let A be a 2×2 matrix, which means $d = 2$ and $n_1 = n_2 = 2$. Then if $u, v \in \mathbb{R}^2$ and $x = [u^T \; v^T]^T$ we have

$$\begin{aligned} f_A(x) \;&=\; u^T A v \;=\; (u_1 \;\; u_2) \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} \\ &=\; a_{11} u_1 v_1 + a_{12} u_1 v_2 + a_{21} u_2 v_1 + a_{22} u_2 v_2 \\ &=\; a_{11} x_1 x_3 + a_{12} x_1 x_4 + a_{21} x_2 x_3 + a_{22} x_2 x_4 \end{aligned}$$

and thus the polarization F_A is given by

$$F_A(x^{(1)}, x^{(2)}) \;=\; \frac{1}{2!} \frac{\partial}{\partial \lambda_1} \frac{\partial}{\partial \lambda_2} f_A\left(\lambda_1 x^{(1)} + \lambda_2 x^{(2)}\right) \Big|_{\boldsymbol{\lambda}=\mathbf{0}}$$

and

$$\begin{aligned} f_A(\lambda_1 x^{(1)} + \lambda_2 x^{(2)}) \;=\; & a_{11}(\lambda_1 x_1^{(1)} + \lambda_2 x_1^{(2)})(\lambda_1 x_3^{(1)} + \lambda_2 x_3^{(2)}) + \cdots \\ & a_{12}(\lambda_1 x_1^{(1)} + \lambda_2 x_1^{(2)})(\lambda_1 x_4^{(1)} + \lambda_2 x_4^{(2)}) + a_{21}(\lambda_1 x_2^{(1)} + \lambda_2 x_2^{(2)})(\lambda_1 x_3^{(1)} + \lambda_2 x_3^{(2)}) \\ & + a_{22}(\lambda_1 x_2^{(1)} + \lambda_2 x_2^{(2)})(\lambda_1 x_4^{(1)} + \lambda_2 x_4^{(2)}). \end{aligned}$$

After performing the differentiation we get

$$\begin{aligned} F_A(x^{(1)}, x^{(2)}) \;=\; & \frac{1}{2}\Big[a_{11}(x_1^{(1)} x_3^{(2)} + x_3^{(1)} x_1^{(2)}) + a_{12}(x_1^{(1)} x_4^{(2)} + x_4^{(1)} x_1^{(2)}) + \cdots \\ & a_{21}(x_2^{(1)} x_3^{(2)} + x_3^{(1)} x_2^{(2)}) + a_{22}(x_2^{(1)} x_4^{(2)} + x_4^{(1)} x_2^{(2)})\Big] \\ =\; & \frac{1}{2}(x^{(1)})^T \begin{pmatrix} 0 & 0 & a_{11} & a_{12} \\ 0 & 0 & a_{21} & a_{22} \\ a_{11} & a_{21} & 0 & 0 \\ a_{12} & a_{22} & 0 & 0 \end{pmatrix} x^{(2)} \\ =\; & \frac{1}{2}(x^{(1)})^T \mathbf{sym}(A) x^{(2)} \end{aligned}$$

as stated by Theorem 2.6.

2.3 Rayleigh Quotients and Stationary Values

In matrix computations, the *Rayleigh quotient* of a symmetric matrix $A \in \mathbb{R}^{n\times n}$ and a vector $x \in \mathbb{R}^n$ is the function

$$\phi_A(x) \;=\; \frac{x^T A x}{\| x \|_2^2}.$$

This quotient has many important properties [26], such as

$$\lambda_{\min}(A) \;\leq\; \phi_A(x) \leq \lambda_{\max}(A)$$

where ϕ_A achieves the upper and lower bounds when x is the eigenvector of A corresponding to the largest and smallest eigenvalue of A, respectively. Indeed, the eigenvalues and eigenvectors of A are the critical values and critical points of ϕ_A. Furthermore, setting $\lambda = \phi_A(x)$ minimizes $\| (A - \lambda I_n)x \|_2$, so if x is an approximate eigenvector then $\phi_A(x)$ is a reasonable choice for the corresponding eigenvalue.

Generalizing the Rayleigh quotient to a not-necessarily-symmetric n_1-by-n_2 matrix A and vectors $x \in \mathbb{R}^{n_1}, y \in \mathbb{R}^{n_2}$

$$\phi_A(x, y) \;=\; \frac{x^T A y}{\| x \|_2 \| y \|_2}.$$

It can be shown that the singular vectors of A are the critical points of $\phi_A(x, y)$ and

$$\sigma_{\min}(A) \;\leq\; |\phi_A(x, y)| \;\leq\; \sigma_{\max}(A).$$

Indeed, $\sigma_{\min}(A) = \phi_A(u_1, v_1), \sigma_{\max}(A) = \phi_A(u_k, v_k)$ where $k = \min(n_1, n_2)$ and $U\Sigma V^T$ is the SVD of A.

In light of these many important connections between Rayleigh quotients and eigenpairs and singular values and singular vectors for matrices it is reasonable to try to extend Rayleigh quotients to higher-order tensors and see if a similar situation involving tensor equivalents of eigenvalues and eigenvectors arises. This is the approach taken in [48].

Suppose $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $u_k \in \mathbb{R}^{n_k}$ for $k = 1, 2, \ldots, d$. Analogous to (2.3) we define the multilinear Rayleigh Quotient

$$\begin{aligned} \phi_{\mathcal{A}}(u_1, \ldots, u_d) \;&=\; \left(\sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \mathcal{A}(\mathbf{i}) u_1(i_1) \cdots u_d(i_d) \right) \Big/ \left(\| u_1 \|_2 \cdots \| u_d \|_2 \right) \qquad (2.29) \\ &=\; \frac{\mathcal{A} \cdot (u_1, \ldots, u_d)}{\| u_1 \|_2 \cdots \| u_d \|_2} \end{aligned}$$

If $\mathcal{C} = \mathbf{sym}(\mathcal{A})$, $N = n_1 + \cdots n_d$, and $x \in \mathbb{R}^N$, then corresponding to (2.4) we have

$$\begin{aligned} \phi_{\mathcal{A}}^{(sym)}(x) &= \frac{1}{d!}\left(\sum_{\mathbf{i=1}}^{\mathbf{N}} \mathcal{C}(\mathbf{i}) x(i_1)\cdots x(i_d)\right) \Big/ (\| x \|_2)^d \\ &= \frac{\mathcal{C}\cdot(x,\ldots,x)}{d!\,\| x \|_2^d} \end{aligned} \tag{2.30}$$

In this section we examine these multilinear Rayleigh quotients, specify their gradients, and relate the singular values of $\mathcal{A}$ to the eigenvalues of $\mathbf{sym}(\mathcal{A})$.

2.3.1 The Singular Values of a General Tensor

The gradient of $\phi_{\mathcal{A}}(u_1,\ldots,u_d)$ relates to a collection of matrix-vector products that involve the modal unfoldings of $\mathcal{A}$ and Kronecker products of the u-vectors.

Theorem 2.7. *If $\mathcal{A} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ and for $k = 1,\ldots,d$ the vectors $u_k \in \mathbb{R}^{n_k}$ each have unit 2-norm, then*

$$\nabla\phi_{\mathcal{A}}(u_1,\ldots,u_d) = \begin{bmatrix} \mathcal{A}_{(1)}\tilde{u}_1 \\ \vdots \\ \mathcal{A}_{(d)}\tilde{u}_d \end{bmatrix} - \phi_{\mathcal{A}}(u_1,\ldots,u_d)\cdot\begin{bmatrix} u_1 \\ \vdots \\ u_d \end{bmatrix}$$

where $\tilde{u}_k = (u_d \otimes \cdots \otimes u_{k+1} \otimes u_{k-1} \otimes \cdots \otimes u_1)$.

Proof. From Lemma 2.3 we have

$$\phi_{\mathcal{A}}(u_1,\ldots,u_d) = \left(u_k^T\,\mathcal{A}_{(k)}\,\tilde{u}_k\right) / \left(\| u_1 \|_2 \cdots \| u_d \|_2\right).$$

For $k = 1,2,\ldots,d$ we have

$$\nabla_{u_k}\left(u_k^T\,\mathcal{A}_{(k)}\,\tilde{u}_k\right) = \mathcal{A}_{(k)}\,\tilde{u}_k \quad \text{and} \quad \nabla_{u_k}(\| u_1 \|_2 \cdots \| u_d \|_2) = u_k.$$

and so

$$\begin{aligned}\nabla_{u_k}\phi_{\mathcal{A}} &= \frac{(\| u_1 \|_2 \cdots \| u_d \|_2)\mathcal{A}_{(k)}\tilde{u}_k \;-\; (u_k^T\mathcal{A}_{(k)}\tilde{u}_k)u_k}{(\| u_1 \|_2 \cdots \| u_d \|_2)^2} \\ &= \mathcal{A}_{(k)}\tilde{u}_k \;-\; \phi_{\mathcal{A}}(u_1,\ldots,u_d)u_k.\end{aligned}$$

The theorem follows by simply "stacking" these subvectors of the gradient. □

The variational approach to tensor singular values and vectors set forth in [48] is based on equating the gradient of $\phi_{\mathcal{A}}$ to zero.

Definition 2.8. *The scalar $\sigma \in \mathbb{R}$ is a* Z-singular value *of a general tensor $\mathcal{A} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ if there are unit vectors $u_k \in \mathbb{R}^{n_k}$ such that*

$$\mathcal{A}_{(k)}\tilde{u}_k = \sigma u_k, \tag{2.31}$$

for $k = 1,\ldots,d$. The vector u_k is the mode-k Z-singular vector *associated with σ.*

The normalization condition $u_k^T u_k = 1$ is necessary, since if $v_k = au_k$ for all $k = 1,2,\ldots,d$ then $\mathcal{A}_{(k)}\tilde{v}_k = a^{d-1}\mathcal{A}_{(k)}\tilde{u}_k = a^{k-1}\sigma u_k = (a^{k-2}\sigma)v_k$ for any $a \in \mathbb{R}$. It can be shown that at least one Z-singular value and associated Z-singular vectors exist for any tensor (cf. [48]). Note that equation (2.31) can also be written using the multilinear product in the form

$$\mathcal{A}\cdot(u_1,\ldots,u_{k-1},I_{n_k},u_{k+1},\ldots,u_d) \;=\; \sigma u_k.$$

2.3.2 The Eigenvalues of a Symmetric Tensor

For a symmetric tensor $\mathcal{C}$, the stationary values of $\phi_{\mathcal{C}}(x,\ldots,x)$ define the notion of a tensor eigenvalue.

Theorem 2.9. *If $\mathcal{C} \in \mathbb{R}^{N\times\cdots\times N}$ is symmetric and $x \in \mathbb{R}^N$ has unit norm, then*

$$\nabla_x \phi_{\mathcal{C}}(x,\ldots,x) \;=\; d\left(\mathcal{C}_{(1)}x^{\otimes(d-1)} \;-\; \phi_{\mathcal{C}}(x,\ldots,x)x\right).$$

Proof. From Lemma 2.4 we have

$$\phi_{\mathcal{C}}(x,\ldots,x) \;=\; x^T\mathcal{C}_{(1)}x^{\otimes(d-1)}/\parallel x \parallel^d.$$

Since

$$\nabla_x x^T\mathcal{C}_{(1)}x^{\otimes(d-1)} \;=\; d\mathcal{C}_{(1)}x^{\otimes(d-1)}$$

and

$$\nabla_x (x^Tx)^{d/2} \;=\; d(x^Tx)^{d/2-1}x$$

it follows that

$$\begin{aligned}\nabla_x \phi_{\mathcal{C}}(x,\ldots,x) \;&=\; d\,\frac{(x^Tx)^{d/2}\mathcal{C}_{(1)}x^{\otimes(d-1)} \;-\; \left(x^T\mathcal{C}_{(1)}x^{\otimes(d-1)}\right)(x^Tx)^{d/2-1}x}{(x^Tx)^d}\\ &=\; d\left(\mathcal{C}_{(1)}x^{\otimes(d-1)} \;-\; \left(x^T\mathcal{C}_{(k)}x^{\otimes(d-1)}\right)x\right)\\ &=\; d\left(\mathcal{C}_{(1)}x^{\otimes(d-1)} \;-\; \phi_{\mathcal{C}}(x,\ldots,x)x\right)\end{aligned}$$

completing the proof of the theorem. □

By setting the gradient of $\phi_{\mathcal{C}}(x,\ldots,x)$ to zero we arrive at the notion of a tensor eigenvalue [60].

Definition 2.10. *If $\mathcal{C} \in \mathbb{R}^{N\times\cdots\times N}$ is symmetric and $x \in \mathbb{R}^N$ is a unit vector such that*

$$\mathcal{C}_{(1)}x^{\otimes(d-1)} \;=\; \lambda x \tag{2.32}$$

then $\lambda = \phi_{\mathcal{C}}(x,\ldots,x)$ is a Z-eigenvalue *of $\mathcal{C}$ and x the associated* Z-eigenvector.

Note that if $\mathcal{C}_{(1)}x^{\otimes(d-1)} = \lambda x$ and $\alpha \in \mathbb{R}$, then $\mathcal{C}_{(1)}(\alpha x)^{\otimes(d-1)} = (\alpha^{d-2}\lambda)(\alpha x)$. Thus, we resolve a uniqueness issue by requiring Z-eigenvectors to have unit length, something that is not necessary in the matrix ($d = 2$) case. The equation (2.32) can also be written as

$$\mathcal{C} \cdot (I_N, x, \ldots, x) \;=\; \lambda x.$$

In [62, 60] it is shown that Z-eigenvalues and associated Z-eigenvectors always exist for symmetric tensors. Recently it has been shown that a symmetric tensor has at most $((d-1)^N - 1)/(d-2)$ complex Z-eigenvalues (known as *E-eigenvalues*), counted with multiplicity [5].

In [48] Lim refers to Z-singular values and Z-eigenvalues as l^2-singular values and l^2-eigenvalues, respectively, because of their connections to the 2-norm Rayleigh quotients (2.29) and (2.30).

2.3.3 The Eigenvalues of sym($\mathcal{A}$)

Since $\mathcal{C} = \mathbf{sym}(\mathcal{A})$ is so structured, we anticipate that the eigenvalue-defining equation $\nabla\phi_{\mathcal{A}}^{(sym)}(x) = 0$ will have some special features. From the Definitions 2.8 and 2.10 and Theorem 2.9, we have

$$\nabla\phi_{\mathcal{A}}^{(sym)}(x) \;=\; \frac{1}{d!}\nabla\phi_{\mathcal{C}}(x, \ldots, x) \;=\; \frac{1}{(d-1)!}\left(\mathcal{C}_{(1)}x^{\otimes(d-1)} \;-\; \phi_{\mathcal{C}}(x, \ldots, x)x\right) \tag{2.33}$$

We first characterize the gradient of $\phi_{\mathcal{A}}^{(sym)}$ in terms of matrix-vector products that involve $\mathcal{A}$'s modal unfoldings.

Theorem 2.11. *If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and x has unit 2-norm, then*

$$\nabla_x \phi_{\mathcal{A}}^{(sym)}(x) \;=\; \begin{bmatrix} \mathcal{A}_{(1)}\tilde{u}_1 \\ \vdots \\ \mathcal{A}_{(d)}\tilde{u}_d \end{bmatrix} \;-\; d \begin{bmatrix} (u_1^T \mathcal{A}_{(1)}\tilde{u}_1)u_1 \\ \vdots \\ (u_d^T \mathcal{A}_{(d)}\tilde{u}_d)u_d \end{bmatrix}. \tag{2.34}$$

Proof. From Lemmas 2.3 and 2.5 and the (2.29) and (2.30) we have

$$\begin{aligned} \phi_{\mathcal{A}}^{(sym)}(x) \;&=\; \frac{1}{d!}\left(\sum_{\mathbf{i=1}}^{\mathbf{N}} \mathcal{C}(\mathbf{i})x(i_1)\cdots x(i_d)\right) \Big/ (\parallel x \parallel_2)^d \\ &=\; \left(\sum_{\mathbf{i=1}}^{\mathbf{n}} \mathcal{A}(\mathbf{i})u_1(i_1)\cdots u_(i_d)\right) \Big/ (\parallel x \parallel_2)^d \\ &=\; \frac{u_k^T \mathcal{A}_{(k)}\tilde{u}_k}{(u_1^T u_1 + \cdots + u_d^T u_d)^{d/2}} \end{aligned}$$

for $k = 1, 2, \ldots, d$. Since

$$\nabla_{u_k}(u_1^T u_1 \;+\; \cdots + \; u_d^T u_d)^{d/2} \;=\; d \cdot (u_1^T u_1 \;+\; \cdots + \; u_d^T u_d)^{d/2-1} u_k \;=\; d \cdot u_k.$$

Since $x^T x = u_1^T u_1 + \cdots u_d^T u_d$, we can conclude that

$$\begin{aligned} \nabla_{u_k} \phi_{\mathcal{A}}^{(sym)}(x) \;&=\; \frac{(x^T x)^{d/2}\mathcal{A}_{(k)}\tilde{u}_k - d \cdot (u_k^T \mathcal{A}_{(k)}\tilde{u}_k)(x^T x)^{d/2-1}u_k}{(x^T x)^d} \\ &=\; \mathcal{A}_{(k)}\tilde{u}_k \;-\; d \cdot (u_k^T \mathcal{A}_{(k)}\tilde{u}_k)u_k \end{aligned}$$

completing the proof of the theorem. □

It turns out that if the gradient of $\phi_{\mathcal{A}}^{(sym)}(x)$ is zero, then the vector x generally has the property that each subvector u_k has the same norm.

Corollary 2.12. *If $\nabla \phi_{\mathcal{A}}^{(sym)}(x) = 0$ and $x^T x = 1$, then either $\mathcal{A}_{(k)}\tilde{u}_k = 0$ for $k = 1, 2, \ldots, d$ or*

$$\begin{bmatrix} \mathcal{A}_{(1)}\tilde{u}_1 \\ \vdots \\ \mathcal{A}_{(d)}\tilde{u}_d \end{bmatrix} \;=\; d \cdot \phi_{\mathcal{A}}^{(sym)}(x) \begin{bmatrix} u_1 \\ \vdots \\ u_d \end{bmatrix}$$

and $\parallel u_1 \parallel_2 = \parallel u_2 \parallel_2 = \cdots = \parallel u_d \parallel_2 = 1/\sqrt{d}$.

Proof. Since $\nabla\phi_{\mathcal{A}}^{(sym)}(x) = 0$, we know from Theorem 2.11 that

$$\mathcal{A}_{(k)}\tilde{u}_k \;=\; d \cdot (u_k^T \mathcal{A}_{(k)}\tilde{u}_k)u_k$$

for $k = 1, 2, \ldots, d$. Thus,

$$u_k^T \mathcal{A}_{(k)}\tilde{u}_k \;=\; d \cdot (u_k^T \mathcal{A}_{(k)}\tilde{u}_k)(u_k^T u_k).$$

From Lemma 2.3, if $u_k^T \mathcal{A}_{(k)}\tilde{u}_k = 0$ for some k, then it is zero for all k. In this case we conclude from (2.34) that $\mathcal{A}_{(k)}\tilde{u}_k = 0$ for $k = 1, 2, \ldots, d$. Otherwise, $1 = du_k^T u_k$, $k = 1, 2, \ldots, d$. It follows that $\| u_1 \|_2 \;=\; \cdots \;=\; \| u_d \|_2 \;=\; 1/\sqrt{d}$. □

We are now ready for the main result that relates the Z-eigenvalues and vectors of $\mathbf{sym}(\mathcal{A})$ to the Z-singular values and vectors of $\mathcal{A}$.

Theorem 2.13. *If σ is a nonzero Z-singular value of $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ with unit modal Z-singular vectors $u_1, \ldots, u_d$, then*

$$x_\alpha \;=\; \frac{1}{\sqrt{d}} \begin{bmatrix} \alpha_1 u_1 \\ \alpha_2 u_2 \\ \vdots \\ \alpha_d u_d \end{bmatrix} \qquad \alpha \;=\; [1, \pm 1, \ldots, \pm 1]$$

is a Z-eigenvector for $\mathbf{sym}(\mathcal{A})$ corresponding to Z-eigenvalue

$$\lambda_\alpha \;=\; \alpha_1\alpha_2 \cdots \alpha_d \, \frac{d!}{\sqrt{d^d}}\sigma.$$

Note that α_1 is set to +1 to resolve a uniqueness issue. See discussion after definition 2.10 and also equation (2.2) *for the matrix case.*

Proof. We must show that $g \;=\; \nabla\phi_{\mathcal{A}}^{(sym)}(x_\alpha) \;=\; 0$. If $\tau \;=\; \alpha_1 \cdots \alpha_d$ then for all $k = 1, 2, \ldots, d$ we have from (2.34) that

$$g_k \;=\; \frac{\tau}{\alpha_k d^{(d-1)/2}}\mathcal{A}_k\tilde{u}_k \;-\; d\frac{\alpha_k\tau}{d^{(d+1)/2}} \left(u_k^T \mathcal{A}_{(k)}\tilde{u}_k\right) u_k$$

But since $\sigma = u_k^T \mathcal{A}_k \tilde{u}_k$ and $\mathcal{A}_k \tilde{u}_k = \sigma u_k$, we have

$$g_k \;=\; \alpha_k \tau \left(\frac{1}{d^{(d-1)/2}} \;-\; \frac{1}{d^{(d-1)/2}} \right) u_k \;=\; 0.$$

Since $\lambda_\alpha \;=\; x_\alpha^T \mathcal{C}_{(1)} x_\alpha^{\otimes(d-1)}$ we have from Lemma 2.5 that

$$\lambda_\alpha \;=\; \left(\frac{1}{\sqrt{d}} \begin{bmatrix} \alpha_1 u_1 \\ \vdots \\ \alpha_d u_d \end{bmatrix} \right)^T \left(\frac{(d-1)!}{d^{(d-1)/2}} \begin{bmatrix} (\tau/\alpha_1)\mathcal{A}_{(1)}\tilde{u}_1 \\ \vdots \\ (\tau/\alpha_d)\mathcal{A}_{(d)}\tilde{u}_d \end{bmatrix} \right)$$

$$= \; \frac{1}{\sqrt{d}} \cdot \frac{(d-1)!}{d^{(d-1)/2}} \cdot \tau \cdot \sum_{k=1}^{d} u_k^T \mathcal{A}_{(k)} \tilde{u}_k \;=\; \frac{(d-1)!}{\sqrt{d^d}} \cdot \tau \cdot \sum_{k=1}^{d} \sigma \;=\; \frac{d!}{\sqrt{d^d}} \cdot \tau \cdot \sigma$$

completing the proof of the theorem. □

Thus, for each Z-singular value and vector for $\mathcal{A}$ we have 2^{d-1} Z-eigenvalue/eigenvector pairs for $\mathbf{sym}(\mathcal{A})$.

2.4 Higher Order Power Methods

We now briefly review various tensor power methods and consider them in light of the singular- and eigenvalue connections between $\mathcal{A}$ and $\mathbf{sym}(\mathcal{A})$.

For details on the implementations of the algorithms discussed in this section, see Appendix A.

2.4.1 The HOPM

The matrix power method method can be generalized to tensors by replacing the matrix-vector multiplication with multilinear transforms. The *Higher-Order Power*

Method of [21, 19] for finding a Z-singular value and associated Z-singular vectors of general order-d tensors proceeds in an alternating fashion to update each of the mode-j Z-singular vectors u_j.

Algorithm 2.1 The higher-order power method (HOPM) [21, 19]

Given an order-d tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$.

Require: $u_j^{(0)} \in \mathbb{R}^{n_j}$ with $\|u_j^{(0)}\|_2 = 1$. Let $\sigma^{(0)} = \mathcal{A} \cdot (u_1^{(0)}, \ldots, u_d^{(0)})$.

1: **for** $k = 0, 1, \ldots$ **do**

2: **for** $j = 1, 2, \ldots, d$ **do**

3: $\hat{u}_j^{(k+1)} \leftarrow \mathcal{A} \cdot (u_1^{(k+1)}, \ldots, u_{j-1}^{(k+1)}, I_{n_j}, u_{j+1}^{(k)}, \ldots, u_d^{(k)})$

4: $u_j^{(k+1)} \leftarrow \hat{u}_j^{(k+1)} / \|\hat{u}_j^{(k+1)}\|_2$

5: **end for**

6: $\sigma^{(k+1)} \leftarrow \mathcal{A} \cdot (u_1^{(k+1)}, \ldots, u_d^{(k+1)})$

7: **end for**

Different initial values for the u_j vectors will in general result in convergence to different Z-singular values. See §2.4.4 for a discussion on popular choices for higher-order power method initial values.

The HOPM can also be viewed as a way of finding the best rank-1 tensor approximation $\hat{\mathcal{A}}$ to $\mathcal{A}$ [21]. It can be shown that the HOPM converges to a local minimum of the functional $f(\hat{\mathcal{A}}) \equiv \| \mathcal{A} - \hat{\mathcal{A}} \|^2$, where $\hat{\mathcal{A}} = \sigma\, u_1 \circ \cdots \circ u_d$ is a rank-1 approximation to $\mathcal{A}$ and recall from (1.29) that the norm of a tensor $\mathcal{T}$ is defined as $\| \mathcal{T} \| \equiv \sqrt{\sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \mathcal{T}(\mathbf{i})^2}$. See [36].

The HOPM can be applied to an order-d symmetric $N \times \cdots \times N$ tensor, starting with a symmetric initial guess $u_1^{(0)} = u_2^{(0)} = \cdots = u_d^{(0)} \in \mathbb{R}^N$. The solution found by the algorithm will be symmetric but intermediate results may break symmetry. Indeed, after one iteration the u_j vectors will in general all be distinct, but $u_j^{(k)} \to u$

as $k \to \infty$ for some $u \in \mathbb{R}^N$ [21].

2.4.2 The S-HOPM

Recently, [37] investigated a modified version of the HOPM to compute Z-eigenpairs of symmetric tensors. This approach was originally dismissed by [21] as unreliable since in general it is not guaranteed to converge. This algorithm is called the *Symmetric Higher Order Power Method* (S-HOPM) and does converge for certain classes of symmetric tensors. For example, suppose $\mathcal{C}$ is a symmetric tensor of even order and that M is a square unfolding of $\mathcal{C}$. If M is semidefinite then the S-HOPM converges [37].

Algorithm 2.2 Symmetric higher-order power method (S-HOPM) [21, 37]

Given an order-d symmetric tensor $\mathcal{C} \in \mathbb{R}^{N \times \cdots \times N}$.

Require: $x^{(0)} \in \mathbb{R}^N$ with $\|x^{(0)}\|_2 = 1$. Let $\lambda^{(0)} = \mathcal{C} \cdot (x^{(0)}, \ldots, x^{(0)})$.

1: **for** $k = 0, 1, \ldots$ **do**

2: $\quad \hat{x}^{(k+1)} \leftarrow \mathcal{C} \cdot (I_N, x^{(k)}, \ldots, x^{(k)})$

3: $\quad x^{(k+1)} \leftarrow \hat{x}^{(k+1)} / \|\hat{x}^{(k+1)}\|_2$

4: $\quad \lambda^{(k+1)} \leftarrow \mathcal{C} \cdot (x^{(k+1)}, \ldots, x^{(k+1)})$

5: **end for**

This approach avoids the awkward situation, mentioned previously, of encountering non-symmetric intermediate values when using the HOPM on a symmetric tensor.

Since $\mathbf{sym}(\mathcal{A})$ is symmetric for any tensor $\mathcal{A}$, the S-HOPM can be applied to $\mathcal{A}$ through its embedding. By using facts previously established, we can reduce all operations on $\mathbf{sym}(\mathcal{A})$ to equivalent ones on $\mathcal{A}$.

Algorithm 2.3 Symmetric higher-order power method on $\mathbf{sym}(\mathcal{A})$

Given an order-d tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$.

Require: $u_j^{(0)} \in \mathbb{R}^{n_j}$ with $\|u_j^{(0)}\|_2 = 1$. Let $\sigma^{(0)} = \mathcal{A} \cdot (u_1^{(0)}, \ldots, u_d^{(0)})$.

1: **for** $k = 0, 1, \ldots$ **do**

2: **for** $j = 1, 2, \ldots, d$ **do**

3: $\hat{u}_j^{(k+1)} \leftarrow \mathcal{A} \cdot (u_1^{(k)}, \ldots, u_{j-1}^{(k)}, I_{n_j}, u_{j+1}^{(k)}, \ldots, u_d^{(k)})$

4: $u_j^{(k+1)} \leftarrow \hat{u}_j^{(k+1)} / \|\hat{u}_j^{(k+1)}\|_2$

5: **end for**

6: $\sigma^{(k+1)} \leftarrow \mathcal{A} \cdot (u_1^{(k+1)}, \ldots, u_d^{(k+1)})$

7: **end for**

This algorithm computes a Z-singular value σ for $\mathcal{A}$ and the mode-j Z-singular vectors u_j. The normalization used in Algorithm 2.3 is slightly different than a direct application of the S-HOPM on $\mathbf{sym}(\mathcal{A})$ would imply; the S-HOPM would set $u_j^{(k+1)} = \hat{u}_j^{(k+1)} / \sqrt{\|\hat{u}_1^{(k+1)}\|_2^2 + \cdots + \|\hat{u}_d^{(k+1)}\|_2^2}$. However, numerical experiments suggest that using $u_j^{(k+1)} = \hat{u}_j^{(k+1)} / \|\hat{u}_j^{(k+1)}\|_2$ improves convergence. If $\mathcal{A}$ is itself symmetric, then Algorithm 2.3 reduces to the S-HOPM as all the u_j will be equal, assuming $u_1^{(0)} = \cdots = u_d^{(0)}$.

Note that Algorithm 2.3 is very similar to the regular HOPM except the most recently available information on $u_1, \ldots, u_{j-1}$ is not used when computing $u_j^{(k+1)}$ for $j > 1$. The difference between the HOPM and Algorithm 2.3 is thus somewhat like the difference between the Jacobi and Gauss-Seidel iterative linear system solvers [26].

Recall that the Jacobi and Gauss-Seidel algorithms are the classical iterative methods of solving a linear system $Ax = b$. The Jacobi iteration is

for $i = 1, 2, \ldots, n$ **do**

$$x_i^{(k+1)} \leftarrow \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \Big/ a_{ii}$$

end for

and the similar Gauss-Seidel iteration is given by

for $i = 1, 2, \ldots, n$ **do**

$$x_i^{(k+1)} \leftarrow \left(b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n} a_{ij} x_j^{(k)} \right) \Big/ a_{ii}$$

end for

Note that the primary difference between the two methods is that the Jacobi iteration does not use the most recently computed x-values. The two methods have different convergence properties and are appropriate in different situations. See [26] for details and analysis. Understanding these important matrix algorithms gives an important insight into the higher-order power methods discussed here.

Unlike the HOPM, Algorithm 2.3 does not always converge. Furthermore, it can be shown that a square unfolding of $\mathbf{sym}(\mathcal{A})$ is indefinite unless all the entries in $\mathcal{A}$ are zero, hence the convergence criteria in [37] do not apply.

2.4.3 The SS-HOPM and sym$(\cdot)$

Recently, Kolda and Mayo [42] developed a shifted version of the S-HOPM and proved that for a suitable choice of shift their algorithm will converge to a Z-eigenpair (λ, x) for any symmetric tensor $\mathcal{C}$.

Algorithm 2.4 Shifted symmetric higher-order power method (SS-HOPM) [42]

Given an order-d symmetric tensor $\mathcal{C} \in \mathbb{R}^{N \times \cdots \times N}$.

Require: $x^{(0)} \in \mathbb{R}^N$ with $\|x^{(0)}\|_2 = 1$. Let $\lambda^{(0)} = \mathcal{C} \cdot (x^{(0)}, \ldots, x^{(0)})$.

1: **for** $k = 0, 1, \ldots$ **do**

2: $\quad \hat{x}^{(k+1)} \leftarrow \mathcal{C} \cdot (I_N, x^{(k)}, \ldots, x^{(k)}) + \alpha_{\mathcal{C}}\, x^{(k)}$

3: $\quad x^{(k+1)} \leftarrow \hat{x}^{(k+1)} / \|\hat{x}^{(k+1)}\|_2$

4: $\quad \lambda^{(k+1)} \leftarrow \mathcal{C} \cdot (x^{(k+1)}, \ldots, x^{(k+1)})$

5: **end for**

If the shift $\alpha_{\mathcal{C}}$ satisfies $|\alpha_{\mathcal{C}}| > (d-1) \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{N}} |\mathcal{C}(\mathbf{i})|$ then the SS-HOPM will converge to a Z-eigenpair [42].

When $\mathcal{C} = \mathbf{sym}(\mathcal{A})$ the algorithm can be simplified and expressed in terms of operations on $\mathcal{A}$. Since this algorithm is "Jacobi-style" and uses a shift, we refer to it as the *Shifted Jacobi Higher Order Power Method* (SJ-HOPM).

Algorithm 2.5 Shifted Jacobi higher order power method (SJ-HOPM).

Given an order-d tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$.

Require: $u_j^{(0)} \in \mathbb{R}^{n_j}$ with $\|u_j^{(0)}\|_2 = 1/\sqrt{d}$. Let $\sigma^{(0)} = \mathcal{A} \cdot (u_1^{(0)}, \ldots, u_d^{(0)})$.

1: **for** $k = 0, 1, \ldots$ **do**

2: $\quad$ **for** $j = 1, 2, \ldots, d$ **do**

3: $\quad\quad \hat{u}_j^{(k+1)} \leftarrow \mathcal{A} \cdot (u_1^{(k)}, \ldots, u_{j-1}^{(k)}, I_{n_j}, u_{j+1}^{(k)}, \ldots, u_d^{(k)}) + d\, \alpha_{\mathcal{A}} u_j^{(k)}$

4: $\quad$ **end for**

5: $\quad$ **for** $j = 0, 1, \ldots$ **do**

6: $\quad\quad u_j^{(k+1)} \leftarrow \hat{u}_j^{(k+1)} / \sqrt{\|\hat{u}_1^{(k+1)}\|_2^2 + \cdots + \|\hat{u}_d^{(k+1)}\|_2^2}$

7: $\quad$ **end for**

8: $\quad \sigma^{(k+1)} \leftarrow \mathcal{A} \cdot (u_1^{(k+1)}, \ldots, u_d^{(k+1)})$

9: **end for**

Using Theorem 2.13, a simple normalization of the values returned by the SJ-HOPM gives a Z-singular value and associated unit norm Z-singular vectors of the tensor $\mathcal{A}$.

The shift $\alpha_{\mathcal{A}}$ must in general satisfy the inequality

$$|\alpha_{\mathcal{A}}| \; > \; (d-1) \sum_{\mathbf{i=1}}^{\mathbf{n}} |\mathcal{A}(\mathbf{i})|$$

to guarantee convergence, although a smaller shift might be sufficient for any particular tensor $\mathcal{A}$.

Example: Let $\mathcal{A}$ be the $2 \times 2 \times 2 \times 2$ tensor given by the unfolding

$$\mathcal{A}_{(1)} = \begin{bmatrix} 1.1650 & 0.2641 & -0.6965 & 1.2460 & 0.0751 & -1.4462 & 0.0591 & 0.5774 \\ 0.6268 & 0.8717 & 1.6961 & -0.6390 & 0.3516 & -0.7012 & 1.7971 & -0.3600 \end{bmatrix}.$$

We ran 100 trials of the HOPM, Algorithm 2.3 and the SJ-HOPM using different random starting points $u_i^{(0)}$ chosen from a uniform distribution on $[-1, 1]^{n_i}$ and suitably normalized for each algorithm. The algorithms are considered to have converged when $|\sigma^{(k+1)} - \sigma^{(k)}| < 10^{-16}$. For this example, all three algorithms converged for every starting point.

The HOPM found the singular values 2.7248 and 1.7960. Algorithm 2.3 converged to $\sigma = \pm 2.7248$. SJ-HOPM with a positive shift $\alpha_{\mathcal{A}}$ found 2.7248 and 1.7960 and using a negative shift produced the values -2.7248 and -1.7960.

For this tensor $\mathcal{A}$ the theory suggests a shift $\alpha_{\mathcal{A}}$ greater than 37.72 in absolute value to guarantee convergence. However, using $\alpha_{\mathcal{A}}$ as small as 1 will still lead to convergence and does so in many fewer iterations, sometimes by as much as a

factor of 30 when compared to the suggested shift. Setting $\alpha_{\mathcal{A}}$ to zero caused the algorithm to fail to converge for all chosen starting points.

2.4.4 Starting Values

A standard way to initialize higher-order power methods is to use a truncated form of the Higher-Order Singular Value Decomposition (HOSVD) of [20], as described in §1.4.3.

To initialize the HOPM, for example, the values $u_j^{(0)} = U_j(:, 1)$ have been shown [21] to often lie close to the best rank-1 approximation to $\mathcal{A}$.

If desired, it is possible to create the HOSVD of $\mathcal{C} = \mathbf{sym}(\mathcal{A})$ from the HOSVD of $\mathcal{A}$. For example, if $\mathcal{C} = (U_{\mathcal{C}}, \ldots, U_{\mathcal{C}}) \cdot \mathcal{S}_{\mathcal{C}}$ is the HOSVD of $\mathcal{C}$ then it can be shown that $U_{\mathcal{C}}$ is a column permutation of the block-diagonal matrix $\mathbf{diag}(U_1, U_2, \ldots, U_d)$.

There are many other ways to initialize tensor power methods. In [36] Regalia and Kofidis derive a procedure for symmetric tensors that can outperform the HOSVD-based approach. Their heuristic can be generalized to any tensor size but they describe the procedure for a fourth order symmetric tensor $\mathcal{A}$. It involves forming the unfolding $\mathcal{A}_{[1\ 2]\times[3\ 4]}$ and computing its singular value decomposition $\mathcal{A}_{[1\ 2]\times[3\ 4]} = U\Sigma V^T$. Then they find the closest (i.e. that minimizes the subspace angle) vector of the form $z \otimes z$ to $U(:, 1)$. For details, see [36].

Another possibility is to compute a tensor generalization of the QR decomposition with partial pivoting, of the form $\mathcal{A} = (Q_1, \ldots, Q_d) \cdot \mathcal{R}$ where $\mathcal{A}_{(k)} = Q_k R_k \Pi_k$ are the pivoted QR decompositions of the unfolding $\mathcal{A}_{(k)}$. It can be shown that this "HOQRD" decomposition retains some of the approximation properties of the

truncated matrix pivoted QR decomposition and can thus give a reasonable initial guess for a tensor power method. We explore this method further in §5.1. As for the HOSVD, the HOQRD of $\mathbf{sym}(\mathcal{A})$ can be constructed from the HOQRD of $\mathcal{A}$.

2.5 Tensor Rank and the sym Operation

There are several definitions of tensor rank, each of which represents some reasonable generalization of matrix rank. For an excellent review see [22]. In this brief section we relate the multilinear rank and the outer product rank of $\mathbf{sym}(\mathcal{A})$ to the multilinear rank and the outer product rank of $\mathcal{A}$.

2.5.1 Multilinear Rank

Recall from §1.3 that the *multilinear rank* of $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is the d-tuple

$$\operatorname{rank}_{\boxplus}(\mathcal{A}) \;\equiv\; (r_1(\mathcal{A}), r_2(\mathcal{A}), \ldots, r_d(\mathcal{A}))$$

where $r_i(\mathcal{A}) = \operatorname{rank}(\mathcal{A}_{(i)})$. Note that if the tensor $\mathcal{C} \in \mathbb{R}^{N \times \cdots \times N}$ is symmetric, and $R = \operatorname{rank}(\mathcal{C}_{(1)})$, then

$$\operatorname{rank}_{\boxplus}(\mathcal{C}) \;=\; (R, R, \ldots, R) \tag{2.35}$$

because $\mathcal{C}_{(1)} = \mathcal{C}_{(2)} = \cdots = \mathcal{C}_{(d)}$. If $\mathcal{C} = \mathbf{sym}(\mathcal{A})$, then it is possible to connect $\operatorname{rank}_{\boxplus}(\mathcal{C})$ to $\operatorname{rank}_{\boxplus}(\mathcal{A})$.

Theorem 2.14. *If* $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ *and* $\operatorname{rank}_{\boxplus}(\mathcal{A}) = (r_1, \ldots, r_d)$, *then*

$$\operatorname{rank}_{\boxplus}(\mathbf{sym}(\mathcal{A})) \;=\; (R, \ldots, R)$$

where $R = r_1 + \cdots + r_d$.

Proof. Suppose $\mathcal{C} = \mathbf{sym}(\mathcal{A})$ and $\mathcal{C}_{\mathbf{i}}$ is $\mathcal{C}$'s $\mathbf{i}$th block, $\mathbf{1} \leq \mathbf{i} \leq \mathbf{d}$. Let $\mathcal{C}^{(k)}$ be a $1 \times d \times \cdots \times d$ block tensor defined by

$$\mathcal{C}^{(k)}_{\mathbf{i}} \;=\; \mathcal{C}_{\mathbf{i}}$$

where $i_1 = k$ and $1 \leq i_j \leq d$ for $j = 2{:}d$. Note that if $\mathbf{i}(2{:}d)$ is a permutation of $[\,1{:}k{-}1 \;\; k{+}1{:}d\,]$, then

$$\mathcal{C}^{(k)}_{\mathbf{i}} \;=\; \mathcal{A}^{<[\,k\;\mathbf{i}(2:d)\,]>}.$$

It follows that

$$\mathrm{range}(\mathcal{C}^{(k)}_{(1)}) \;=\; \mathrm{range}(\mathcal{A}_{(k)}) \qquad k = 1, 2, \ldots, d \tag{2.36}$$

If

$$\mathcal{C}_{(1)} \;=\; \begin{bmatrix} C_1 \\ \vdots \\ C_d \end{bmatrix} \begin{matrix} \} \, n_1 \\ \vdots \\ \} \, n_d \end{matrix}$$

is a block row partitioning of $\mathcal{C}_{(1)}$, then C_k is a column permutation of $\mathcal{C}^{(k)}_{(1)}$ and so using (2.36) we have

$$\mathrm{rank}(C_k) \;=\; \mathrm{rank}(\mathcal{C}^{(k)}_{(1)}) \;=\; r_k. \tag{2.37}$$

If

$$v \;=\; \begin{bmatrix} v_1 \\ \vdots \\ v_d \end{bmatrix} \begin{matrix} \} \, n_1 \\ \vdots \\ \} \, n_d \end{matrix}$$

is a column of $\mathcal{C}_{(1)}$ then it is a a mode-1 fiber of $\mathcal{C}$ and thus can "pass through" at most one $\mathcal{C}$-block having an index that is a permutation of $1{:}d$. This means that at most one of v's subvectors is zero. It follows from (2.37) that

$$\mathrm{rank}(\mathcal{C}_{(1)}) \;=\; \sum_{k=1}^{d} \mathrm{rank}(C_k) \;=\; \sum_{k=1}^{d} r_k$$

completing the proof of the theorem. □

2.5.2 Outer Product Rank

The outer product rank of the tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ was defined in §1.3 the minimum number of rank-1 tensors, as defined in (1.6), that are needed to represent it as a sum

$$\operatorname{rank}_{\circledcirc}(\mathcal{A}) \equiv \min \left\{ r : \mathcal{A} = \sum_{i=1}^{r} u_1^{(i)} \circ u_2^{(i)} \circ \cdots \circ u_d^{(i)}, \quad u_j^{(i)} \in \mathbb{R}^{n_j} \right\}.$$

For matrices $\operatorname{rank}(\mathbf{sym}(A)) = 2 \cdot \operatorname{rank}(A)$. Indeed, If $A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$ is the SVD of A, then

$$\mathbf{sym}(A) = \sum_{i=1}^{r} \sigma_i \left(\begin{bmatrix} 0 \\ v_i \end{bmatrix} \begin{bmatrix} u_i^T & 0 \end{bmatrix} + \begin{bmatrix} u_i \\ 0 \end{bmatrix} \begin{bmatrix} 0 & v_i^T \end{bmatrix} \right).$$

Motivated by this expansion we make a definition.

Definition 2.15. *If $\mathcal{T} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is the rank-1 tensor $\mathcal{T} = t_1 \circ \cdots \circ t_d$ and $N = n_1 + \cdots n_d$, then $\pi(\mathcal{T}) \in \mathbb{R}^{N \times \cdots \times N}$ is the rank-1 tensor*

$$\pi(\mathcal{T}) = s_1 \circ \cdots \circ s_d$$

where

$$s_k = \begin{bmatrix} 0 \\ t_k \\ 0 \end{bmatrix} \begin{matrix} \} \, n_1 + \cdots + n_{k-1} \\ \} \, n_k \\ \} \, n_{k+1} + \cdots + n_d \end{matrix}$$

With this construction, we can produce an outer product expansion of $\mathbf{sym}(\mathcal{A})$ given an outer product expansion of $\mathcal{A}$.

Theorem 2.16. *If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and*

$$\mathcal{A} = \sum_{i=1}^{r} u_i^{(1)} \circ u_i^{(2)} \circ \cdots \circ u_i^{(d)}$$

where $u_1^{(k)}, \ldots, u_r^{(k)} \in \mathbb{R}^{n_k}$*, then*

$$\mathbf{sym}(\mathcal{A}) \;=\; \sum_{\mathbf{p}\in S_d} \sum_{i=1}^{r} \pi(u_i^{(1)} \circ u_i^{(2)} \circ \cdots \circ u_i^{(d)})^{<\mathbf{p}>} \tag{2.38}$$

where S_d *is the set of all permutations of* $1{:}d$*.*

Proof. Let $\mathcal{C}$ be the sum on the right side of (2.38) and note that

$$\mathcal{C} \;=\; \sum_{\mathbf{p}\in S_d} \sum_{i=1}^{r} \pi(u_i^{(p_1)} \circ u_i^{(p_2)} \circ \cdots \circ u_i^{(p_d)}).$$

We must show that the $\mathbf{q}$th block of $\mathbf{sym}(\mathcal{A})$ equals the $\mathbf{q}$th block of $\mathcal{C}_{\mathbf{q}}$. If $\mathbf{q}$ is not a permutation of $1{:}d$, then these blocks are both zero. Otherwise

$$\mathcal{C}_{\mathbf{q}} \;=\; \sum_{i=1}^{r} u_i^{(q_1)} \circ u_i^{(q_2)} \circ \cdots \circ u_i^{(q_d)} \;=\; \left(\sum_{i=1}^{r} u_i^{(1)} \circ u_i^{(2)} \circ \cdots \circ u_i^{(d)} \right)^{<\mathbf{q}>} \;=\; \mathcal{A}^{<\mathbf{q}>}.$$

completing the proof of the theorem. □

Since the double summation in (2.38) involves $rd!$ terms, it follows that

$$\operatorname{rank}_{\circledcirc}(\mathbf{sym}(\mathcal{A})) \;\leq\; d! \cdot \operatorname{rank}_{\circledcirc}(\mathcal{A}) \tag{2.39}$$

We conjecture that equality prevails. This is somewhat reminiscent of the *direct sum conjecture* [67], i.e. that $\operatorname{rank}_{\circledcirc}(\mathcal{A} \oplus \mathcal{B}) = \operatorname{rank}_{\circledcirc}(\mathcal{A}) + \operatorname{rank}_{\circledcirc}(\mathcal{B})$ where $\mathcal{A} \oplus \mathcal{B}$ is the $2 \times 2 \times \cdots \times 2$ order-d block tensor that has all zero blocks except the $(1, 1, \ldots, 1)$ block is $\mathcal{A}$ and the $(2, 2, \ldots, 2)$ block is $\mathcal{B}$. The matrix equivalent is that the rank of the matrix

$$\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix}$$

is equal to $\operatorname{rank}(A) + \operatorname{rank}(B)$, which is easily proven by using the SVD. The direct sum conjecture has been proven for a few special cases, most notably if $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3}, \mathcal{B} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ where $2 \in \{n_1, n_2, n_3, m_1, m_2, m_3\}$. See [34].

Intuitively, $\mathbf{sym}(\mathcal{A})$ contains $d!$ distinct copies of $\mathcal{A}$ in nonoverlapping index regions so if the matrix case were to generalize, any expansion of $\mathbf{sym}(\mathcal{A})$ into a sum of $\leq d!r$ rank-1 terms could be reduced to (2.38) without adding terms, thus having exactly $d!r$ terms. We have so far been unable to prove this. However, we can establish a lower bound on $\operatorname{rank}_{\circledcirc}(\mathbf{sym}(\mathcal{A}))$ by using the following lemma from [22].

Lemma 2.17 (Lim-De Silva [22])**.** *For $i = 1, 2, \ldots, d$, let $x_1^{(i)}, \ldots, x_r^{(i)} \in \mathbb{R}^{n_i}$ be linearly independent. Then the tensor*

$$\mathcal{T} \;=\; \sum_{j=1}^{r} x_j^{(1)} \circ x_j^{(2)} \circ \cdots \circ x_j^{(d)}$$

has $\operatorname{rank}_{\circledcirc}(\mathcal{T}) = r$.

Corollary 2.18. *For any $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ we have*

$$d \cdot \operatorname{rank}_{\circledcirc}(\mathcal{A}) \;\leq\; \operatorname{rank}_{\circledcirc}(\mathbf{sym}(\mathcal{A})) \tag{2.40}$$

Proof. Let $K \subset S_d$ be the set

$$\begin{aligned} K \;&=\; \{1{:}d,\ [d,\ 1{:}d{-}1],\ [d{-}1,\ d,\ 1{:}d{-}2], \ldots,\ [2{:}d{-}1,\ 1]\} \\ &=\; \{\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_d\} \end{aligned}$$

and let $\mathcal{A}$ and $\mathbf{sym}(\mathcal{A})$ be as in (2.38) and define

$$\begin{aligned} \mathcal{T} \;&=\; \sum_{i=1}^{r} \sum_{\mathbf{p} \in K} \pi(u_i^{(1)} \circ u_i^{(2)} \circ \cdots \circ u_i^{(d)})^{<\mathbf{p}>} \\ &=\; \sum_{i=1}^{r} \sum_{j=1}^{d} \pi(u_i^{(1)} \circ u_i^{(2)} \circ \cdots \circ u_i^{(d)})^{<\mathbf{p}_j>} \\ &=\; \sum_{i=1}^{r} \sum_{j=1}^{d} x_i^{(\mathbf{p}_j(1))} \circ x_i^{(\mathbf{p}_j(2))} \circ \cdots \circ x_i^{(\mathbf{p}_j(d))} \end{aligned}$$

where $r = \text{rank}_{\circledcirc}(\mathcal{A})$ and

$$x_i^{(\mathbf{p}_j(k))} = \begin{bmatrix} 0 \\ u_i^{(\mathbf{p}_j(k))} \\ 0 \end{bmatrix} \begin{matrix} \} \, n_1 + \cdots + n_{\mathbf{p}_j(k)-1} \\ \} \, n_{\mathbf{p}_j(k)} \\ \} \, n_{\mathbf{p}_j(k)+1} + \cdots + n_d \end{matrix} .$$

Because of this zero-nonzero structure we see by Lemma 2.17 that for i fixed the tensor

$$\mathcal{T}_i = \sum_{j=1}^{d} x_i^{(\mathbf{p}_j(1))} \circ x_i^{(\mathbf{p}_j(2))} \circ \cdots \circ x_i^{(\mathbf{p}_j(d))}$$

has $\text{rank}_{\circledcirc}(\mathcal{T}_i) = d$. Furthermore, for any i_1, i_2 and $j_1 \neq j_2$ the vectors

$$x_{i_1}^{(\mathbf{p}_{j_1}(1))} \circ \cdots \circ x_{i_1}^{(\mathbf{p}_{j_1}(d))} \text{ and } x_{i_2}^{(\mathbf{p}_{j_2}(1))} \circ \cdots \circ x_{i_2}^{(\mathbf{p}_{j_2}(d))}$$

have non-overlapping nonzero entries and are thus linearly independent. Finally, since $\text{rank}_{\circledcirc}(\mathcal{A}) = r$, for $j = 1, 2, \ldots, d$ we have that each tensor

$$\mathcal{S}_j = \sum_{i=1}^{r} x_i^{(\mathbf{p}_j(1))} \circ x_i^{(\mathbf{p}_j(2))} \circ \cdots \circ x_i^{(\mathbf{p}_j(d))}$$

has $\text{rank}_{\circledcirc}(\mathcal{S}_j) = r$. Combining all of these facts we have that $\text{rank}_{\circledcirc}(\mathcal{T}) = d \cdot r$.

Since none of the the other rank-1 terms in (2.38) can cancel any of the terms in $\mathcal{T}$ we have that $\mathcal{A}$ must have outer product rank at least $d \cdot r$. □

2.6 Other Eigenvalue and Singular Value Definitions

The definitions used thus far in this chapter are not the only possible extensions of singular values and eigenvalues to tensors. Indeed, other definitions exist that are more appropriate in some situations.

2.6.1 H-Singular Values

If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots n_d}$ then we say that the scalar λ is an *H-singular value* [48, 7] if there exist nonzero *H-singular vectors* $u_k \in \mathbb{R}^{n_k}$, $\| u_k \|_d = 1$ such that

$$\mathcal{A} \cdot (u_1, \ldots, u_{k-1}, I_{n_k}, u_{k+1}, \ldots, u_d) \;=\; \sigma u_k^{[d-1]} \tag{2.41}$$

for $k = 1, 2, \ldots, d$, where we define $v^{[r]} \equiv [v_1^r \;\cdots\; v_n^r]^T$ for any $v \in \mathbb{R}^n$ and $r \in \mathbb{R}$. When d is even, the H-singular vectors are a critical point of the modified tensor Rayleigh quotient

$$\frac{\mathcal{A} \cdot (u_1, \ldots, u_d)}{\| u_1 \|_d \cdots \| u_d \|_d}$$

and the H-singular value is the corresponding critical value. Lim [48] refers to these as *l^d-singular values* and shows that they exist for any even order tensor $\mathcal{A}$. The situation is more complicated if d is odd. Indeed, H-singular values are not guaranteed to exist for such tensors.

If $\mathcal{C} \in \mathbb{R}^{N \times \cdots \times N}$ is a symmetric tensor then $\lambda \in \mathbb{R}$ is an *H-eigenvalue* [48, 60] (referred to by Lim [48] as an *l^d-eigenvalue*) if there exists a vector $x \in \mathbb{R}^N \backslash \{0\}$, called an *H-eigenvector*, satisfying

$$\mathcal{C} \cdot (I_N, x, \ldots, x) \;=\; \lambda x^{[d-1]}. \tag{2.42}$$

H-eigenvalues always exist for a symmetric tensor [60]. Note that (2.42) is a homogeneous linear system so the vector x can be rescaled as desired, i.e. there is no norm constraint on H-eigenvectors.

Like the Z-singular values and Z-eigenvalues, the H-singular values and H-eigenvalues reduce to the familiar matrix cases when $d = 2$.

The relationship between H-singular values and H-eigenvalues is similar to the

one shown previously for Z-singular values and Z-eigenvalues. The proof of the following result closely mirrors the proof of Theorem 2.13 and is omitted.

Corollary 2.19. *If σ is a nonzero H-singular value of $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ with associate H-singular vectors $u_1, \ldots, u_d$, then any nonzero scalar multiple of*

$$x_\alpha \;=\; \begin{bmatrix} \alpha_1 u_1 \\ \alpha_2 u_2 \\ \vdots \\ \alpha_d u_d \end{bmatrix} \qquad \alpha \;=\; [1, \pm 1, \ldots, \pm 1]$$

is an H-eigenvector for $\mathbf{sym}(\mathcal{A})$ corresponding to the H-eigenvalue

$$\lambda_\alpha \;=\; \alpha_1 \alpha_2 \cdots \alpha_d \, (d-1)! \, \sigma.$$

2.6.2 Properties

Unfortunately, not much research has been done on the properties of H-singular values and Z-singular values (and their complex generalizations). However, very recently a closely related concept of singular values of so-called *rectangular tensors* has attracted attention of researchers [7, 8, 75]. Inspired by this work, we will explore some additional properties of the Z- and H-singular values here.

If $\sigma = 0$ and $u_i = 0$ for some, but not all, i, we say that σ and $(u_1, \ldots, u_d)$ is a *trivial value* [7] for $\mathcal{A}$ since it trivially satisfies equations (2.31) and (2.41).

We say that $\sigma \in \mathbb{C}$ is a *C-singular value* for $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ if

$$\mathcal{A} \cdot (u_1, \ldots, u_{i-1}, I_{n_i}, u_{i+1}, \ldots, u_d) \;=\; \sigma u_i^{[d-1]}$$

with *C-singular vectors* $u_i \in \mathbb{C}^{n_i}$ where $\| u_i \|_d = 1$ for $i = 1, \ldots, d$. Similarly, for

a symmetric tensor $\mathcal{C} \in \mathbb{R}^{N\times\cdots\times N}$ we say that $\lambda \in \mathbb{C}$ is a *C-eigenvalue*[1] [60] with associate *C-eigenvector* $x \in \mathbb{C}^N \backslash \{0\}$ if

$$\mathcal{C} \cdot (I_N, x, \ldots, x) \;=\; \lambda x^{[d-1]}.$$

Note that the definitions of C-singular values and C-eigenvalues are the same as those for H-singular values and H-eigenvalues except we allow complex solutions. Thus all H-singular values are also C-singular values. We note that Corollary 2.19 also holds for C-eigenvalues and C-singular values.

Recall the function $f_{\mathcal{A}}$ from (2.27):

$$f_{\mathcal{A}}(u_1, \ldots, u_d) \;=\; \mathcal{A} \cdot (u_1, \ldots, u_d).$$

Theorem 2.20. *If $\mathcal{A} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ then we have the following conclusions on the singular values of $\mathcal{A}$:*

(a) *Z-singular values of $\mathcal{A}$ exist. If d is even then $\mathcal{A}$ has H-singular values.*

(b) *If σ is a Z-singular value of $\mathcal{A}$ then so is $-\sigma$. If μ is an H- or C-singular value of $\mathcal{A}$ and d is even then $-\mu$ is also an H- or C-singular value, respectively.*

(c) *If σ is an H-, C-, or Z-singular value of $\mathcal{A}$ with associate singular values $u_1, \ldots, u_d$ then $f_{\mathcal{A}}(u_1, \ldots, u_d) = \sigma$.*

(d) *$\mathcal{A}$ has at most $N(d-1)^{N-1}$ C-singular values.*

(e) *If d is even then the sum of the C-singular values of $\mathcal{A}$ is 0.*

(f) *If $Q_i \in \mathbb{R}^{n_i\times n_i}$ are orthogonal for $i = 1, \ldots, d$ and $\mathcal{B} = (Q_1, \ldots, Q_d) \cdot \mathcal{A}$ has Z-singular value σ with associate Z-singular vectors $v_1, \ldots, v_d$ then σ is also a Z-singular value of $\mathcal{A}$ with Z-singular values $u_i = Q_i v_i$.*

[1] This notation is nonstandard in the literature, such values are often referred to simply as "eigenvalues".

Proof. (a) This follows directly from [48, Proposition 1].

(b) If $\sigma \in \mathbb{R}$ and nonzero unit vectors $u_1, \ldots, u_d$ satisfy

$$\mathcal{A} \cdot (u_1, \ldots, u_{i-1}, I_{n_i}, u_{i+1}, \ldots, u_d) \;=\; \sigma u_i$$

for $i = 1, \ldots, d$ then by substitution we see that so do $-\sigma$ and $-u_1, u_2, \ldots, u_d$. Similarly, if d is even and and σ and $u_1, \ldots, u_d$ satisfy

$$\mathcal{A} \cdot (u_1, \ldots, u_{i-1}, I_{n_i}, u_{i+1}, \ldots, u_d) \;=\; \sigma u_i^{[d-1]}$$

then so do $-\sigma$ and $-u_1, u_2, \ldots, u_d$.

(c) If σ is an H- or C-singular value of $\mathcal{A}$ with associate singular values $u_1, \ldots, u_d$ then

$$\begin{aligned} f_{\mathcal{A}}(u_1, \ldots, u_d) \;&=\; u_i^T \left(\mathcal{A} \cdot (u_1, \ldots, u_{i-1}, I_{n_i}, u_{i+1}, \ldots, u_d)\right) \\ &=\; u_i^T \left(\sigma u_i^{[d-1]}\right) \;=\; \sigma \| \, u_i \, \|_d^d \;=\; \sigma. \end{aligned}$$

A similar computation shows that $f_{\mathcal{A}}(u_1, \ldots, u_d) = \sigma$ for Z-singular values as well.

(d) We know from [60, Theorem 1(b)] that a real order-d $N \times \cdots \times N$ symmetric tensor has at most $N(d-1)^{N-1}$ C-eigenvalues. By Corollary 2.19 we know that the C-eigenvalues of $\mathbf{sym}(\mathcal{A})$ are nonzero multiples of all the C-singular values of $\mathcal{A}$ and the trivial values of $\mathcal{A}$. Hence $\mathcal{A}$ has no more than $N(d-1)^{N-1}$ C-singular values.

(e) From [60, Theorem 1(d)] we have that the sum of the C-eigenvalues of an N-dimensional order-d symmetric tensor $\mathcal{C}$ is $(d-1)^{N-1}\mathrm{tr}(\mathcal{C})$. By Corollary 2.19 and part (b) we know if d is even then the C-eigenvalues of $\mathbf{sym}(\mathcal{A})/(d-1)!$ are exactly the C-singular values of $\mathcal{A}$ and the trivial values of $\mathcal{A}$. Since the trivial values are zero then the sum of the C-eigenvalues of $\mathbf{sym}(A)/(d-1)!$ is equal to

the sum of the C-singular values of $\mathcal{A}$. However, $\mathbf{sym}(\mathcal{A})$ has a zero diagonal so $\mathrm{tr}(\mathbf{sym}(\mathcal{A})) = 0$. The result follows.

(f) By the associativity of the multilinear product we get

$$\begin{aligned}
\sigma v_i &= \mathcal{B} \cdot (v_1, \ldots, v_{i-1}, I_{n_i}, v_{i+1}, \ldots, v_d) \\
&= [\mathcal{A} \cdot (Q_1, \ldots, Q_d)] \cdot (v_1, \ldots, v_{i-1}, I_{n_i}, v_{i+1}, \ldots, v_d) \\
&= \mathcal{A} \cdot (Q_1 v_1, \ldots, Q_{i-1} v_{i-1}, Q_i, Q_{i+1} v_{i+1}, \ldots, Q_d v_d) \\
&= Q_i^T \left[\mathcal{A} \cdot (u_1, \ldots, u_{i-1}, I_{n_i}, u_{i+1}, \ldots, u_d)\right]
\end{aligned}$$

i.e.,

$$\mathcal{A} \cdot (u_1, \ldots, u_{i-1}, I_{n_i}, u_{i+1}, \ldots, u_d) \;=\; \sigma \left[Q_i v_i\right] \;=\; \sigma u_i$$

and $\| u_i \|_2 = \| Q_i v_i \|_2 = \| v_i \| = 1$. This shows that σ is a Z-singular value of $\mathcal{A}$ with associate Z-singular values $u_1, \ldots, u_d$. □

2.7 Conclusions

The symmetrization $\mathbf{sym}(\mathcal{A})$ can be used to connect algorithms for symmetric tensors and ones for general tensors. In this chapter we have shown how algorithms such as the S-HOPM and SS-HOPM give rise to non-symmetric algorithms through the symmetrization in a way that preserves many convergence properties. In particular, the non-symmetric version of the SS-HOPM we derive, and refer to as the SJ-HOPM, is guaranteed to converge for an appropriately chosen shift $\alpha_{\mathcal{A}}$. Are there other tensor methods where the symmetrization could be used to spot new connections or derive useful algorithms?

The rank properties of the symmetrization in some ways mirror the matrix case, but fundamental questions regarding the outer product rank of $\mathbf{sym}(\mathcal{A})$ remain

open. Resolution of these questions may help bridge the conceptual gap that exists between matrix rank and tensor rank.

CHAPTER 3

BLOCK TENSOR UNFOLDINGS

In this chapter we develop a notational framework for block tensor computations and define a new type of tensor unfolding scheme that respects blocking and allows for efficient and intuitive algorithms.

In §3.1 we review well-known connections between vec(·), Kronecker products, transposition, and the perfect shuffle permutation. A block version of vec(·) is defined in §3.2 and a related permutation is used to define the notion of a block unfolding. In §3.3 we show how to formulate a tensor contraction as a block matrix multiplication using the tools developed.

As an example of what we will do in this chapter, consider the mode-1 unfolding $\mathcal{A}_{(1)}$ of a 9-by-5-by-8 tensor $\mathcal{A}$ with the modes blocked by

$$\begin{aligned} 1\!:\!9 &= \left[\begin{array}{c|c|c} 1\!:\!2 & 3\!:\!5 & 6\!:\!9 \end{array}\right] \\ 1\!:\!5 &= \left[\begin{array}{c|c} 1\!:\!3 & 4\!:\!5 \end{array}\right] \\ 1\!:\!8 &= \left[\begin{array}{c|c|c|c} 1\!:\!2 & 3\!:\!4 & 5\!:\!6 & 7\!:\!8 \end{array}\right]. \end{aligned} \tag{3.1}$$

The unfolding, which is displayed in Figure 3.1, is a 9-by-40 matrix whose i-th row is $\text{vec}(\mathcal{A}(i,:,:))^T$. (Recall that vec-of-a-matrix is the vector obtained by stacking its columns.) Notice that in the unfolding, $\mathcal{A}$'s flattened blocks are not contiguous. The primary purpose of this chapter is to show how to permute the rows and columns of a vec-oriented unfolding so that its blocks are unfoldings of the tensor

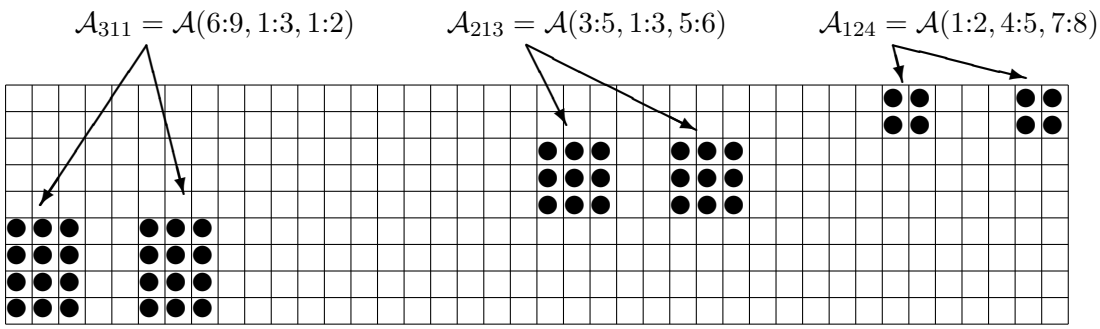


Figure 3.1: A vec-ordered, mode-1 unfolding of $\mathcal{A} \in \mathbb{R}^{9\times5\times8}$ with blocking (3.1).

2	$(\mathcal{A}_{111})_{(1)}$	$(\mathcal{A}_{121})_{(1)}$	$(\mathcal{A}_{112})_{(1)}$	$(\mathcal{A}_{122})_{(1)}$	$(\mathcal{A}_{113})_{(1)}$	$(\mathcal{A}_{123})_{(1)}$	$(\mathcal{A}_{114})_{(1)}$	$(\mathcal{A}_{124})_{(1)}$
3	$(\mathcal{A}_{211})_{(1)}$	$(\mathcal{A}_{221})_{(1)}$	$(\mathcal{A}_{212})_{(1)}$	$(\mathcal{A}_{222})_{(1)}$	$(\mathcal{A}_{213})_{(1)}$	$(\mathcal{A}_{223})_{(1)}$	$(\mathcal{A}_{214})_{(1)}$	$(\mathcal{A}_{224})_{(1)}$
4	$(\mathcal{A}_{311})_{(1)}$	$(\mathcal{A}_{321})_{(1)}$	$(\mathcal{A}_{312})_{(1)}$	$(\mathcal{A}_{322})_{(1)}$	$(\mathcal{A}_{313})_{(1)}$	$(\mathcal{A}_{323})_{(1)}$	$(\mathcal{A}_{314})_{(1)}$	$(\mathcal{A}_{324})_{(1)}$
	6	4	6	4	6	4	6	4

Figure 3.2: A "block vec"-ordered, mode-1 unfolding of $\mathcal{A} \in \mathbb{R}^{9\times5\times8}$ with blocking (3.1).

blocks. An example of such an unfolding is displayed in Figure 3.2.

3.1 Basic Notation and Operations

This chapter uses a lot of specialized block tensor notation and our analysis requires familiarity with several types of matrix permutations and their connection to the Kronecker product.

3.1.1 Transposition, Vec, Kronecker Products, and Permutation

There is an important connection between matrix transposition and perfect shuffle permutations. In particular, if $A \in \mathbb{R}^{q\times r}$ and $s = qr$, then

$$\text{vec}(A^T) \;=\; \Pi_{q,r}^T \text{vec}(A) \tag{3.2}$$

where $\Pi_{q,r} \in \mathbb{R}^{s\times s}$ is the (q,r) *perfect shuffle* permutation defined by

$$\Pi_{q,r} z \;=\; \begin{bmatrix} z(1{:}r{:}s) \\ z(2{:}r{:}s) \\ \vdots \\ z(r{:}r{:}s) \end{bmatrix} \qquad z \in \mathbb{R}^s. \tag{3.3}$$

If $Z \in \mathbb{R}^{r\times q}$ and $Y = Z^T$, then $\text{vec}(Y) = \Pi_{q,r}\text{vec}(Z)$. It is also easy to verify that $\Pi_{q,r}^T = \Pi_{r,q}$. See §3.4.1 for a discussion on efficient representation and implementations of perfect shuffle permutations in MATLAB.

If $f \in \mathbb{R}^q$ and $g \in \mathbb{R}^r$, then $g \otimes f$ is a perfect shuffle of $f \otimes g$:

$$\Pi_{q,r}\,(f \otimes g) \;=\; g \otimes f. \tag{3.4}$$

An important consequence of this result applies to the case when g is a block vector:

$$\text{diag}(\Pi_{\rho_1,q}, \ldots, \Pi_{\rho_\mu,q}) \cdot \Pi_{q,r} \cdot \left(f \otimes \begin{bmatrix} g_1 \\ \vdots \\ g_\mu \end{bmatrix} \right) \;=\; \begin{bmatrix} f \otimes g_1 \\ \vdots \\ f \otimes g_\mu \end{bmatrix}. \tag{3.5}$$

Here, $g_i \in \mathbb{R}^{\rho_i}$ and $r = \rho_1 + \cdots + \rho_\mu$.

Tensor transposition can also be characterized in terms of vec$(\cdot)$ and perfect shuffles. The following lemma can be regarded as a generalization of (3.2):

Lemma 3.1. *If* $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3 \times N_4}$ *and* $\mathcal{B} = \mathcal{A}^{<[\,1\,3\,2\,4\,]>}$*, then*

$$\mathrm{vec}(\mathcal{B}) \;=\; (I_{N_4} \otimes \Pi_{N_3,N_2} \otimes I_{N_1})\mathrm{vec}(\mathcal{A}).$$

Proof. The proof follows from well-known facts that relate Kronecker products, vec$(\cdot)$, and the perfect shuffle. See [30] and [70]. □

Although Lemma 3.1 addresses an order-4 transposition, the result can be applied to tensors of arbitrary order simply by "fusing" adjacent modes. For example, Suppose $\mathcal{C} \in \mathbb{R}^{n_1 \times \cdots \times n_7}$ and set $N_1 = n_1 n_2$, $N_2 = n_3$, $N_3 = n_4 n_5$, and $N_4 = n_6 n_7$. Define $\mathcal{A} \in \mathbb{R}^{N_1 \times N_2 \times N_3 \times N_4}$ by

$$\mathcal{C}(\mathbf{i}) \;=\; \mathcal{A}(j_1, j_2, j_3, j_4) \qquad \begin{cases} j_1 &=& ivec(\mathbf{i}(1{:}2), \mathbf{n}(1{:}2)) \\ j_2 &=& ivec(\mathbf{i}(3{:}3), \mathbf{n}(3{:}3)) \\ j_3 &=& ivec(\mathbf{i}(4{:}5), \mathbf{n}(4{:}5)) \\ j_4 &=& ivec(\mathbf{i}(6{:}7), \mathbf{n}(6{:}7)) \end{cases}.$$

Observe that $\mathrm{vec}(\mathcal{A}) = \mathrm{vec}(\mathcal{C})$ and

$$\begin{aligned} (I_{N_4} \otimes \Pi_{N_3,N_2} \otimes I_{N_1})\mathrm{vec}(\mathcal{C}) &= (I_{N_4} \otimes \Pi_{N_3,N_2} \otimes I_{N_1})\mathrm{vec}(\mathcal{A}) \\ &= \mathrm{vec}(\mathcal{A}^{<[\,1\,3\,2\,4\,]>}) \\ &= \mathrm{vec}(\mathcal{C}^{<[\,1\,2\,4\,5\,3\,6\,7\,]>}). \end{aligned}$$

Two special applications of Lemma 3.1 are worth noting. Assume $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$. If $\mathbf{p} \;=\; [\,1{:}k{-}1\,,\; k{+}1\,,\; k\,,\; k{+}2{:}d\,]$, then

$$\mathrm{vec}(\mathcal{A}^{<\mathbf{p}>}) \;=\; (I_{N_4} \otimes \Pi_{n_k,n_{k+1}} \otimes I_{N_1})\mathrm{vec}(\mathcal{A}) \tag{3.6}$$

where $N_1 = n_1 \cdots n_{k-1}$ and $N_4 = n_{k+2} \cdots n_d$. This transposition swaps two adjacent modes, e.g.,

$$\mathcal{B} = \mathcal{A}^{<\,[\,1\,2\,4\,3\,5\,]\,>} \Rightarrow \mathcal{A}(i_1, i_2, i_3, i_4, i_5) \;=\; \mathcal{B}(i_1, i_2, i_4, i_3, i_5).$$

On the other hand, if $\mathbf{p} = [\, k\,,\ 1{:}k{-}1\,,\ k{+}1{:}d\,]$, then

$$\mathrm{vec}(\mathcal{A}^{<\mathbf{p}>}) \ = \ (I_{N_4} \otimes \Pi_{N_2,n_k})\mathrm{vec}(\mathcal{A}) \tag{3.7}$$

where $N_2 = n_1 \cdots n_{k-1}$ and $N_4 = n_{k+1} \cdots n_d$. This transposition "moves" a designated mode "to the front," e.g.,

$$\mathcal{B} = \mathcal{A}^{<[\,3\,1\,2\,4\,5\,]>} \Rightarrow \mathcal{A}(i_1, i_2, i_3, i_4, i_5) \ = \ \mathcal{B}(i_3, i_1, i_2, i_4, i_5).$$

3.1.2 Unfolding Rank-1 Tensors

The tensor unfolding results shown in (1.7)-(1.18) take on a special form when $\mathcal{A}$ is a rank-1 tensor. Suppose $\mathcal{A} = a^{(1)} \circ \cdots \circ a^{(d)}$ where $a^{(k)} \in \mathbb{R}^{n_k}$ for $k = 1, 2, \ldots, d$, i.e.,

$$\mathcal{A}(i_1, \ldots, i_d) \ = \ a^{(1)}(i_1) \cdots a^{(d)}(i_d) \qquad \mathbf{1} \le \mathbf{i} \le \mathbf{n}.$$

It follows from (1.7)-(1.11) that if

$$v \ = \ \mathrm{vec}(a^{(1)} \circ \cdots \circ a^{(d)}),$$

then

$$v \ = \ a^{(d)} \otimes \cdots \otimes a^{(1)} \tag{3.8}$$

and

$$v_{ivec(\mathbf{i},\mathbf{n})} \ = \ a^{(1)}(i_1) \cdots a^{(d)}(i_d) \qquad \mathbf{1} \le \mathbf{i} \le \mathbf{n}. \tag{3.9}$$

If $\mathbf{p}$ is a permutation of $1{:}d$, then from the definition of the $\mathbf{p}$-transpose in (1.3) and the definition of $\mathcal{A}_{\mathbf{r}\times\mathbf{c}}$ in (1.12)-(1.18) we have

$$\mathcal{A}^{<\mathbf{p}>} \ = \ a^{(p_1)} \circ \cdots \circ a^{(p_d)} \tag{3.10}$$

and

$$\mathcal{A}_{\mathbf{r}\times\mathbf{c}} \;=\; \mathrm{vec}(a^{(r_1)} \circ \cdots \circ a^{(r_e)}) \;\cdot\; \mathrm{vec}(a^{(c_1)} \circ \cdots \circ a^{(c_{d-e})})^T. \tag{3.11}$$

In other words, the unfolding of a rank-1 tensor is a rank-1 matrix. These rank-1 facts simplify some of the proofs that follow in the next section.

We consider another special case that relates to the multilinear product, see §3.3.2. Suppose $\mathcal{B} = B^{(1)} \circ \cdots \circ B^{(d)}$ where $B^{(k)} \in \mathbb{R}^{q_k \times n_k}$ for $k = 1, 2, \ldots, d$, i.e.,

$$\mathcal{B}(i_1, j_1, \ldots, i_d, j_d) \;=\; B^{(1)}(i_1, j_1) \cdots B^{(d)}(i_d, j_d). \tag{3.12}$$

Note that $\mathcal{B}$ is an order-$2d$ tensor. If $\mathbf{r} = 1{:}2{:}2d$, $\mathbf{c} = 2{:}2{:}2d$, and $\mathbf{p} = [\,\mathbf{r}\;\mathbf{c}\,]$, then for all $\mathbf{i}$ and $\mathbf{j}$ that satisfy $\mathbf{1} \le \mathbf{i} \le \mathbf{q}$ and $\mathbf{1} \le \mathbf{j} \le \mathbf{n}$ we have

$$\mathcal{B}_{\mathbf{r}\times\mathbf{c}}(\alpha, \beta) \;=\; B^{(1)}(i_1, j_1) \cdots B^{(d)}(i_d, j_d)$$

where $\alpha = ivec(\mathbf{i}, \mathbf{q})$ and $\beta = ivec(\mathbf{j}, \mathbf{n})$. However, this is precisely the (α, β) entry of the matrix $B^{(d)} \otimes \cdots \otimes B^{(1)}$. Thus,

$$\left(B^{(1)} \circ \cdots \circ B^{(d)}\right)_{[\,1:2:2d\,]\times[\,2:2:2d\,]} \;=\; B^{(d)} \otimes \cdots \otimes B^{(1)}. \tag{3.13}$$

3.2 Block Notation and Operations

In this section we formalize the notion of a block tensor, develop a block version of vec(·), and explain how to permute $\mathcal{A}_{\mathbf{r}\times\mathbf{c}}$ into a block matrix whose blocks are unfoldings of $\mathcal{A}$'s blocks. The presentation is simplified if we make use of multi-indexed subscripts. Suppose

$$\mathbf{1} \le \mathbf{i} \le \mathbf{s} = [s_1, \ldots, s_e] \qquad S = s_1 \cdots s_e$$
$$\mathbf{1} \le \mathbf{j} \le \mathbf{t} = [t_1, \ldots, t_f] \qquad T = t_1 \cdots t_f$$

To say that $v_{\mathbf{i}}$ is the $\mathbf{i}$-th component of vector $v \in \mathbb{R}^S$ is to say that $v_{\mathbf{i}} = v_{ivec(\mathbf{i},\mathbf{s})}$. Similarly, if $D_1, \ldots, D_S$ are square matrices and $D = \text{diag}(\ldots, D_{\mathbf{i}}, \ldots)$, then D is a block diagonal matrix whose $\mathbf{i}$-th diagonal block is $D_{ivec(\mathbf{i},\mathbf{s})}$. Finally, if $C = (C_{ij})$ is an S-by-T block matrix, then $C_{\mathbf{i},\mathbf{j}}$ is its $(\mathbf{i},\mathbf{j})$-th block, i.e., $C_{\mathbf{i},\mathbf{j}} = C_{ivec(\mathbf{i},\mathbf{s}),ivec(\mathbf{j},\mathbf{t})}$.

3.2.1 Tensor Blockings

Recall our use of blocking vectors in §2.2.1. Expanding on this, we say that the collection

$$\mathbf{M} \;=\; \left\{\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(d)}\right\} \tag{3.14}$$

is a *blocking* for $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ if

$$\mathbf{m}^{(k)} \;=\; \left[\, m_1^{(k)}, \ldots, m_{b_k}^{(k)} \,\right] \tag{3.15}$$

is a vector of positive integers that sums to n_k for $k = 1, 2, \ldots, d$. If $\mathbf{1} \le \mathbf{i} \le \mathbf{b}$, then block $\mathbf{i}$ is the $m_{i_1}^{(1)} \times \cdots \times m_{i_d}^{(d)}$ tensor defined by

$$\mathcal{A}_{\mathbf{i}} \;=\; \mathcal{A}(\, \ell_{i_1}^{(1)}{:}u_{i_1}^{(1)}, \ldots, \ell_{i_d}^{(d)}{:}u_{i_d}^{(d)} \,) \tag{3.16}$$

where the lower and upper bound vectors $\boldsymbol{\ell}^{(1)}, \ldots, \boldsymbol{\ell}^{(d)}$ and $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(d)}$ are defined by

$$\ell_j^{(k)} \;=\; m_1^{(k)} + \cdots + m_{j-1}^{(k)} \;+\; 1 \tag{3.17}$$

$$u_j^{(k)} \;=\; m_1^{(k)} + \cdots + m_{j-1}^{(k)} \;+\; m_j^{(k)} \tag{3.18}$$

for $k = 1, 2, \ldots, d$. The blocking $\mathbf{M}$ identifies $\mathcal{A}$ as a $b_1 \times b_2 \times \cdots \times b_d$ block tensor. The number of elements in each tensor block $\mathcal{A}_{\mathbf{i}}$ turns out to be a quantity of importance and to that end we define the "volume function" $\text{vol}_{\mathbf{M}}(\cdot)$ by

$$\text{vol}_{\mathbf{M}}(\mathbf{i}) \;=\; m_{i_1}^{(1)} \cdots m_{i_d}^{(d)} \qquad \mathbf{1} \le \mathbf{i} \le \mathbf{b}. \tag{3.19}$$

We have written an extension of the Tensor Toolbox [40]. It includes a new class: `bltensor`, which represents a blocked tensor object which allows for fast and easy prototyping of block tensors algorithms and manipulations. A full description is in Appendix A.

As an example of how to use this class, here is how to create a block tensor with the blocking (3.1).

```
A = tensor(rand(9,5,8));
M = {[2 3 4], [3 2], [2 2 2 2]};
A_blocked = bltensor(A, M);
```

Then the command `A(1,1,1)` will return the (1,1,1) block of $\mathcal{A}$, which is a $2 \times 3 \times 2$ tensor, and the command `A(2,1,3)=0` will set the (2,1,3) block to be all zeros.

3.2.2 The $\mathrm{Vec}_{\mathbf{M}}(\cdot)$ Operation

If $\mathbf{M}$ is a blocking of $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ given by (3.14)-(3.18), then $\mathrm{vec}_{\mathbf{M}}(\mathcal{A})$ is the block vector

$$\mathrm{vec}_{\mathbf{M}}(\mathcal{A}) \;=\; \begin{bmatrix} v_{\mathbf{1}} \\ \vdots \\ v_{\mathbf{b}} \end{bmatrix} \qquad v_{\mathbf{i}} \;=\; \mathrm{vec}(\mathcal{A}_{\mathbf{i}}) \tag{3.20}$$

where $\mathbf{1} \leq \mathbf{i} \leq \mathbf{b}$. In other words, $\mathrm{vec}_{\mathbf{M}}(\mathcal{A})$ stacks the vec's of $\mathcal{A}$'s blocks where the blocks are taken in the vec-order.

To illustrate this notation in the familiar matrix case, if

$$\mathbf{M} \;=\; \{m^{(1)}, m^{(2)}\} \;=\; \{\, [\, m_1^{(1)}\; m_2^{(1)}\,]\,,\; [\, m_1^{(2)}\; m_2^{(2)}\; m_3^{(2)}\,]\, \}$$

is a blocking for $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2}$, then we are choosing to regard $\mathcal{A}$ as a 2-by-3 block

matrix

$$\mathcal{A} \;=\; \begin{bmatrix} \mathcal{A}_{11} & \mathcal{A}_{12} & \mathcal{A}_{13} \\ \mathcal{A}_{21} & \mathcal{A}_{22} & \mathcal{A}_{23} \end{bmatrix} \begin{matrix} m_1^{(1)} \\ m_2^{(1)} \end{matrix} \\ \begin{matrix} m_1^{(2)} & m_2^{(2)} & m_3^{(2)} \end{matrix} \qquad (3.21)$$

In this case, $\mathrm{vec}_{\mathbf{M}}(\cdot)$ and $\mathrm{vol}_{\mathbf{M}}(\cdot)$ are given by

$$\mathrm{vec}_{\mathbf{M}}(A) \;=\; \begin{bmatrix} v_{[1,1]} \\ v_{[2,1]} \\ v_{[1,2]} \\ v_{[2,2]} \\ v_{[1,3]} \\ v_{[2,3]} \end{bmatrix} = \begin{bmatrix} \mathrm{vec}(\mathcal{A}_{11}) \\ \mathrm{vec}(\mathcal{A}_{21}) \\ \mathrm{vec}(\mathcal{A}_{12}) \\ \mathrm{vec}(\mathcal{A}_{22}) \\ \mathrm{vec}(\mathcal{A}_{13}) \\ \mathrm{vec}(\mathcal{A}_{23}) \end{bmatrix} \qquad \mathrm{vol}_{\mathbf{M}}(\mathbf{i}) \;=\; \begin{cases} m_1^{(1)} m_1^{(2)} & \text{if } \mathbf{i} = [1,1] \\ m_2^{(1)} m_1^{(2)} & \text{if } \mathbf{i} = [2,1] \\ m_1^{(1)} m_2^{(2)} & \text{if } \mathbf{i} = [1,2] \\ m_2^{(1)} m_2^{(2)} & \text{if } \mathbf{i} = [2,2] \\ m_1^{(1)} m_3^{(2)} & \text{if } \mathbf{i} = [1,3] \\ m_2^{(1)} m_3^{(2)} & \text{if } \mathbf{i} = [2,3] \end{cases}.$$

As we mentioned in the introduction, our goal is to permute the rows and columns of the unfolding $\mathcal{A}_{\mathbf{r}\times\mathbf{c}}$ so that its blocks are unfoldings of $\mathcal{A}$'s blocks. To be more precise, if $\mathcal{A} = (\mathcal{A}_{\mathbf{i}})$ is a block tensor our goal is to determine permutation matrices $P_{\mathbf{R}}$ and $P_{\mathbf{C}}$ so that

$$\mathcal{A}_{\mathbf{R}\times\mathbf{C}} \;=\; P_{\mathbf{R}} \mathcal{A}_{\mathbf{r}\times\mathbf{c}} P_{\mathbf{C}}^T \qquad (3.22)$$

is a block matrix whose blocks are the matrices $(\mathcal{A}_{\mathbf{k}})_{\mathbf{r}\times\mathbf{c}}$. It turns out that the permutations that do this map "vec-of-a-tensor" to "$\mathrm{vec}_{\mathbf{M}}$-of-a-tensor." This is not surprising since the rows and columns of $\mathcal{A}_{\mathbf{r}\times\mathbf{c}}$ are vec's of reduced order block tensors, see (1.17)-(1.20).

Theorem 3.2. *Suppose* $\mathbf{M} = \{\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(d)}\}$ *is a blocking of* $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ *with*

$$\mathbf{m}^{(k)} \;=\; [m_1^{(1)}, \ldots, m_{b_k}^{(k)}] \qquad k \;=\; 1, 2, \ldots, d.$$

For $k = 1, 2, \ldots, d$ *set*

$$\begin{aligned} N_k &= n_1 \cdots n_k, \\ \mathbf{M}_k &= \{\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(k)}\}, . \end{aligned}$$

and define

$$Q_k \;=\; \begin{cases} I_{N_d} & \text{if } k = 1 \\ I_{N_d/N_k} \otimes \Gamma^{(k)} & \text{if } 1 < k \leq d \end{cases} \tag{3.23}$$

where $N_d/N_k = n_{k+1}n_{k+2}\cdots n_d$,

$$\Gamma^{(k)} \;=\; \mathrm{diag}(\Gamma_1^{(k)}, \ldots, \Gamma_{b_k}^{(k)}) \tag{3.24}$$

and

$$\Gamma_j^{(k)} \;=\; \mathrm{diag}(\ldots, \Pi_{\mathrm{vol}_{\mathbf{M}_{k-1}}(\mathbf{i}),m_j^{(k)}}, \ldots) \cdot \Pi_{m_j^{(k)},N_{k-1}} \qquad \mathbf{1} \leq \mathbf{i} \leq \mathbf{b}(1{:}k{-}1). \tag{3.25}$$

The permutation matrix $P_{\mathbf{M}}$ *defined by*

$$P_{\mathbf{M}} \;=\; Q_d \cdots Q_2 Q_1$$

has the property that

$$\mathrm{vec}_{\mathbf{M}}(\mathcal{A}) \;=\; P_{\mathbf{M}}\mathrm{vec}(\mathcal{A}).$$

Proof. Since both $\mathrm{vec}(\cdot)$ and $\mathrm{vec}_M(\cdot)$ are linear operators and any tensor is the sum of rank-1 tensors, it suffices to prove the theorem for the case

$$\mathcal{A} \;=\; a^{(1)} \circ \cdots \circ a^{(d)}$$

where each $a^{(k)} \in \mathbb{R}^{n_k}$ is blocked as follows:

$$a^{(k)} \;=\; \begin{bmatrix} a_1^{(k)} \\ \vdots \\ a_{b_k}^{(k)} \end{bmatrix} \begin{matrix} \} m_1^{(k)} \\ \\ \} m_{b_k}^{(k)} \end{matrix} .$$

We proceed by induction noting that the theorem is true if $d = 1$ because in that case, $\mathrm{vec}_{\mathbf{M}}(\mathcal{A}) = \mathrm{vec}(\mathcal{A})$. Assume that the theorem holds for block tensors with order $d-1$ or less with $d > 1$. Define

$$\begin{aligned} \widehat{\mathcal{A}} &= a^{(1)} \circ \cdots \circ a^{(d-1)} \\ \widehat{\mathbf{M}} &= \mathbf{M}_{d-1} \\ \widehat{\mathbf{b}} &= \mathbf{b}(1{:}d{-}1). \end{aligned}$$

and observe that $\widehat{\mathbf{M}}$ is a blocking for $\widehat{\mathcal{A}}$, an order-$(d-1)$ tensor. It follows by induction that

$$\mathrm{vec}_{\widehat{\mathbf{M}}}(\widehat{\mathcal{A}}) \;=\; P_{\widehat{\mathbf{M}}}\,\mathrm{vec}(\widehat{\mathcal{A}}). \tag{3.26}$$

From the definition of $\mathrm{vec}_{\mathbf{M}}(\cdot)$ in (3.20), we have

$$\mathrm{vec}_{\widehat{\mathbf{M}}}(\widehat{\mathcal{A}}) \;=\; \begin{bmatrix} v_{\mathbf{1}} \\ \vdots \\ v_{\widehat{\mathbf{b}}} \end{bmatrix} \qquad v_{\mathbf{i}} \;=\; a^{(d-1)}_{i_{d-1}} \otimes \cdots \otimes a^{(1)}_{i_1} \tag{3.27}$$

for all $\mathbf{i}$ that satisfy $\mathbf{1} \le \mathbf{i} \le \widehat{\mathbf{b}}$. Equation (3.8) says that

$$\mathrm{vec}(\mathcal{A}) \;=\; a^{(d)} \otimes (a^{(d-1)} \otimes \cdots \otimes a^{(1)}) \;=\; a^{(d)} \otimes \mathrm{vec}(\widehat{\mathcal{A}}),$$

and so

$$(I_{n_d} \otimes P_{\widehat{\mathbf{M}}})\mathrm{vec}(\mathcal{A}) \;=\; a^{(d)} \otimes v \;=\; \begin{bmatrix} a^{(d)}_1 \\ \vdots \\ a^{(d)}_{b_d} \end{bmatrix} \otimes v \;=\; \begin{bmatrix} a^{(d)}_1 \otimes v \\ \vdots \\ a^{(d)}_{b_d} \otimes v \end{bmatrix}. \tag{3.28}$$

Using (3.4) we have for $j = 1, 2, \ldots, b_d$ that

$$\Gamma^{(d)}_j \left(a^{(d)}_j \otimes v \right) \;=\; \begin{bmatrix} a^{(d)}_j \otimes v_{\mathbf{1}} \\ \vdots \\ a^{(d)}_j \otimes v_{\widehat{\mathbf{b}}} \end{bmatrix}$$

where

$$\Gamma^{(d)}_j \;=\; \mathrm{diag}\left(\Pi_{\mathrm{vol}_{\widehat{\mathbf{M}}}(\mathbf{1}),m^{(d)}_j}, \ldots, \Pi_{\mathrm{vol}_{\widehat{\mathbf{M}}}(\mathbf{b}(1:d-1)),m^{(d)}_j} \right) \cdot \Pi_{m^{(d)}_j, N/n_d}.$$

Thus, if $\Gamma^{(d)} = \mathrm{diag}(\Gamma_1^{(d)}, \ldots, \Gamma_{b_d}^{(d)})$, then

$$\Gamma^{(d)} \left[\begin{array}{c} a_1^{(d)} \otimes v \\ \hline \vdots \\ \hline a_{b_d}^{(d)} \otimes v \end{array}\right] = \left[\begin{array}{c} a_1^{(d)} \otimes v_{\mathbf{1}} \\ \vdots \\ a_1^{(d)} \otimes v_{\widehat{\mathbf{b}}} \\ \hline \vdots \\ \hline a_{b_d}^{(d)} \otimes v_{\mathbf{1}} \\ \vdots \\ a_{b_d}^{(d)} \otimes v_{\widehat{\mathbf{b}}} \end{array}\right] = \mathrm{vec}_{\mathbf{M}}(\mathcal{A}). \tag{3.29}$$

Combining this equation with (3.28) we have

$$\Gamma^{(d)}(I_{n_d} \otimes P_{\widehat{\mathbf{M}}})\mathrm{vec}(\mathcal{A}) \;=\; \mathrm{vec}_{\mathbf{M}}(\mathcal{A})$$

and so

$$P_{\mathbf{M}} = \Gamma^{(d)}(I_{n_d} \otimes P_{\widehat{\mathbf{M}}}). \tag{3.30}$$

But by induction

$$P_{\widehat{\mathbf{M}}} \;=\; \widehat{Q}_{d-1} \cdots \widehat{Q}_2 \, \widehat{Q}_1$$

where

$$\widehat{Q}_k \;=\; \begin{cases} I_{N_{d-1}} & \text{if } k = 1 \\ I_{N_{d-1}/N_k} \otimes \Gamma^{(k)} & \text{if } 1 < k \leq d-1 \end{cases}.$$

It follows that

$$\begin{aligned} P_{\mathbf{M}} &= \Gamma^{(d)}(I_{n_d} \otimes P_{\widehat{\mathbf{M}}}) \\ &= \Gamma^{(d)}(I_{n_d} \otimes \widehat{Q}_{d-1}) \cdots (I_{n_d} \otimes \widehat{Q}_2)(I_{n_d} \otimes \widehat{Q}_1) \\ &= (I_{N_d/N_d} \otimes \Gamma^{(d)})(I_{N_d/N_{d-1}} \otimes \Gamma^{(d-1)}) \cdots (I_{N_d/N_2} \otimes \Gamma^{(2)})(I_{N_d}) \\ &= Q_d \, Q_{d-1} \cdots Q_2 \, Q_1 \end{aligned}$$

completing the proof. □

The permutation $P_{\mathbf{M}}$ has a particularly simple form if the blocking is uniform in each dimension.

Corollary 3.3. *Suppose* $\mathbf{M}$ *is defined by* (3.14)-(3.18). *If*

$$\begin{aligned} m_1^{(k)} &= \cdots = m_{b_k}^{(k)} = \mu_k \\ N_k &= n_1 \cdots n_k \\ B_k &= b_1 \cdots b_k \\ D_k &= \mu_1 \cdots \mu_k \end{aligned}$$

for $k = 1, 2, \ldots, d$, *then* $P_{\mathbf{M}} = Q_d \cdots Q_2 Q_1$ *where*

$$Q_k = \begin{cases} I_{N_d} & \text{if } k = 1 \\ I_{b_k N_d / N_k} \otimes \Pi_{\mu_k, B_{k-1}} \otimes I_{D_{k-1}} & \text{if } 1 < k \leq d \end{cases}$$

Proof. Observe that $\mathrm{vol}_{\mathbf{M}_{k-1}}(\mathbf{i}) = \mu_1 \cdots \mu_{k-1}$. It follows from the definition of $\Gamma_j^{(k)}$ in (3.25) that

$$\Gamma_j^{(k)} = \left(I_{B_{k-1}} \otimes \Pi_{D_{k-1}, \mu_k}\right) \Pi_{\mu_k, N_{k-1}}.$$

Using the well-known Kronecker product identity

$$\left(I_s \otimes \Pi_{r,q}\right) \Pi_{q,rs} = \Pi_{q,s} \otimes I_r$$

see [70], it follows that

$$\Gamma_j^{(k)} = \Pi_{\mu_k, B_{k-1}} \otimes I_{D_{k-1}}.$$

From (3.24) we see that

$$\Gamma^{(k)} = I_{b_k} \otimes \Pi_{\mu_k, B_{k-1}} \otimes I_{D_{k-1}}$$

and so

$$Q_k = I_{N_d / N_k} \otimes \Gamma^{(k)} = I_{N_d b_k / N_k} \otimes \Pi_{\mu_k, B_{k-1}} \otimes I_{D_{k-1}}$$

thereby completing the proof. □

It is interesting to note that the transition from $\mathrm{vec}(\mathcal{A})$ to $\mathrm{vec}_{\mathbf{M}}(\mathcal{A})$ via the sequence

$$Q_2 \cdot \mathrm{vec}(\mathcal{A}) \;\rightarrow\; Q_3 \cdot (Q_2 \cdot \mathrm{vec}(\mathcal{A})) \;\rightarrow \cdots \rightarrow\; Q_d \cdot (Q_{d-1} \cdots Q_2 \cdot \mathrm{vec}(\mathcal{A}))$$

is actually a sequence of transpositions. To illustrate, assume $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$ and define the order-8 tensor $\mathcal{A}^{(1)}$ by

$$\mathcal{A}(i_1, i_2, i_3, i_4) \;=\; \mathcal{A}^{(1)}(\delta_1, \beta_1, \delta_2, \beta_2, \delta_3, \beta_3, \delta_4, \beta_4)$$

where $\mathbf{1} \leq \mathbf{i} \leq \mathbf{n}$ and the δ_k and β_k are uniquely defined by

$$i_k = \delta_k + (\beta_k - 1) b_k \qquad 1 \leq \delta_k \leq \mu_k.$$

This says that $\mathcal{A}^{(1)} \in \mathbb{R}^{\mu_1 \times b_1 \times \mu_2 \times b_2 \times \mu_3 \times b_3 \times \mu_4 \times b_4}$. In the $d = 4$ case, the Q-matrices in Corollary 3.3 are given by

$$\begin{aligned} Q_2 &= I_{b_2 n_3 n_4} \otimes \Pi_{\mu_2, b_1} \otimes I_{\mu_1} \\ Q_3 &= I_{b_3 n_4} \otimes \Pi_{\mu_3, b_1 b_2} \otimes I_{\mu_1 \mu_2} \\ Q_4 &= I_{b_4} \otimes \Pi_{\mu_4, b_1 b_2 b_3} \otimes I_{\mu_1 \mu_2 \mu_3}. \end{aligned}$$

Note from Lemma 3.1 that these permutations correspond to transpositions. Indeed, if we define the tensors $\mathcal{A}^{(2)}$, $\mathcal{A}^{(3)}$ $\mathcal{A}^{(4)}$ by

$$\left.\begin{aligned} &\mathcal{A}^{(2)}(\delta_1, \delta_2, \beta_1, \beta_2, \delta_3, \beta_3, \delta_4, \beta_4) \\ &\mathcal{A}^{(3)}(\delta_1, \delta_2, \delta_3, \beta_1, \beta_2, \beta_3, \delta_4, \beta_4) \\ &\mathcal{A}^{(4)}(\delta_1, \delta_2, \delta_3, \delta_4, \beta_1, \beta_2, \beta_3, \beta_4) \end{aligned}\right\} \;=\; \mathcal{A}^{(1)}(\delta_1, \beta_1, \delta_2, \beta_2, \delta_3, \beta_3, \delta_4, \beta_4)$$

then it can be shown via Lemma 3.1 that

$$\begin{aligned} \mathrm{vec}(\mathcal{A}^{(1)}) &= Q_1 \mathrm{vec}(\mathcal{A}) \;=\; \mathrm{vec}(\mathcal{A}) \\ \mathrm{vec}(\mathcal{A}^{(2)}) &= Q_2 \mathrm{vec}(\mathcal{A}^{(1)}) \\ \mathrm{vec}(\mathcal{A}^{(3)}) &= Q_3 \mathrm{vec}(\mathcal{A}^{(2)}) \\ \mathrm{vec}_{\mathbf{M}}(\mathcal{A}) \;=\; \mathrm{vec}(\mathcal{A}^{(4)}) &= Q_4 \mathrm{vec}(\mathcal{A}^{(3)}). \end{aligned}$$

Thus, the order-8 tensor $\mathcal{A}^{(4)}$ has the property that $\text{vec}(\mathcal{A}^{(4)}) = \text{vec}_{\mathbf{M}}(\mathcal{A})$. Moreover, $\mathcal{A}(\mathbf{i}) = \mathcal{A}_{\boldsymbol{\beta}}(\boldsymbol{\delta})$ showing that entry $\mathbf{i}$ is entry $\boldsymbol{\delta}$ of block $\boldsymbol{\beta}$.

3.2.3 Block Unfoldings

We now specify the permutation matrices $P_{\mathbf{R}}$ and $P_{\mathbf{C}}$ in (3.22) that turn $\mathcal{A}_{\mathbf{r}\times\mathbf{c}}$ into a block matrix with block entries that are $\mathbf{r}\times\mathbf{c}$ unfoldings of $\mathcal{A}$'s blocks.

Theorem 3.4. *Suppose* $\mathbf{M} = \{m^{(1)}, \ldots, m^{(d)}\}$ *is a blocking of* $\mathcal{A} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ *with*

$$\mathbf{m}^{(k)} \;=\; [m_1^{(k)}, \ldots, m_{b_k}^{(k)}] \qquad k \;=\; 1, 2, \ldots, d.$$

Let e *be an integer that satisfies* $1 \le e < d$ *and assume that* $\mathbf{p}$ *is a permutation of* 1:d. *Define*

$$\begin{array}{llllll} \mathbf{r} &=& \mathbf{p}(1{:}e) & \mathbf{R} &=& \{\mathbf{m}^{(r_1)}, \ldots, \mathbf{m}^{(r_e)}\} \qquad B_{rows} = b_{r_1}\cdots b_{r_e} \\ \mathbf{c} &=& \mathbf{p}(e+1{:}d) & \mathbf{C} &=& \{\mathbf{m}^{(c_1)}, \ldots, \mathbf{m}^{(c_{d-e})}\} \qquad B_{cols} = b_{c_1}\cdots b_{c_{d-e}}. \end{array}$$

The matrix

$$\mathcal{A}_{\mathbf{R}\times\mathbf{C}} \;=\; P_{\mathbf{R}}\mathcal{A}_{\mathbf{r}\times\mathbf{c}}P_{\mathbf{C}}^T.$$

is a B_{rows}*-by-*B_{cols} *block matrix whose block entries are specified by*

$$(\mathcal{A}_{\mathbf{R}\times\mathbf{C}})_{\mathbf{k(r)},\mathbf{k(c)}} \;=\; (\mathcal{A}_{\mathbf{k}})_{\mathbf{r}\times\mathbf{c}} \qquad \mathbf{1} \le \mathbf{k} \le \mathbf{b}. \tag{3.31}$$

That is to say, if $\mu = ivec(\mathbf{k}(\mathbf{r}), \mathbf{b}(\mathbf{r}))$ *and* $\tau = ivec(\mathbf{k}(\mathbf{c}), \mathbf{b}(\mathbf{c}))$, *then the* (μ, τ) *block of* $\mathcal{A}_{\mathbf{R}\times\mathbf{C}}$ *is the* $\mathbf{r}\times\mathbf{c}$ *unfolding of the* $\mathbf{k}$*-th block of* $\mathcal{A}$.

Proof. By linearity there is no loss of generality in assuming that

$$\mathcal{A} \;=\; a^{(1)} \circ \cdots \circ a^{(d)}$$

where each $a^{(k)} \in \mathbb{R}^{n_k}$ is blocked as follows:

$$a^{(k)} \;=\; \begin{bmatrix} a_1^{(k)} \\ \vdots \\ a_{b_k}^{(k)} \end{bmatrix} \begin{matrix} \} \, m_1^{(k)} \\ \\ \} \, m_{b_k}^{(k)} \end{matrix} .$$

From (3.11) we know that

$$A_{\mathbf{r}\times\mathbf{c}} \;=\; \mathrm{vec}(\, a^{(r_1)} \circ \cdots \circ a^{(r_e)} \,) \cdot \mathrm{vec}(\, a^{(c_1)} \circ \cdots \circ a^{(c_{d-e})} \,)^T .$$

Since $\mathbf{R}$ is a blocking for $a^{(r_1)} \circ \cdots \circ a^{(r_e)}$ and $\mathbf{C}$ is a blocking for $a^{(c_1)} \circ \cdots \circ a^{(c_{d-e})}$, it follows from Theorem 3.2 that

$$P_{\mathbf{R}} A_{\mathbf{r}\times\mathbf{c}} P_{\mathbf{C}}^T \;=\; yz^T$$

where $y \;=\; \mathrm{vec}_{\mathbf{R}}(a^{(r_1)} \circ \cdots \circ a^{(r_e)})$ and $z \;=\; \mathrm{vec}_{\mathbf{C}}(a^{(c_1)} \circ \cdots \circ a^{(c_{d-e})})$. These block vectors are specified by

$$y \;=\; \begin{bmatrix} y_{\mathbf{1}} \\ \vdots \\ y_{\mathbf{b(r)}} \end{bmatrix} \qquad y_{\mathbf{i}} \;=\; \mathrm{vec}(a_{i_1}^{(r_1)} \circ \cdots \circ a_{i_e}^{(r_e)}) \qquad \mathbf{1} \le \mathbf{i} \le \mathbf{b(r)} \tag{3.32}$$

$$z \;=\; \begin{bmatrix} z_{\mathbf{1}} \\ \vdots \\ z_{\mathbf{b(c)}} \end{bmatrix} \qquad z_{\mathbf{j}} \;=\; \mathrm{vec}(a_{j_1}^{(c_1)} \circ \cdots \circ a_{j_{d-e}}^{(c_{d-e})}) \qquad \mathbf{1} \le \mathbf{j} \le \mathbf{b(c)} \tag{3.33}$$

and so the $(\mathbf{i},\mathbf{j})$-th block of $\mathcal{A}_{\mathbf{R}\times\mathbf{C}}$ is given by

$$(\mathcal{A}_{\mathbf{R}\times\mathbf{C}})_{\mathbf{i},\mathbf{j}} \;=\; y_{\mathbf{i}} \, z_{\mathbf{j}}^T . \tag{3.34}$$

On the other hand, from (3.31)

$$\begin{aligned} (\mathcal{A}_{\mathbf{k}})_{\mathbf{r}\times\mathbf{c}} \;&=\; \left(a_{k_1}^{(1)} \circ \cdots \circ a_{k_d}^{(d)} \right)_{\mathbf{r}\times\mathbf{c}} \\ &=\; \mathrm{vec}\left(a_{k_{r_1}}^{(r_1)} \circ \cdots \circ a_{k_{r_e}}^{(r_e)} \right) \cdot \mathrm{vec}\left(a_{k_{c_1}}^{(c_1)} \circ \cdots \circ a_{k_{c_{d-e}}}^{(c_{d-e})} \right)^T . \end{aligned}$$

It follows from (3.32)-(3.34) that if $\mathbf{i} = \mathbf{k}(\mathbf{r})$ and $\mathbf{j} = \mathbf{k}(\mathbf{c})$, then

$$(\mathcal{A}_{\mathbf{k}})_{\mathbf{r}\times\mathbf{c}} \;=\; y_{\mathbf{i}}\, z_{\mathbf{j}}^{T} \;=\; (\mathcal{A}_{\mathbf{R}\times\mathbf{C}})_{\mathbf{i},\mathbf{j}}$$

which completes the proof. □

To illustrate the theorem, suppose $\mathcal{A}$ is 2-by-4-by-3-by-2 block tensor. If $\mathbf{r} = [1\ 3]$ and $\mathbf{c} = [2\ 4]$, then

$$\mathcal{A}_{\mathbf{R}\times\mathbf{C}} \;=\; \begin{bmatrix} \widetilde{\mathcal{A}}_{1111} & \widetilde{\mathcal{A}}_{1211} & \widetilde{\mathcal{A}}_{1311} & \widetilde{\mathcal{A}}_{1411} & \widetilde{\mathcal{A}}_{1112} & \widetilde{\mathcal{A}}_{1212} & \widetilde{\mathcal{A}}_{1312} & \widetilde{\mathcal{A}}_{1412} \\ \widetilde{\mathcal{A}}_{2111} & \widetilde{\mathcal{A}}_{2211} & \widetilde{\mathcal{A}}_{2311} & \widetilde{\mathcal{A}}_{2411} & \widetilde{\mathcal{A}}_{2112} & \widetilde{\mathcal{A}}_{2212} & \widetilde{\mathcal{A}}_{2312} & \widetilde{\mathcal{A}}_{2412} \\ \widetilde{\mathcal{A}}_{1121} & \widetilde{\mathcal{A}}_{1221} & \widetilde{\mathcal{A}}_{1321} & \widetilde{\mathcal{A}}_{1421} & \widetilde{\mathcal{A}}_{1122} & \widetilde{\mathcal{A}}_{1222} & \widetilde{\mathcal{A}}_{1322} & \widetilde{\mathcal{A}}_{1422} \\ \widetilde{\mathcal{A}}_{2121} & \widetilde{\mathcal{A}}_{2221} & \widetilde{\mathcal{A}}_{2321} & \widetilde{\mathcal{A}}_{2421} & \widetilde{\mathcal{A}}_{2122} & \widetilde{\mathcal{A}}_{2222} & \widetilde{\mathcal{A}}_{2322} & \widetilde{\mathcal{A}}_{2422} \\ \widetilde{\mathcal{A}}_{1131} & \widetilde{\mathcal{A}}_{1231} & \widetilde{\mathcal{A}}_{1331} & \widetilde{\mathcal{A}}_{1431} & \widetilde{\mathcal{A}}_{1132} & \widetilde{\mathcal{A}}_{1232} & \widetilde{\mathcal{A}}_{1332} & \widetilde{\mathcal{A}}_{1432} \\ \widetilde{\mathcal{A}}_{2131} & \widetilde{\mathcal{A}}_{2231} & \widetilde{\mathcal{A}}_{2331} & \widetilde{\mathcal{A}}_{2431} & \widetilde{\mathcal{A}}_{2132} & \widetilde{\mathcal{A}}_{2232} & \widetilde{\mathcal{A}}_{2332} & \widetilde{\mathcal{A}}_{2432} \end{bmatrix} \begin{matrix} (1,1) \\ (2,1) \\ (1,2) \\ (2,2) \\ (1,3) \\ (2,3) \end{matrix}$$

$$\begin{matrix} (1,1) & (2,1) & (3,1) & (4,1) & (1,2) & (2,2) & (3,2) & (4,2) \end{matrix}$$

where $\widetilde{\mathcal{A}}_{\alpha\beta\gamma\delta} \;=\; (\mathcal{A}_{\alpha\beta\gamma\delta})_{\mathbf{r}\times\mathbf{c}}$. Note the multi-indexing of the block rows and columns.

Another MATLAB class developed for this chapter is `bltenmat`, which allows us to form block unfoldings easily. A full description is available in Appendix A. The following code creates an example similar to the one shown above:

```
% Create a 2-by-4-by-4-by-2 block tensor:
A_block = bltensor(rand(5,8,10,7),...
                   {[3 2],[2 2 2 2],[3 4 3],[3 4]});
% Form the [1 3]-by-[2 4] block unfolding:
A_unfold = bltenmat(A_block, [1 3], [2 4]);
```

3.2.4 A Special Case

Returning to the example (3.12), suppose

$$\mathcal{B} \;=\; B^{(1)} \circ \cdots \circ B^{(d)}$$

where

$$B^{(\ell)} \in \mathbb{R}^{q_\ell \times n_\ell}$$

for $\ell = 1, 2, \ldots, d$. Assume that $[\mathbf{u}^{(\ell)}, \mathbf{v}^{(\ell)}]$ is a blocking for $B^{(\ell)}$ and note that

$$\mathbf{M} \;=\; \left\{\mathbf{u}^{(1)}, \mathbf{v}^{(1)}, \ldots, \mathbf{u}^{(d)}, \mathbf{v}^{(d)}\right\} \tag{3.35}$$

is a blocking for $\mathcal{B}$. Let $B^{(\ell)}_{\mu,\tau}$ denote block (μ, τ) of $B^{(\ell)}$. If

$$\mathbf{k} \;=\; [\, i_1, j_1, \ldots, i_d, j_d \,]$$

then the $\mathbf{k}$-th block of $\mathcal{B}$ is given by

$$\mathcal{B}_{\mathbf{k}} \;=\; B^{(1)}_{i_1,j_1} \circ \cdots \circ B^{(d)}_{i_d,j_d}.$$

If

$$\begin{aligned}
\mathbf{r} &= 1{:}2{:}2d \\
\mathbf{c} &= 2{:}2{:}2d \\
\mathbf{R} &= \left\{\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(d)}\right\} && (3.36) \\
\mathbf{C} &= \left\{\mathbf{v}^{(1)}, \ldots, \mathbf{v}^{(d)}\right\}, && (3.37)
\end{aligned}$$

then by applying (3.31) and (3.13) we see that

$$(\mathcal{B}_{\mathbf{R}\times\mathbf{C}})_{\mathbf{i},\mathbf{j}} \;=\; \left(B^{(1)}_{i_1,j_1} \circ \cdots \circ B^{(d)}_{i_d,j_d}\right)_{\mathbf{r}\times\mathbf{c}} \;=\; B^{(d)}_{i_d,j_d} \otimes \cdots \otimes B^{(1)}_{i_1,j_1}. \tag{3.38}$$

Here, the notation $(\mathcal{B}_{\mathbf{R}\times\mathbf{C}})_{\mathbf{i},\mathbf{j}}$ denotes block $(ivec(\mathbf{i}, \mathbf{q}), ivec(\mathbf{j}, \mathbf{n}))$. This result is key to the development of a block-level multilinear product. See §3.3.2.

3.3 Blocked Contractions

We next apply our block tensor "technology" to the problem of computing a contraction between two tensors. We use the multi-index summation notation (1.1) to describe the summations.

3.3.1 The General Case

It is instructive to work through a small, motivating example before we present the main results. Suppose we are given $\mathcal{F} \in \mathbb{R}^{\alpha_1 \times \cdots \times \alpha_4}$ and $\mathcal{G} \in \mathbb{R}^{\beta_1 \times \cdots \times \beta_5}$ and wish to compute the order-5 tensor $\mathcal{H} \in \mathbb{R}^{\alpha_1 \times \alpha_2 \times \beta_3 \times \beta_4 \times \beta_5}$ defined by

$$\mathcal{H}(i_1, i_2, j_1, j_2, j_3) \;=\; \sum_{k_1=1}^{\alpha_3} \sum_{k_2=1}^{\alpha_4} \mathcal{F}(i_1, i_2, k_1, k_2) \cdot \mathcal{G}(k_1, k_2, j_1, j_2, j_3). \tag{3.39}$$

Of course, for this to make sense, we must have $\alpha_3 = \beta_1$ and $\alpha_4 = \beta_2$. A tensor contraction such as this can be "reshaped" into a single matrix-matrix multiplication. To see this we rewrite (3.39) using multi-index notation,

$$\mathcal{H}(\mathbf{i}, \mathbf{j}) \;=\; \sum_{\mathbf{k}=\mathbf{1}}^{\boldsymbol{\alpha}(3:4)} \mathcal{F}(\mathbf{i}, \mathbf{k}) \cdot \mathcal{G}(\mathbf{k}, \mathbf{j}). \tag{3.40}$$

Define the index vectors

$$\mathbf{r} = [\,1\;2\,] \qquad \boldsymbol{\lambda} = [\,3\;4\,] \qquad \boldsymbol{\psi} \;=\; [\,1\;2\,] \qquad \mathbf{c} = [\,3\;4\;5\,]$$

and note that $\mathbf{1} \leq \mathbf{i} \leq \boldsymbol{\alpha}(\mathbf{r})$ and $\mathbf{1} \leq \mathbf{j} \leq \boldsymbol{\beta}(\mathbf{c})$ in (3.40). Recall from (1.17)-(1.20) that the rows and columns of a tensor unfolding are vecs of reduced-order subtensors. In particular

$$\begin{aligned} \mathcal{F}_{\mathbf{r} \times \boldsymbol{\lambda}}(\mathbf{i}, :) &= \operatorname{vec}(\mathcal{F}^{(\mathbf{i})})^T \\ \mathcal{G}_{\boldsymbol{\lambda} \times \mathbf{c}}(:, \mathbf{j}) &= \operatorname{vec}(\mathcal{G}^{(\mathbf{j})}) \end{aligned}$$

where $\mathcal{F}^{(\mathbf{i})} \in \mathbb{R}^{\alpha_1 \times \alpha_2}$ and $\mathcal{G}^{(\mathbf{j})} \in \mathbb{R}^{\beta_3 \times \beta_4 \times \beta_5}$ are defined by

$$\begin{aligned} \mathcal{F}^{(\mathbf{i})}(k_1, k_2) &= \mathcal{F}(i_1, i_2, k_1, k_2) \\ \mathcal{G}^{(\mathbf{j})}(k_1, k_2) &= \mathcal{G}(k_1, k_2, j_1, j_2, j_3). \end{aligned}$$

It follows from (3.40) that

$$\mathcal{H}(\mathbf{i}, \mathbf{j}) \;=\; \sum_{k_1=1}^{\alpha_3} \sum_{k_2=1}^{\alpha_2} \mathcal{F}^{(\mathbf{i})}(k_1, k_2) \cdot \mathcal{G}^{(\mathbf{j})}(k_1, k_2) \;=\; \mathcal{F}_{\mathbf{r} \times \boldsymbol{\lambda}}(\mathbf{i}, :) \cdot \mathcal{G}_{\boldsymbol{\psi} \times \mathbf{c}}(:, \mathbf{j})$$

and thus

$$\mathcal{H}_{[1\ 2] \times [3\ 4\ 5]} \;=\; \mathcal{F}_{\mathbf{r} \times \boldsymbol{\lambda}} \cdot \mathcal{G}_{\boldsymbol{\psi} \times \mathbf{c}}.$$

Using the Matlab Tensor Toolbox, these computations can be performed with the following commands, assumg $\mathcal{H}, \mathcal{F}$ and $\mathcal{G}$ are already defined:

```
r = [1 2]; lambda = [3 4]; psi = [1 2]; c =[3 4 5];
F_unfold = tenmat(F,r,lambda);
G_unfold = tenmat(G,psi,c);
H_unfold = F_unfold * G_unfold;
```

In this example, the summation was over the last two modes of $\mathcal{F}$ and the first two modes of $\mathcal{G}$. These are convenient locations for the summation indices because the contraction $\mathcal{H}$ is then easily seen to be "isomorphic" to a matrix-matrix product of simple tensor unfoldings.

If the summation modes are arbitrarily positioned, then they can be moved to these friendly locations through transposition. This result is widely known and exploited, e.g. [3, 39]. Nevertheless, in keeping with the spirit of this chapter we think that it is useful to include a formal verification of this important maneuver.

Theorem 3.5. *Suppose* $\mathcal{F} \in \mathbb{R}^{\alpha_1 \times \cdots \times \alpha_{f+\ell}}$, $\mathcal{G} \in \mathbb{R}^{\beta_1 \times \cdots \times \beta_{g+\ell}}$, *and that* $\mathbf{p}$ *and* $\mathbf{q}$ *are*

permutations of $1{:}f{+}\ell$ *and* $1{:}g{+}\ell$ *respectively. Define*

$$\boldsymbol{\lambda} \;=\; \mathbf{p}(1{:}\ell) \qquad \mathbf{r} \;=\; \mathbf{p}((\ell{+}1){:}(\ell{+}f))$$

$$\boldsymbol{\psi} \;=\; \mathbf{q}(1{:}\ell) \qquad \mathbf{c} \;=\; \mathbf{q}((\ell{+}1){:}(\ell{+}g))$$

and assume $\boldsymbol{\alpha}(\boldsymbol{\lambda}) = \boldsymbol{\beta}(\boldsymbol{\psi})$. *If* $\mathcal{H} \in \mathbb{R}^{\alpha_{r_1}\times\cdots\times\alpha_{r_f}\times\beta_{c_1}\times\cdots\times\beta_{c_g}}$ *is defined by*

$$\mathcal{H}(\mathbf{i},\mathbf{j}) \;=\; \sum_{\mathbf{k}=\mathbf{1}}^{\boldsymbol{\alpha}(\boldsymbol{\lambda})} \mathcal{F}^{<\mathbf{p}>}(\mathbf{k},\mathbf{i})\,\mathcal{G}^{<\mathbf{q}>}(\mathbf{k},\mathbf{j}) \qquad\qquad \mathbf{1}\le\mathbf{i}\le\boldsymbol{\alpha}(\mathbf{r}),\quad \mathbf{1}\le\mathbf{j}\le\boldsymbol{\beta}(\mathbf{c}), \tag{3.41}$$

then

$$\mathcal{H}_{[1:f]\times[f+1:f+g]} \;=\; \mathcal{F}_{\mathbf{r}\times\boldsymbol{\lambda}}\cdot\,\mathcal{G}_{\boldsymbol{\psi}\times\mathbf{c}} \tag{3.42}$$

Proof. The assumption $\boldsymbol{\alpha}(\boldsymbol{\lambda}) = \boldsymbol{\beta}(\boldsymbol{\psi})$ ensures that the summations in (3.41) are well defined. Using (1.14)-(1.20) we have

$$\mathcal{F}_{\boldsymbol{\lambda}\times\mathbf{r}}(:,\mathbf{i}) \;=\; \mathrm{vec}(\mathcal{F}^{(\mathbf{i})})$$

$$\mathcal{G}_{\boldsymbol{\psi}\times\mathbf{c}}(:,\mathbf{j}) \;=\; \mathrm{vec}(\mathcal{G}^{(\mathbf{j})})$$

where $\mathcal{F}^{(\mathbf{i})} \in \mathbb{R}^{\alpha_{\lambda_1}\times\cdots\times\alpha_{\lambda_\ell}}$ and $\mathcal{G}^{(\mathbf{j})} \in \mathbb{R}^{\beta_{\psi_1}\times\cdots\times\beta_{\psi_\ell}}$ are defined by

$$\mathcal{F}^{(\mathbf{i})}(\mathbf{k}) \;=\; \mathcal{F}^{<\mathbf{p}>}(k_1,\ldots,k_\ell,i_1,\ldots,i_f)$$

$$\mathcal{G}^{(\mathbf{j})}(\mathbf{k}) \;=\; \mathcal{G}^{<\mathbf{q}>}(k_1,\ldots,k_\ell,j_1,\ldots,j_g).$$

It follows that for all $\mathbf{i}$ and $\mathbf{j}$ that satisfy $\mathbf{1}\le\mathbf{i}\le\boldsymbol{\alpha}(\mathbf{r})$ and $\mathbf{1}\le\mathbf{j}\le\boldsymbol{\beta}(\mathbf{c})$ we have

$$\begin{aligned}\mathcal{H}(\mathbf{i},\mathbf{j}) \;&=\; \sum_{\mathbf{k}=\mathbf{1}}^{\boldsymbol{\alpha}(\boldsymbol{\lambda})} \mathcal{F}^{<\mathbf{p}>}(\mathbf{k},\mathbf{i})\mathcal{G}^{<\mathbf{q}>}(\mathbf{k},\mathbf{j}) \\ &=\; \sum_{\mathbf{k}=\mathbf{1}}^{\boldsymbol{\alpha}(\boldsymbol{\lambda})} \mathcal{F}^{(\mathbf{i})}(\mathbf{k})\mathcal{G}^{(\mathbf{j})}(\mathbf{k}) \;=\; \mathcal{F}_{\mathbf{r}\times\boldsymbol{\lambda}}(:,\mathbf{i})\cdot\mathcal{G}_{\boldsymbol{\psi}\times\mathbf{c}}(:,\mathbf{j})\end{aligned}$$

which, using (1.14)-(1.20), implies (3.42), completing the proof. □

As another example, consider the case where $\mathcal{F} \in \mathbb{R}^{\alpha_1\times\cdots\times\alpha_5}$ and $\mathcal{G} \in \mathbb{R}^{\beta_1\times\beta_2}$ with $\alpha_2 = \beta_2$, $\alpha_3 = \beta_1$ and the tensor $\mathcal{H} \in \mathbb{R}^{\alpha_5\times\alpha_1\times\alpha_4}$ is given by the contraction

$$\mathcal{H}(i_1,i_2,i_3) \;=\; \sum_{\mathbf{k}=\mathbf{1}}^{\boldsymbol{\alpha}(2:3)} \mathcal{F}(i_2,k_1,k_2,i_3,i_1)\mathcal{G}(k_2,k_1).$$

In the notation of Theorem 3.5 we have $f = 3, \ell = 2, g = 0$ and the permutations $\mathbf{p}$ and $\mathbf{q}$ are

$$\mathbf{p} \;=\; [\,5\;1\;4\;2\;3\,] \quad \text{and} \quad \mathbf{q} \;=\; [\,2\;1\,].$$

Therefore

$$\mathbf{r} \;=\; [\,5\;1\;4\,] \qquad \mathbf{c} \;=\; \emptyset \qquad \boldsymbol{\lambda} \;=\; [\,2\;3\,] \qquad \boldsymbol{\psi} \;=\; [\,2\;1\,]$$

and so by (4.4)

$$\begin{aligned} \mathcal{H}_{[1:3]\times\emptyset} \;=\; \mathrm{vec}(\mathcal{H}) \;&=\; \mathcal{F}_{[5\;1\;4]\times[2\;3]} \cdot \mathcal{G}_{[2\;1]\times\emptyset} \\ &=\; \mathcal{F}_{[5\;1\;4]\times[2\;3]} \cdot \mathrm{vec}(\mathcal{G}^T). \end{aligned}$$

If the tensors $\mathcal{F}$ and $\mathcal{G}$ are "blocked conformally", then (3.41) can be reformulated as a product of two block matrices.

Corollary 3.6. *Assume that the notation and conditions of the Theorem 3.5 hold. Let*

$$\mathbf{S} \;=\; \{\mathbf{s}^{(1)}, \ldots, \mathbf{s}^{(f+\ell)}\} \tag{3.43}$$

be a blocking for $\mathcal{F}$ and set

$$\mathbf{R} \;=\; \{\mathbf{s}^{(r_1)}, \ldots, \mathbf{s}^{(r_f)}\} \qquad \boldsymbol{\Lambda} \;=\; \{\mathbf{s}^{(\lambda_1)}, \ldots, \mathbf{s}^{(\lambda_\ell)}\}.$$

Likewise, let

$$\mathbf{T} \;=\; \{\mathbf{t}^{(1)}, \ldots, \mathbf{t}^{(g+\ell)}\} \tag{3.44}$$

be a blocking for $\mathcal{G}$ and set

$$\boldsymbol{\Psi} \;=\; \{\mathbf{t}^{(\psi_1)}, \ldots, \mathbf{t}^{(\psi_\ell)}\} \qquad \mathbf{C} \;=\; \{\mathbf{t}^{(c_1)}, \ldots, \mathbf{t}^{(c_g)}\}.$$

If

$$\mathbf{s}^{(\lambda_k)} = \mathbf{t}^{(\psi_k)} \qquad k = 1, 2, \ldots, \ell \tag{3.45}$$

then with respect to the tensor $\mathcal{H}$, $\mathbf{R}$ *is a blocking for modes 1 through* f, $\mathbf{C}$ *is a blocking for modes* $f+1$ *through* $f+g$, *and*

$$\mathcal{H}_{\mathbf{R}\times\mathbf{C}} \;=\; \mathcal{F}_{\mathbf{R}\times\mathbf{\Lambda}} \cdot \mathcal{G}_{\mathbf{\Psi}\times\mathbf{C}}. \tag{3.46}$$

Proof. From Theorem 3.4 we have

$$\begin{aligned} \mathcal{F}_{\mathbf{R}\times\mathbf{\Lambda}} &= P_{\mathbf{R}} \mathcal{F}_{\mathbf{r}\times\boldsymbol{\lambda}} P_{\mathbf{\Lambda}}^T \\ \mathcal{G}_{\mathbf{\Psi}\times\mathbf{C}} &= P_{\mathbf{\Psi}} \mathcal{G}_{\boldsymbol{\psi}\times\mathbf{c}} P_{\mathbf{C}}^T. \end{aligned}$$

Since $\{\mathbf{s}^{(r_1)}, \ldots, \mathbf{s}^{(r_f)}, \mathbf{t}^{(c_1)}, \ldots, \mathbf{t}^{(c_g)}\}$ is a blocking for $\mathcal{H}$ we also have

$$\mathcal{H}_{\mathbf{R}\times\mathbf{C}} \;=\; P_{\mathbf{R}} \cdot \mathcal{H}_{[1:f]\times[f+1:f+g]} \cdot P_{\mathbf{C}}^T.$$

The conformability condition (3.45) implies $P_{\mathbf{\Lambda}} = P_{\mathbf{\Psi}}$ and so it follows from (3.42) that

$$\begin{aligned} \mathcal{H}_{\mathbf{R}\times\mathbf{C}} &= P_{\mathbf{R}} (\mathcal{F}_{\mathbf{r}\times\boldsymbol{\lambda}} \cdot \mathcal{G}_{\boldsymbol{\psi}\times\mathbf{c}}) P_{\mathbf{C}}^T \\ &= (P_{\mathbf{R}} \mathcal{F}_{\mathbf{r}\times\boldsymbol{\lambda}} P_{\mathbf{\Lambda}}^T)(P_{\mathbf{\Psi}} \mathcal{G}_{\boldsymbol{\psi}\times\mathbf{c}} P_{\mathbf{C}}^T) \;=\; \mathcal{F}_{\mathbf{R}\times\mathbf{\Lambda}} \cdot \mathcal{G}_{\mathbf{\Psi}\times\mathbf{C}} \end{aligned}$$

completing the proof. □

Thus, the tensor $\mathcal{H}$ in (3.41) can be computed as either a matrix product (3.42) or as a block matrix product (3.46). For the latter case, we develop recipes for the blocks of $\mathcal{H}_{\mathbf{R}\times\mathbf{C}}$. Let $b_j^{(\mathbf{S})}$ be the length of the blocking vector $\mathbf{s}^{(j)}$ in (3.43) and let $b_j^{(\mathbf{T})}$ be the length of the blocking vector $\mathbf{t}^{(j)}$ in (3.44). Note that if

$$\begin{aligned} b_{rows}^{(\mathcal{F})} &= b_{r_1}^{(\mathbf{S})} \cdots b_{r_f}^{(\mathbf{S})} \qquad b_{cols}^{(\mathcal{F})} = b_{\lambda_1}^{(\mathbf{S})} \cdots b_{\lambda_\ell}^{(\mathbf{S})} \\ b_{rows}^{(\mathcal{G})} &= b_{\psi_1}^{(\mathbf{T})} \cdots b_{\psi_\ell}^{(\mathbf{T})} \qquad b_{cols}^{(\mathcal{G})} = b_{c_1}^{(\mathbf{T})} \cdots b_{c_g}^{(\mathbf{T})} \end{aligned}$$

then (3.45) implies $b_{cols}^{(\mathcal{F})} = b_{rows}^{(\mathcal{G})}$ and we observe that

$$\left\{ \begin{array}{c} \mathcal{F}_{\mathbf{R}\times\mathbf{\Lambda}} \\ \mathcal{G}_{\mathbf{\Lambda}\times\mathbf{C}} \\ \mathcal{H}_{\mathbf{R}\times\mathbf{C}} \end{array} \right\} \quad \text{is a} \quad \left\{ \begin{array}{c} b_{rows}^{(\mathcal{F})}\text{-by-}b_{cols}^{(\mathcal{F})} \\ b_{rows}^{(\mathcal{G})}\text{-by-}b_{cols}^{(\mathcal{G})} \\ b_{rows}^{(\mathcal{F})}\text{-by-}b_{cols}^{(\mathcal{G})} \end{array} \right\} \quad \text{block matrix.}$$

If we have $\mathbf{1} \leq \boldsymbol{\mu} \leq \mathbf{b}^{(\mathbf{S})}(\mathbf{r})$ and $\mathbf{1} \leq \boldsymbol{\tau} \leq \mathbf{b}^{(\mathbf{T})}(\mathbf{c})$, $\mu = ivec(\boldsymbol{\mu}, \mathbf{b}^{(\mathbf{S})}(\mathbf{r}))$ and $\tau = ivec(\boldsymbol{\tau}, \mathbf{b}^{(\mathbf{T})}(\mathbf{c}))$, then block (μ, τ) of $\mathcal{H}_{\mathbf{R}\times\mathbf{C}}$ is given by

$$(\mathcal{H}_{\mathbf{R}\times\mathbf{C}})_{\boldsymbol{\mu},\boldsymbol{\tau}} \;=\; \sum_{\mathbf{q}=\mathbf{1}}^{\mathbf{b}^{(\mathbf{S})}(\boldsymbol{\lambda})} (\mathcal{F}_{\mathbf{R}\times\boldsymbol{\Lambda}})_{\boldsymbol{\mu},\mathbf{q}} \, (\mathcal{G}_{\boldsymbol{\Psi}\times\mathbf{C}})_{\mathbf{q},\boldsymbol{\tau}} \,.$$

Using (3.31) this can be rewritten in terms of subtensor unfoldings. Indeed, if index vectors $\mathbf{k}$, $\mathbf{i}^{(\mathbf{q})}$, and $\mathbf{j}^{(\mathbf{q})}$ are defined by

$$\begin{aligned} \mathbf{k}(\mathbf{r}) &= \boldsymbol{\mu} & \mathbf{k}(\mathbf{c}) &= \boldsymbol{\tau} \\ \mathbf{i}^{(\mathbf{q})}(\mathbf{r}) &= \mathbf{k}(\mathbf{r}) & \mathbf{i}^{(\mathbf{q})}(\boldsymbol{\lambda}) &= \mathbf{q} \\ \mathbf{j}^{(\mathbf{q})}(\boldsymbol{\psi}) &= \mathbf{q} & \mathbf{j}^{(\mathbf{q})}(\mathbf{c}) &= \mathbf{k}(\mathbf{c}) \end{aligned}$$

then

$$(\mathcal{H}_{\mathbf{k}})_{[1:f]\times[f+1:f+g]} \;=\; \sum_{\mathbf{q}=\mathbf{1}}^{\mathbf{b}^{(\mathbf{S})}(\boldsymbol{\lambda})} (\mathcal{F}_{\mathbf{i}^{(\mathbf{q})}})_{\mathbf{r}\times\boldsymbol{\lambda}} \left(\mathcal{G}_{\mathbf{j}^{(\mathbf{q})}}\right)_{\boldsymbol{\psi}\times\mathbf{c}} . \tag{3.47}$$

As an example, suppose the tensor $\mathcal{H} \in \mathbb{R}^{12\times8\times9}$ is given by the contraction

$$\mathcal{H}(i_1, i_2, j_1) \;=\; \sum_{\mathbf{k}=\mathbf{1}}^{[4\ 10]} \mathcal{F}(k_1, i_1, i_2, k_2) \cdot \mathcal{G}(k_1, j_1, k_2)$$

where $\mathcal{F} \in \mathbb{R}^{4\times12\times8\times10}$ and $\mathcal{G} \in \mathbb{R}^{4\times9\times10}$ have the blockings

$$\begin{aligned} \mathbf{S} &= \{[2\ \ 2], [6\ \ 3\ \ 3], [2\ \ 4\ \ 2], [3\ \ 2\ \ 5]\}, \\ \mathbf{T} &= \{[2\ \ 2], [3\ \ 5\ \ 1], [3\ \ 2\ \ 5]\}, \end{aligned}$$

respectively. Then

$$\mathbf{M} \;=\; \{[6\ \ 3\ \ 3], [2\ \ 4\ \ 2], [3\ \ 5\ \ 1]\}$$

is the corresponding blocking of $\mathcal{H}$ and in the notation of Theorem 3.5 and Corollary 3.6 we have $\mathbf{r} = [\,2\ 3\,], \mathbf{c} = [\,2\,], \boldsymbol{\lambda} = [\,1\ 4\,]$ and $\boldsymbol{\psi} = [\,1\ 3\,]$ and the corresponding modal blockings from (3.46) are

$$\begin{aligned} \mathbf{R} &= \{[6\ \ 3\ \ 3], [2\ \ 4\ \ 2]\}, & \boldsymbol{\Lambda} &= \{[2\ \ 2], [3\ \ 2\ \ 5]\}, \\ \mathbf{C} &= \{[3\ \ 5\ \ 1]\}, & \boldsymbol{\Psi} &= \{[2\ \ 2], [3\ \ 2\ \ 5]\}. \end{aligned}$$

Using the blocked extension of the Matlab Tensor Toolbox the following commands form the block unfoldings $\mathcal{F}_{\mathbf{R}\times\mathbf{\Lambda}}$ and $\mathcal{G}_{\mathbf{\Psi}\times\mathbf{C}}$ and perform the contraction of this example on randomly generated tensors $\mathcal{F}$ and $\mathcal{G}$:

```
S = {[2 2],[6 3 3],[2 4 2],[3 2 4]};
T = {[2 2],[3 5 1],[3 2 5]};
r = [2 3]; c = 2; lambda = [1 4]; psi = [1 3];
F = bltensor(rand(4,12,8,10),S);
G = bltensor(rand(4,9,10),T);
F_blunfold = bltenmat(F,r,lambda);
G_blunfold = bltenmat(G,c,psi);
H_blunfold = F_blunfold * G_blunfold;
```

After the computations are performed, the matrix `H_blunfold` will contain $\mathcal{H}_{\mathbf{R}\times\mathbf{C}}$.

3.3.2 Blocked Multilinear Products

As an example of how the preceding results can be adapted to handle structured contractions, we briefly consider the multilinear product since we have developed the supporting formulae in §3.1.2 and §3.2.4. Suppose $\mathcal{A} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ and that

$$B^{(k)} \in \mathbb{R}^{q_k\times n_k} \qquad k = 1,\ldots,d.$$

Let the tensor $\mathcal{C} \in \mathbb{R}^{q_1\times\cdots\times q_d}$ be specified by

$$\mathcal{C} \;=\; (B^{(1)},\ldots,B^{(d)})\cdot\mathcal{A}.$$

If the order-$(2d)$ tensor $\mathcal{B}$ is defined by

$$\mathcal{B} \;=\; B^{(1)}\circ\cdots\circ B^{(d)}$$

then we see that $\mathcal{C}$ is a contraction of the form

$$\mathcal{C}(\mathbf{i}) \;=\; \sum_{\mathbf{k=1}}^{\mathbf{n}} \mathcal{A}(\mathbf{k})\mathcal{B}(i_1, k_1, \ldots, i_d, k_d).$$

We apply Theorem 3.5 with $\mathcal{F} = \mathcal{B}$, $\mathcal{G} = \mathcal{A}$, $\ell = d$, $\mathbf{r} = 1{:}2{:}2d$, $\boldsymbol{\lambda} = 2{:}2{:}2d$, $\boldsymbol{\psi} = 1{:}d$ and $\mathbf{c} = \emptyset$. It follows that $\mathcal{A}_{\boldsymbol{\Psi}\times\mathbf{c}} = \mathrm{vec}(\mathcal{A})$ and $\mathcal{C}_{[1:f]\times[f+1:f+g]} = \mathcal{C}_{[1:\ell]\times[\ell+1:\ell]} = \mathrm{vec}(\mathcal{C})$. It then follows from Theorem 3.5 and (3.13) that

$$\mathrm{vec}(\mathcal{C}) \;=\; \left(B^{(d)} \otimes \cdots \otimes B^{(1)}\right) \mathrm{vec}(\mathcal{A}). \tag{3.48}$$

If the B matrices are blocked according to (3.35) and $\mathbf{R}$ and $\mathbf{C}$ are defined by (3.36)-(3.37), then $\mathbf{R}$ is a blocking for $\mathcal{C}$, $\mathbf{C}$ is a blocking for $\mathcal{A}$ and

$$P_{\mathbf{R}}\mathrm{vec}(\mathcal{C}) \;=\; \left(P_{\mathbf{R}} \left(B^{(d)} \otimes \cdots \otimes B^{(1)}\right) P_{\mathbf{C}}^T\right) P_{\mathbf{C}}\mathrm{vec}(\mathcal{A}). \tag{3.49}$$

From (3.38) we see that the matrix

$$\mathcal{B}_{\mathbf{R}\times\mathbf{C}} \;=\; P_{\mathbf{R}} \left(B^{(d)} \otimes \cdots \otimes B^{(1)}\right) P_{\mathbf{C}}^T \tag{3.50}$$

is a block matrix whose entries are Kronecker products. Indeed, $\mathcal{B}_{\mathbf{R}\times\mathbf{C}}$ is essentially the Tracy-Singh product of the B-matrices, see [68]. Thus, from (3.48)-(3.50) we have the following block specification for $\mathcal{C}$:

$$\mathrm{vec}_{\mathbf{R}}(\mathcal{C}) \;=\; \mathcal{B}_{\mathbf{R}\times\mathbf{C}}\mathrm{vec}_{\mathbf{C}}(\mathcal{A}). \tag{3.51}$$

Example: If $\mathcal{C} = (G, F, E) \cdot \mathcal{A}$, tensor $\mathcal{A}$ is a $2 \times 2 \times 2$ block tensor and the matrices E, F and G are 2×2 block matrices,

$$E = \begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix},\; F = \begin{bmatrix} F_{11} & F_{12} \\ F_{21} & F_{22} \end{bmatrix},\; G = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}.$$

Then the matrix $P_{\mathbf{R}}(E \otimes F \otimes G)P_{\mathbf{C}}^T$ is given by

$$\begin{bmatrix} E \otimes F_{11} \otimes G_{11} & E \otimes F_{12} \otimes G_{11} & E \otimes F_{11} \otimes G_{12} & E \otimes F_{12} \otimes G_{12} \\ E \otimes F_{21} \otimes G_{11} & E \otimes F_{22} \otimes G_{11} & E \otimes F_{21} \otimes G_{12} & E \otimes F_{22} \otimes G_{12} \\ E \otimes F_{11} \otimes G_{21} & E \otimes F_{12} \otimes G_{21} & E \otimes F_{11} \otimes G_{22} & E \otimes F_{12} \otimes G_{22} \\ E \otimes F_{21} \otimes G_{21} & E \otimes F_{22} \otimes G_{21} & E \otimes F_{21} \otimes G_{22} & E \otimes F_{22} \otimes G_{22} \end{bmatrix}.$$

3.3.3 Visualization

As is the case in block matrix computations, it is sometimes important to view a given blocked tensor contraction from different viewpoints. We offer a small example to build an appreciation for this point.

Suppose $\mathcal{F}$ is a $3 \times 4 \times 2$ block tensor and $\mathcal{G}$ is a $2 \times 3 \times 5$ block tensor such that the blockings in mode 3 in $\mathcal{F}$ and mode 1 in $\mathcal{G}$ conform. Let $\mathcal{H}$ be the $3 \times 4 \times 3 \times 5$ block tensor whose elements are given by

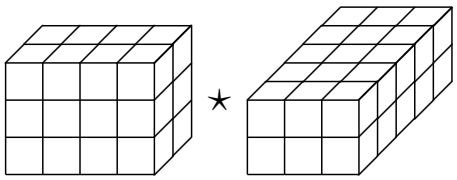
$$\mathcal{H}(i_1, i_2, j_1, j_2) \;=\; \sum_k \mathcal{F}(i_1, i_2, k) \cdot \mathcal{G}(k, j_1, j_2).$$

For convenience, denote the operation of contracting two third order tensors $\mathcal{T}_1$ and $\mathcal{T}_2$ in this way (i.e. over mode 3 and mode 1, respectively) as $\mathcal{T}_1 \star \mathcal{T}_2$. Figure 3.3 shows how this blocked contraction can be visualized in three different ways.

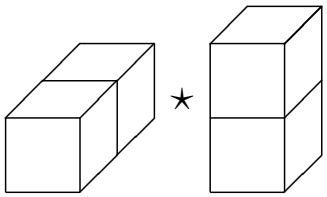
Ultimately, in terms of unfoldings, block $[a, b, c, d]$ in $\mathcal{H}$ can be computed through the matrix equation

$$(\mathcal{H}_{abcd})_{[1\,2]\times[3\,4]} = (\mathcal{F}_{ab1})_{[1\,2]\times[3]} \cdot (\mathcal{G}_{1cd})_{[1]\times[2\,3]} \;+\; (\mathcal{F}_{ab2})_{[1\,2]\times[3]} \cdot (\mathcal{G}_{2cd})_{[1]\times[2\,3]},$$

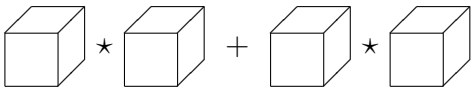
which is a result of (3.47) and is the unfolded version of part (3) of Figure 3.3.



(1) The tensor contraction $\mathcal{H} = \mathcal{F} \star \mathcal{G}$ of two order-3 tensors viewed graphically as a contraction of conformally blocked tensors.



(2) A particular block $\mathcal{H}_{abcd} = \mathcal{H}(\alpha_1{:}\alpha_2, \beta_1{:}\beta_2, \gamma_1{:}\gamma_2, \delta_1{:}\delta 2)$ is a $\star$-contraction of two "block fibers", one from $\mathcal{F}$ and one from $\mathcal{G}$, i.e. $\mathcal{H}_{abcd} = \mathcal{F}(\alpha_1{:}\alpha_2, \beta_1{:}\beta_2, :) \star \mathcal{G}(:, \gamma_1{:}\gamma_2, \delta_1{:}\delta_2)$.



(3) The $\star$-contraction of the two block fibers is itself a sum of $\star$-contractions of fibers' block components, i.e. $\mathcal{H}_{abcd} = \mathcal{F}_{ab1} \star \mathcal{G}_{1cd} + \mathcal{F}_{ab2} \star \mathcal{G}_{2cd}$.

Figure 3.3: Visualizing a Blocked Contraction

3.4 Notes on Implementation

Given the nature of this chapter, it is important to be reminded in this closing section that there is a big difference between a cryptic mathematical formula and its utilization in practice. We review a few issues that are key to efficient implementation of block tensor algorithms and block unfoldings.

3.4.1 Permutation Vectors

The permutation matrix $P_{\mathbf{M}}$ that is characterized in Theorem 3.2 should never be computed as a two-dimensional array. Rather, an integer vector should be used to represent the matrix. We offer a few details based on the convention that if $P = I_n(:, \mathbf{v})$ where $\mathbf{v}$ is permutation of $1{:}n$, then $\mathbf{v}$ represents P. We capture this connection with the notation $P_{\mathbf{v}}$. Note that if $y = P_{\mathbf{v}}x$, then $y = x(\mathbf{v})$ while $y(\mathbf{v}) = x$ implies $y = P_{\mathbf{v}}^T x$. Here are some basic facts that concern this representation:

1. If q and r are positive integers and $\mathbf{w} = [1{:}r{:}qr \;\; 2{:}2{:}qr \;\; \cdots \;\; r{:}r{:}qr]$, then $P_{\mathbf{w}} = \Pi_{q,r}$, the (q, r) perfect shuffle. This can be easily implemented in MATLAB with the command

   ```
   w = reshape(reshape(1:q*r,r,q)',1,q*r);
   ```

2. If $\mathbf{u}$ and $\mathbf{v}$ are permutations of $1{:}n$ and $\mathbf{w} = \mathbf{v}(\mathbf{u})$, then $P_{\mathbf{w}} = P_{\mathbf{u}}P_{\mathbf{v}}$.

3. If $\mathbf{u}$ is a permutation of $1{:}n$ and $\mathbf{v}$ is a permutation of $1{:}m$, then $P_{\mathbf{w}} = P_{\mathbf{u}} \otimes P_{\mathbf{v}}$ where $\mathbf{w} = \mathbf{1}_n \otimes \mathbf{v} \; + \; m \cdot (\mathbf{u} - \mathbf{1}_n) \otimes \mathbf{1}_m$.

4. If $\mathbf{u}$ is a permutation of $1{:}n$ and $\mathbf{v}$ is a permutation of $1{:}m$, then $P_{\mathbf{w}} = \mathrm{diag}(P_{\mathbf{u}}, P_{\mathbf{v}})$ where $\mathbf{w} = [\mathbf{u},\ n\cdot\mathbf{1}_m + \mathbf{v}]$

The construction of a vector representation for the matrix $P_{\mathbf{M}}$ involves the systematic use of these properties. A recursive implementation can be based on (3.30): Suppose $\hat{\mathbf{v}}$ represents $P_{\widehat{\mathbf{M}}}$. A vector $\mathbf{g}$ representing $\Gamma^{(d)}$ can be constructed from (3.24)-(3.25) and facts 1, 2 and 4 above. Then

$$\mathbf{v} \;=\; \left(\mathbf{1}_{n_d} \otimes \hat{\mathbf{v}} + N_{d-1}((1{:}n_d) - \mathbf{1}_{n_d}) \otimes \mathbf{1}_{N_{d-1}}\right)(\mathbf{g})$$

represents $P_{\mathbf{M}}$. The Matlab function `P_mat_vec` implements this approach. For example, if $\mathcal{A} \in \mathbb{R}^{10\times16\times8\times5}$ and

$$\mathbf{M} = \{[3\ \ 2\ \ 5], [2\ \ 7\ \ 7], [5\ \ 3], [2\ \ 2\ \ 1]\}, \quad \mathbf{r} = [2\ \ 4], \quad \mathbf{c} = [1\ \ 3],$$

then the following commands generate vectors $v_{\mathbf{r}}$ and $v_{\mathbf{c}}$ that represent the $P_{\mathbf{r}}$ and $P_{\mathbf{c}}$ matrices:

```
M = {[3 2 5],[2 7 7],[5 3],[2 2 1]};
r = [2 4]; c = [1 3];
v_r = P_mat_vec(M,r);
v_c = P_mat_vec(M,c);
```

3.4.2 Multilinear Products

A practical computation of a multilinear product $\mathcal{C} = (B^{(1)}, \ldots, B^{(d)})\cdot\mathcal{A}$ would not use the equation (3.48). Instead it would sequentially perform the multiplications

$$\mathcal{A} \;\leftarrow\; \mathcal{A} \times_i B^{(i)}$$

for all modes $i = 1, \ldots, d$, where $\mathcal{A} \times_i B^{(i)} \equiv (I_{n_1}, \ldots, B^{(i)}, \ldots, I_{n_d}) \cdot \mathcal{A}$ is called the *i-mode product* [20, 41]. By using Theorem 3.5 we see that this is equivalent to the matrix-matrix multiplications

$$\mathcal{A}_{(i)} \leftarrow B^{(i)} \mathcal{A}_{(i)}$$

Similarly, in a block-based implementation of the multilinear product, one would not directly use (3.51), but instead sequentially perform the block-matrix multiplications

$$\mathcal{A}_{\mathbf{J} \times \mathbf{C}} \leftarrow B^{(i)} \mathcal{A}_{\mathbf{I} \times \mathbf{C}}$$

for all modes $i = 1, \ldots, d$, where $\mathbf{I}$ is the original blocking for mode i, $\mathbf{J}$ is the new blocking of mode i inherited from the row blocking of $B^{(i)}$ and $\mathbf{C}$ is a blocking for modes $[1{:}i{-}1 \;\; i{+}1{:}d]$ of $\mathcal{A}$.

3.5 Applications of Block Unfoldings

As we have seen, it is possible to reshape a wide class of tensor contractions in a standard way to an equivalent matrix multiplication problem [3, 39]. Implementations of such formulas would of course use block matrix technology, as it is standard for large matrix problems. These techniques form unfoldings such as $\mathcal{A}_{\mathbf{r} \times \mathbf{c}}$ and then block these matrices according to what type of computer architecture is being used. However, such blockings have no real meaning in terms of the underlying tensor. In this chapter we have seen how we can reorder the matrix multiplication to reflect a tensor-level blocking strategy. This could be considered a more natural approach. For example, the visualization of subtasks shown in Figure 3.3 is not possible for the standard method.

The performance of block tensor based algorithms is dependent on implementation details such as cache locality and memory traffic. Perhaps the most interesting application of block tensor computations would be in parallel implementations of algorithms that perform operations only on single blocks or small groups of blocks. In such situations, each processor would only require access to a handful of blocks of the tensor, allowing for a high degree of parallelization and small communication overhead. Such algorithms have already started to appear and have met with some success [56, 57, 74, 78]. In our view, block unfoldings are a natural fit for such algorithms and we hope that the framework presented here will facilitate the development of further such algorithms.

3.5.1 Block Unfoldings and the CP-Decomposition

To illustrate how block unfoldings can be used for algorithm development, consider the block-based algorithm for computing a CP-decomposition by Phan and Cichocki [56]. The problem statement is as follows. For a tensor $\mathcal{Y} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and given $R > 0$ we wish to minimize $\| \mathcal{Y} - \hat{\mathcal{Y}} \|_F$ where

$$\hat{\mathcal{Y}} \;=\; \sum_{i=1}^{R} a_i^{(1)} \circ \cdots \circ a_i^{(d)} \tag{3.52}$$

and $a_i^{(k)} \in \mathbb{R}^{n_k}$ for all $i = 1, \ldots, R$ and $k = 1, \ldots, d$. Define the matrices

$$A^{(k)} \;=\; [\, a_1^{(k)} \;\; \cdots \;\; a_R^{(k)} \,] \;\in \mathbb{R}^{n_k \times R}.$$

Using the notation introduced in [38] we write (3.52) as $\hat{\mathcal{Y}} \;=\; [\![A^{(1)}, \ldots, A^{(d)}]\!]$. The CP approximation problem can then be succinctly stated as

$$\min_{A^{(1)}, \ldots, A^{(d)}} \| \mathcal{Y} - [\![A^{(1)}, \ldots, A^{(d)}]\!] \|_F. \tag{3.53}$$

The standard way to find an approximate solution to (3.53) is to use an iterative alternating strategy: in every iteration we sequentially solve the d linear optimization problems

$$\min_{A^{(k)}} \| \mathcal{Y} - [\![A^{(1)}, \ldots, A^{(d)}]\!] \|_F \tag{3.54}$$

for $k = 1, \ldots, d$. As shown in [38, 42] the solution to (3.54) using standard linear least squares techniques is

$$A^{(k)} \;=\; \mathcal{Y}_{(k)}(\mathcal{Z}_{(k)})^T \tag{3.55}$$

where $\mathcal{Z} = [\![A^{(1)}, \ldots, A^{(k-1)}, V_k^\dagger, A^{(k+1)}, \ldots, A^{(d)}]\!] \in \mathbb{R}^{n_1 \times \cdots \times n_{k-1} \times R \times n_{k+1} \times \cdots \times n_d}$ and $V_k^\dagger$ is the Moore-Penrose pseudoinverse of the symmetric $R \times R$ matrix

$$V_k = ((A^{(d)})^T A^{(d)}) * \cdots * ((A^{(k+1)})^T A^{(k+1)}) * ((A^{(k-1)})^T A^{(k-1)}) * \cdots * ((A^{(1)})^T A^{(1)}). \tag{3.56}$$

Note the tensor $\mathcal{Z}$ depends on k but we will omit the subscript to avoid confusion. The unfolding $\mathcal{Z}_{(k)}$ can be written in terms of the Khatri-Rao product [38] as defined in (1.31):

$$\mathcal{Z}_{(k)} \;=\; V_k^\dagger (A^{(d)} \odot \cdots \odot A^{(k+1)} \odot A^{(k-1)} \odot \cdots \odot A^{(1)})^T \in \mathbb{R}^{R \times n_1 n_2 \cdots n_{k-1} n_{k+1} \cdots n_d}.$$

Suppose $\mathbf{M} = \{\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(d)}\}$ where $\mathbf{m}^{(i)} \in \mathbb{R}^{b_i}$ is a blocking for $\mathcal{Y}$ and $\hat{\mathcal{Y}}$ viewed as $b_1 \times \cdots \times b_d$ block tensors. This is equivalent to blocking the rows of $A^{(k)}$ by $\mathbf{m}^{(k)}$ for $k = 1, \ldots, d$. Define the matrix blocks $A_j^{(k)} \in \mathbb{R}^{m_j^{(k)} \times R}$ with

$$A_j^{(k)} \;\equiv\; A^{(k)}(\ell_j^{(k)} : u_j^{(k)}, :)$$

where $\boldsymbol{\ell}^{(1)}, \ldots, \boldsymbol{\ell}^{(d)}$ and $\mathbf{u}^{(1)}, \ldots, \mathbf{u}^{(d)}$ are given by (3.17)-(3.18). Let $\mathcal{Z}$ be blocked by $\widehat{\mathbf{M}} = \{\mathbf{m}^{(1)}, \ldots, \mathbf{m}^{(k-1)}, [R], \mathbf{m}^{(k+1)}, \ldots, \mathbf{m}^{(d)}\}$, which means that $\mathcal{Z}$ is blocked identically to $\mathcal{Y}$ in every mode except in mode k, which is one big block (i.e., unblocked) in $\mathcal{Z}$.

The matrix multiplication (3.55) is equivalent to contracting $\mathcal{Y}$ and $\mathcal{Z}$ along all modes except mode k, and these modes are blocked conformally by the above construction. It is an easy exercise to show that for $\mathbf{1} \leq \mathbf{j} \leq [\mathbf{b}(1{:}k{-}1),\ 1,\ \mathbf{b}(k{+}1{:}d)]$ we have that the $\mathbf{j}$th block in $\mathcal{Z}$ is given by

$$\mathcal{Z}_{\mathbf{j}} \ = \ [\![A^{(1)}_{j_1}, \ldots, A^{(k-1)}_{j_{k-1}}, V_k^{\dagger}, A^{(k+1)}_{j_{k+1}}, \ldots, A^{(d)}_{j_d}]\!]$$

and thus

$$(\mathcal{Z}_{\mathbf{j}})_{(k)} \ = \ V_k^{\dagger}(A^{(d)}_{j_d} \odot \cdots \odot A^{(k+1)}_{j_{k+1}} \odot A^{(k-1)}_{j_{k-1}} \odot \cdots \odot A^{(1)}_{j_1})^T.$$

If $\mathbf{K} = \mathbf{M}(k) = \{\mathbf{m}^{(k)}\}$, $\widehat{\mathbf{K}} = \widehat{\mathbf{M}}(k) = \{[R]\}$ and $\mathbf{\Lambda} = \mathbf{M}([1{:}k{-}1,\ k{+}1{:}d])$ then by Corollary 3.6 we can restate (3.55) in terms of block unfoldings:

$$A^{(k)} \ = \ \mathcal{Y}_{\mathbf{K}\times\mathbf{\Lambda}}(\mathcal{Z}_{\widehat{\mathbf{K}}\times\mathbf{\Lambda}})^T. \tag{3.57}$$

The blocks in the block matrix $\mathcal{Y}_{\mathbf{K}\times\mathbf{\Lambda}}$ are the unfoldings $(\mathcal{Y}_{\mathbf{j}})_{(k)}$ and the blocks in $\mathcal{Z}_{\widehat{\mathbf{K}}\times\mathbf{\Lambda}}$ are the unfoldings $(\mathcal{Z}_{\mathbf{i}})_{(k)}$. From (3.57) we can thus easily deduce the block identity

$$A^{(k)}_{j_k} \ = \ \sum_{\substack{j_1,\ldots,j_{k-1} \\ j_{k+1},\ldots,j_d}} (\mathcal{Y}_{\mathbf{j}})_{(k)}(A^{(d)}_{j_d} \odot \cdots \odot A^{(k+1)}_{j_{k+1}} \odot A^{(k-1)}_{j_{k-1}} \odot \cdots \odot A^{(1)}_{j_1})V_k^{\dagger}.$$

Noting that $(A^{(i)})^T A^{(i)} \ = \ \sum_{j=1}^{b_i} (A^{(i)}_j)^T A^{(i)}_j$ and substituting into (3.56) gives us equation (6) in [56]. Phan and Cichocki derive this identity by gradient descent rather than directly through the block techniques we have shown above. Although we have gone into exhaustive detail in deriving the same identity using block unfoldings, many of these details can be safely ignored once we have sufficient familiarity with block unfoldings. We would thus argue that our approach can be considered simpler than the one used in [56].

3.6 Conclusions

Overall, it is reasonable to conclude from this chapter that block tensors behave in much the same way as block matrices. Although the precise formulas are more involved, the basic intuition that "all operations can be done at the block level" is correct. Block unfoldings provide a natural way to do block-based tensor computations though matrix unfoldings and allow for intuitive visualizations. By making precise the notion of a block unfolding and developing a framework for reasoning about block tensor computation, we hope that we have laid a modest foundation for further research on block tensor algorithms.

CHAPTER 4

THE TENSOR KRONECKER PRODUCT SVD

In this chapter we define a new tensor decomposition that is especially well suited for tensors with block structure and many different types of symmetry, giving a data-sparse approximation that allows for efficient computation. To this end we introduce and investigate a new tensor operation that generalizes the matrix Kronecker product.

In §4.1 we review the matrix Kronecker SVD (KSVD) from [58] and explore, through a fourth order-example, a way of connecting it to tensor unfoldings and the tensor outer product and show how some common tensor structures are reflected in the computations. A two-factor outer product nearness problem for tensors is considered in §4.2 as well as its connections to the KSVD and tensor symmetries. In §4.3 a novel tensor operation, the *tensor Kronecker product*, is introduced. It turns out to strongly mirror the properties of the familiar matrix Kronecker product and we show how it interacts with several typical tensor operations.

A tensor version of the Kronecker singular value decomposition, which we call the *TKSVD*, is defined in §4.4 and its properties, especially as they relate to tensor symmetries, are investigated. The properties of the TKSVD for tensors with block-level structure, such as block-diagonal tensors, are explored in §4.5, resulting in an efficient algorithm.

4.1 The Matrix KSVD and Symmetries

This chapter makes heavy use of the Kronecker product, as we extend the operation to tensors. Recall that for two matrices $B \in \mathbb{R}^{m_1 \times m_2}$ and $C \in \mathbb{R}^{n_1 \times n_2}$ their

Kronecker product is the block matrix

$$B \otimes C \;=\; \begin{bmatrix} b_{11}C & b_{12}C & \cdots & b_{1n_1}C \\ b_{21}C & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ b_{m_11}C & b_{m_12}C & \cdots & b_{m_1n_1}C \end{bmatrix} \in \mathbb{R}^{m_1n_1 \times m_2n_2}. \tag{4.1}$$

In [58] it is shown that by using the *Kronecker Product SVD* (KSVD) it is possible for any matrix $A \in \mathbb{R}^{m_1n_1 \times m_2n_2}$ to solve the "closest Kronecker product problem"

$$\min_{B,C} \| A - B \otimes C \|_F \tag{4.2}$$

where $B \in \mathbb{R}^{m_1 \times m_2}$ and $C \in \mathbb{R}^{n_1 \times n_2}$. This procedure treats A as a $m_1 \times m_2$ block matrix $A = (A_{ij})$ with $n_1 \times n_2$ blocks and computes the singular value decomposition of the matrix $\mathscr{R}(A)$ where

$$\mathscr{R}(A) = \begin{pmatrix} A_1 \\ \vdots \\ A_{m_2} \end{pmatrix}, \quad \text{and} \quad A_j = \begin{pmatrix} \mathrm{vec}(A_{1,j})^T \\ \vdots \\ \mathrm{vec}(A_{m_1,j})^T \end{pmatrix}, \quad j = 1, 2, \ldots, m_2. \tag{4.3}$$

If A is a structured matrix then the matrices B and C can be chosen to inherit some of the structure. Examples include non-negativity, symmetry and positive definiteness [58]. Furthermore, the approximation (4.2) can be expanded to an exact minimal decomposition

$$A \;=\; \sum_{i=1}^{R} B_i \otimes C_i \tag{4.4}$$

where R is the minimal integer such that (4.4) is possible, referred to as the *Kronecker rank* of A, denoted

$$R \;=\; \mathrm{rank}_{\otimes}(A).$$

The expansion (4.4) has the Eckhart-Young optimality property in that for any k where $1 \leq k \leq R$ the matrix

$$M_{opt} \;=\; \sum_{i=1}^{k} B_i \otimes C_i \tag{4.5}$$

solves the constrained least squares problem

$$\min_{\text{rank}_{\otimes}(M)\leq k} \| A - M \|_F^2. \tag{4.6}$$

For convenience, we state a slight generalization of an important result shown in [58] about the KSVD of a symmetric matrix.

Theorem 4.1. *Suppose $n = n_1 n_2$, $k > 0$ and the matrix $A \in \mathbb{R}^{n\times n}$ is symmetric. Then the matrices $B_1, \ldots, B_k \in \mathbb{R}^{n_1\times n_1}$ and $C_1, \ldots, C_k \in \mathbb{R}^{n_2\times n_2}$ that minimize $\| A - \sum_{i=1}^{k} B_i \otimes C_i \|_F$ can be chosen so that for each $i = 1, \ldots, k$ the matrices B_i and C_i are either both symmetric or both skew-symmetric.*

4.1.1 The KSVD and Order-4 Tensors

There is a way to connect tensor decompositions and the matrix KSVD. To illustrate this connection, consider a fourth-order tensor $\mathcal{A} \in \mathbb{R}^{n_1\times n_2\times n_3\times n_4}$. We can view this tensor as a block matrix[1] A where $\mathcal{A}(i,j,k,\ell)$ is entry (k,ℓ) in block (i,j). This is in fact equivalent to the unfolding $A = \mathcal{A}_{[1\ 3]\times[2\ 4]}$.

If we view A in the aforementioned way as a $n_1 \times n_3$ block matrix with $n_2 \times n_4$ blocks we can compute the KSVD

$$A \ = \ \sum_{r=1}^{R} B_r \otimes C_r$$

where $R = \text{rank}_{\otimes}(A)$. Since $\mathcal{A}(i,j,k,\ell)$ is entry (k,ℓ) in block (i,j) in the matrix A we have that

$$\mathcal{A}(i,j,k,\ell) \ = \ \sum_{r=1}^{R} B_r(i,j)C_r(k,\ell), \quad \text{i.e.} \quad \mathcal{A} \ = \ \sum_{r=1}^{R} B_r \circ C_r \tag{4.7}$$

[1] not to be confused with block tensor unfoldings

where $\circ$ is the tensor outer product defined in (1.5).

Note that the outer product decomposition in (4.7) could also have been computed by taking the SVD of a different unfolding, $\tilde{A} = \mathcal{A}_{[1\ 2]\times[3\ 4]}$, and reshaping the singular vectors into matrices. Indeed, this is a standard way [41, 52, 53] of computing tensor outer product factorizations such as (4.7). This is actually the computation that the KSVD performs since, if $A = \mathcal{A}_{[1\ 3]\times[2\ 4]}$ is viewed as a $n_1 \times n_3$ block matrix with $n_2 \times n_4$ blocks, then

$$\mathscr{R}(A) \ = \ \mathcal{A}_{[1\ 2]\times[3\ 4]} \tag{4.8}$$

where $\mathscr{R}(A)$ is the rearrangement operator from (4.3). We show in §4.2 that a generalization of (4.7) holds for tensors of any order. The value of making this connection is that it nicely illustrates what is going on both at the tensor level and at the matrix unfolding level and highlights the connections between different tensor unfoldings. It also allows us to use results such as Theorem 4.1 about structured KSVD problems in the tensor setting.

4.1.2 The KSVD and Structured Order-4 Tensors

An interesting special case is where an order-4 tensor $\mathcal{A}$ has the symmetry property $\mathcal{A}(i,j,k,\ell) = \mathcal{A}(k,\ell,i,j)$, i.e. $\mathcal{A}^{<[3\ 4\ 1\ 2]>} = \mathcal{A}$. With $A = \mathcal{A}_{[1\ 3]\times[2\ 4]}$ as before, we can now choose the matrices B_i and C_i in (4.4) such that $C_i = \pm B_i$ for all $i = 1, \ldots, R$. After normalizing so that $\| B_i \|_F = 1$ we get

$$A \ = \ \sum_{i=1}^{R} \lambda_i B_i \otimes B_i \tag{4.9}$$

or equivalently

$$\mathcal{A} \ = \ \sum_{i=1}^{R} \lambda_i B_i \circ B_i$$

where $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_R| > 0$, which can be viewed as a generalization of the matrix Schur decomposition.

Other symmetry properties of $\mathcal{A}$ are also reflected in the KSVD. For example, if $\mathcal{A} = \mathcal{A}^{<[2\ 1\ 4\ 3]>}$ then

$$\mathcal{A}_{[1\ 3]\times[2\ 4]} \ = \ \mathcal{A}_{[2\ 4]\times[1\ 3]} \ = \ (\mathcal{A}_{[1\ 3]\times[2\ 4]})^T,$$

i.e. the unfolding $A = \mathcal{A}_{[1\ 3]\times[2\ 4]}$ is symmetric. By Theorem 4.1 we then have that in the KSVD expansion $A = \sum_{i=1}^{R} B_i \otimes C_i$ the matrices B_i and C_i are either both symmetric or both skew-symmetric for every $i = 1, \ldots, R$.

Similar results can be derived for many possible symmetries, skew-symmetries or combinations thereof for order-4 tensors. Table 4.1 summarizes the above lists a few more possible symmetries for order-4 tensors and how they are reflected in a KSVD approximation (4.7). A general version of these facts is presented in Theorem 4.4.

Table 4.1: Different symmetries for order-4 tensors and how they are reflected in the B_i and C_i matrices in the KSVD expansion (4.7).

$\mathcal{A}$ symmetry	B_i structure	C_i structure	Other
$[\,2\ 1\ 3\ 4\,]$-(skew)	(skew)-symmetric		
$[\,1\ 2\ 4\ 3\,]$-(skew)		(skew)-symmetric	
$[\,2\ 1\ 4\ 3\,]$			$B_i = \pm C_i$
$[\,2\ 1\ 4\ 3\,]$-skew			$B_{2i} = \pm C_{2i-1}$
symmetric	symmetric	symmetric	$B_i = \pm C_i$

4.2 The Two-Factor Nearness Problem

This section shows how the results of §4.1 connecting the KSVD to tensor decomposition for order-4 tensors generalize to tensors of any order.

In §4.2.1 we define the two factor nearness problem and the two factor outer product expansion (TFOPE). The connection to the KSVD is discussed in §4.2.2 and the interaction of the TFOPE with various tensor symmetries is explored in §4.2.3.

4.2.1 Problem Statement and Properties

It is well-known [53, 52, 27] that by using traditional tensor unfoldings it is possible to solve the *two factor outer product nearness* problem

$$\min_{\mathcal{B},\mathcal{C}} \| \mathcal{A} - \mathcal{B} \circ \mathcal{C} \|^2 \tag{4.10}$$

where $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, $\mathcal{B}$ is order-d_1 and $\mathcal{C}$ is order-d_2 with $d = d_1 + d_2$. Indeed, let $b = \text{vec}(\mathcal{B})$ and $c = \text{vec}(\mathcal{C})$. Then, using equations (1.5) and (1.7)-(1.16) on the properties of vec, *ivec* and tensor unfoldings, we have

$$\begin{aligned}
\|\mathcal{A} - \mathcal{B} \circ \mathcal{C}\|^2 &= \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \|\mathcal{A}(\mathbf{i}) - \mathcal{B}(\mathbf{i}(1{:}d_1)) \cdot \mathcal{C}(\mathbf{i}(d_1{+}1){:}d)\|^2 \\
&= \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \|\mathcal{A}_{[1:d_1]\times[d_1+1:d]}(ivec(\mathbf{i}(1{:}d_1), \mathbf{n}(1{:}d_1)), ivec(\mathbf{i}(d_1{+}1{:}d), \mathbf{n}(d_1{+}1{:}d))) \\
&\qquad - b(ivec(\mathbf{i}(1{:}d_1), \mathbf{n}(1{:}d_1))) \cdot c(ivec(\mathbf{i}(d_1{+}1{:}d), \mathbf{n}(d_1{+}1{:}d)))\|_F^2 \\
&= \|\mathcal{A}_{[1:d_1]\times[d_1+1:d]} - bc^T\|_F^2.
\end{aligned}$$

This means that the vectors b and c that solve (4.10) are multiples of the leading left and right singular vectors, respectively, of the unfolding $\mathcal{A}_{[1:d_1]\times[d_1+1:d]}$.

We can also extend the approximation (4.10) to an exact minimal decomposition of the form

$$\mathcal{A} \;=\; \sum_{i=1}^{R} \mathcal{B}_i \circ \mathcal{C}_i \tag{4.11}$$

which we refer to as the *two-factor outer product expansion* (TFOPE). This expansion is the basis for the recursive TT-decomposition [53, 52] which proceeds by decomposing the factor tensors $\mathcal{B}_i$ and $\mathcal{C}_i$.

Note that the decomposition (4.11) has the Eckhart-Young optimality property: the truncated sum $\sum_{i=1}^{k} \mathcal{B}_i \circ \mathcal{C}_i$ is the solution to the k-term approximation problem

$$\min_{\mathcal{M}_i, \mathcal{N}_i} \left\| \mathcal{A} - \sum_{i=1}^{k} \mathcal{M}_i \circ \mathcal{N}_i \right\|^2. \tag{4.12}$$

4.2.2 Connection to the KSVD

Rephrasing the k-term two-factor nearness problem (4.12), it can be solved by using the matrix Kronecker SVD. Let $[\,\mathbf{r}\;\mathbf{c}\,]$ be some permutation of $1{:}d$ and set $A = \mathcal{A}_{\mathbf{r}\times\mathbf{c}}$. Let integers e_1, e_2, f_1, f_2 be such that

$$\texttt{length}(\mathbf{r}) \;=\; e_1 + e_2, \qquad\qquad \texttt{length}(\mathbf{c}) \;=\; f_1 + f_2,$$

and set

$$d_1 \;=\; e_1 + f_1, \qquad\qquad d_2 \;=\; e_2 + f_2.$$

Define the vectors

$$\begin{aligned} \mathbf{r}_{\mathcal{B}} &\;=\; \mathbf{r}(1{:}e_1), & \mathbf{r}_{\mathcal{C}} &\;=\; \mathbf{r}(e_1{:}e_1{+}e_2), \\ \mathbf{c}_{\mathcal{B}} &\;=\; \mathbf{c}(1{:}f_1), & \mathbf{c}_{\mathcal{C}} &\;=\; \mathbf{c}(f_1{:}f_1{+}f_2). \end{aligned}$$

Consider A as a $\texttt{prod}(\mathbf{n}(\mathbf{r}_{\mathcal{B}})) \times \texttt{prod}(\mathbf{n}(\mathbf{c}_{\mathcal{B}}))$ block matrix with blocks of size $\texttt{prod}(\mathbf{n}(\mathbf{r}_{\mathcal{C}})) \times \texttt{prod}(\mathbf{n}(\mathbf{c}_{\mathcal{C}}))$.

Let matrices $B_1, \ldots, B_k$ and $C_1, \ldots, C_k$ be the solutions to (4.6) and reshape them into tensors $\mathcal{B}_i$ and $\mathcal{C}_i$ of size $[\mathbf{n}(\mathbf{r}_\mathcal{B})\ \mathbf{n}(\mathbf{c}_\mathcal{B})]$ and $[\mathbf{n}(\mathbf{r}_\mathcal{C})\ \mathbf{n}(\mathbf{c}_\mathcal{C})]$, respectively, for all $i = 1, \ldots, k$. Note that $\mathcal{B}_1, \ldots, \mathcal{B}_k$ are order-d_1 tensors and $\mathcal{C}_1, \ldots, \mathcal{C}_k$ are order-d_2.

Theorem 4.2. *Let* $\mathbf{p} = [\mathbf{r}_\mathcal{B}\ \mathbf{c}_\mathcal{B}\ \mathbf{r}_\mathcal{C}\ \mathbf{c}_\mathcal{C}]$. *Then* $\mathcal{B}_i$ *and* $\mathcal{C}_i$ *described above are solutions to*

$$\min_{\mathcal{B},\mathcal{C}} \left\| \mathcal{A}^{<\mathbf{p}>} - \sum_{i=1}^{k} \mathcal{B}_i \circ \mathcal{C}_i \right\|.$$

Note that if $\mathbf{p} = 1{:}d$ then this is equal to (4.12).

Proof. Suppose, without loss of generality, that $\mathbf{p} = 1{:}d$. Define the matrices

$$A \ = \ \mathcal{A}_{\mathbf{r}\times\mathbf{c}}, \qquad B_j \ = \ (\mathcal{B}_j)_{(1:e_1)\times(e_1+1:d_1)}, \qquad C_j \ = \ (\mathcal{C}_j)_{(1:e_2)\times(e_2+1:d_2)}$$

for all $j = 1, \ldots, k$. Then

$$\left\| \mathcal{A} - \sum_{j=1}^{k} \mathcal{B}_j \circ \mathcal{C}_j \right\|^2 \ = \ \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \left\| \mathcal{A}(\mathbf{i}) - \sum_{j=1}^{k} \mathcal{B}_j(\mathbf{i}(1{:}d_1)) \cdot \mathcal{C}_j(\mathbf{i}(d_1{+}1{:}d)) \right\|^2.$$

We note that

$$\begin{aligned}
\mathcal{A}(\mathbf{i}) \ &= \ A(ivec(\mathbf{i}(\mathbf{r}), \mathbf{n}(\mathbf{r})), ivec(\mathbf{i}(\mathbf{c}), \mathbf{n}(\mathbf{c}))), \\
\mathcal{B}_j(\mathbf{i}(1{:}d_1)) \ &= \ B_j(ivec(\mathbf{i}(\mathbf{r}_\mathcal{B}), \mathbf{n}(\mathbf{r}_\mathcal{B})), ivec(\mathbf{i}(\mathbf{c}_\mathcal{B}), \mathbf{n}(\mathbf{c}_\mathcal{B}))), \\
\mathcal{C}_j(\mathbf{i}(d_1{+}1{:}d)) \ &= \ C_j(ivec(\mathbf{i}(\mathbf{r}_\mathcal{C}), \mathbf{n}(\mathbf{r}_\mathcal{C})), ivec(\mathbf{i}(\mathbf{c}_\mathcal{C}), \mathbf{n}(\mathbf{c}_\mathcal{C}))).
\end{aligned}$$

Define the integers

$$\begin{aligned}
r &= ivec(\mathbf{i}(\mathbf{r}_\mathcal{B}), \mathbf{n}(\mathbf{r}_\mathcal{B})), \qquad s = ivec(\mathbf{i}(\mathbf{c}_\mathcal{B}), \mathbf{n}(\mathbf{c}_\mathcal{B})), \\
t &= ivec(\mathbf{i}(\mathbf{r}_\mathcal{C}), \mathbf{n}(\mathbf{r}_\mathcal{C})), \qquad u = ivec(\mathbf{i}(\mathbf{c}_\mathcal{C}), \mathbf{n}(\mathbf{c}_\mathcal{C})),
\end{aligned}$$

and $M = \texttt{prod}(\mathbf{n}(\mathbf{r}_{\mathcal{C}})), N = \texttt{prod}(\mathbf{n}(\mathbf{c}_{\mathcal{C}}))$. Then, by the definition (1.9) of *ivec*, we have

$$ivec(\mathbf{i}(\mathbf{r}), \mathbf{n}(\mathbf{r})) \;=\; t + (r-1)M \quad \text{and} \quad ivec(\mathbf{i}(\mathbf{c}), \mathbf{n}(\mathbf{c})) \;=\; u + (s-1)N.$$

Therefore

$$\begin{aligned}\Big\| \mathcal{A} - \sum_{j=1}^{k} \mathcal{B}_j \circ \mathcal{C}_j \Big\|^2 &= \sum_{r,s,t,u} \Big| A(t+(r-1)M, u+(s-1)N) - \sum_{j=1}^{k} B_j(r,s) \cdot C_j(t,u) \Big|_F^2 \\ &= \Big\| A - \sum_{j=1}^{k} B_j \otimes C_j \Big\|_F^2.\end{aligned}$$

This concludes the proof. □

As an example, suppose $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_7}$, $d_1 = 3$, $d_2 = 4$ and

$$\mathbf{r}_{\mathcal{B}} \;=\; [\,1 \;\; 2\,], \;\; \mathbf{r}_{\mathcal{C}} \;=\; [\,4 \;\; 5\,], \;\; \mathbf{c}_{\mathcal{B}} \;=\; [\,3\,], \;\; \mathbf{c}_{\mathcal{C}} \;=\; [\,6 \;\; 7\,].$$

Let $A = \mathcal{A}_{[1\ 2\ 4\ 5]\times[3\ 6\ 7]}$ and consider A as a $n_1 n_2 \times n_3$ block matrix with $n_4 n_5 \times n_6 n_7$ blocks. If $k = 1$ then by Theorem 4.2 solving $\|A - B \otimes C\|_F$ is equivalent to solving $\|\mathcal{A} - \mathcal{B} \circ \mathcal{C}\|_F$ where $\mathcal{B}$ and $\mathcal{C}$ are reshapings of B and C, respectively.

4.2.3 Symmetries

As we showed in §4.1.2 for order-4 tensors, a benefit from rephrasing (4.10) in terms of the matrix KSVD is that we can exploit the established results of the properties of the KSVD factors of structured matrices [58]. If $\mathcal{A}$ has a certain kind of symmetry then that fact will be reflected in the tensors $\mathcal{B}_i$ and $\mathcal{C}_i$ that solve (4.12) as the next theorem shows. The following facts from matrix computations are crucial.

Lemma 4.3. *Let $A \in \mathbb{R}^{m\times n}$ and its singular value decomposition be given by $A = U\Sigma V^T$. Then the following properties hold.*

(a) *If $m = n$ and A is symmetric then U and V can be chosen such that $u_i = \pm v_i$ for all $i = 1, \ldots, n$.*

(b) *If $P \in \mathbb{R}^{m\times m}$ and $Q \in \mathbb{R}^{n\times n}$ are orthogonal matrices and*

$$PAQ^T \;=\; A,$$

then we can choose the SVD of A such that $PU = U$ and $QV = V$.

Using this result we now show how tensor symmetries and skew-symmetries are reflected in the two-factor nearness problem.

Theorem 4.4. *Let $\mathcal{A}$ be an order-d tensor and $k > 0$. For $i = 1, \ldots, k$ let $\mathcal{B}_i$ and $\mathcal{C}_i$ be order-d_1 and order-d_2 tensors, respectively, that solve* (4.12)*. Suppose $\mathcal{A}$ is $\mathbf{p}$-symmetric for some permutation $\mathbf{p}$ of $1{:}d$. Then the following results hold for all $i = 1, \ldots, d$.*

(1) *If $\mathbf{p}(d_1{+}1{:}d) = (d_1{+}1){:}d$, then $\mathcal{B}_i$ can be chosen to be $\mathbf{p}(1{:}d_1)$-symmetric.*

(2) *If $\mathbf{p}(1{:}d_1) = 1{:}d_1$ then the tensor $\mathcal{C}_i$ can be chosen to $\mathbf{p}'$-symmetric, where $\mathbf{p}' = \mathbf{p}(d_1{+}1{:}d) - d_1$.*

(3) *If $\mathbf{p}(1{:}d_1) \subseteq \{1, 2, \ldots, d_1\}$ and $\mathbf{p}(d_1{+}1{:}d) \subseteq \{d_1 + 1, d_1{+}2, \ldots, d\}$ then we can choose $\mathcal{B}_i$ and $\mathcal{C}_i$ such that $\mathcal{B}_i$ is $\mathbf{p}(1{:}d_1)$-symmetric or $\mathbf{p}(1{:}d_1)$-skew-symmetric and $\mathcal{C}_i$ is $\mathbf{p}'$-symmetric or $\mathbf{p}'$-skew-symmetric, where $\mathbf{p}' = \mathbf{p}(d_1{+}1{:}d) - d_1$.*

(4) *If $d_1 = d_2$ and $\mathbf{p}(1{:}d_1) = d_1{+}1{:}d$ then we can choose $\mathcal{B}_i$ and $\mathcal{C}_i$ so that $\mathcal{B}_i = \pm\mathcal{C}_i$.*

Note that if $\mathcal{A}$ is $\mathbf{p}$-skew-symmetric then skew-symmetric versions of (1) *and* (2) *hold.*

Proof. Define $\mathbf{r} = 1{:}d_1$ and $\mathbf{c} = d_1{+}1{:}d$. For part (1), if $\mathbf{p}(\mathbf{c}) = \mathbf{c}$ and $\mathcal{A} = \mathcal{A}^{<\mathbf{p}>}$, i.e. $\mathcal{A}(\boldsymbol{\ell}) = \mathcal{A}(\boldsymbol{\ell}(\mathbf{p}))$ for all $\boldsymbol{\ell}$, then there exists a permutation matrix P such that

$$\mathcal{A}_{\mathbf{p}(\mathbf{r})\times\mathbf{p}(\mathbf{c})} \;=\; \mathcal{A}_{\mathbf{p}(\mathbf{r})\times\mathbf{c}} = P\mathcal{A}_{\mathbf{r}\times\mathbf{c}}.$$

Since $\mathcal{A}$ is $\mathbf{p}$-symmetric we have $\mathcal{A}_{\mathbf{p}(\mathbf{r})\times\mathbf{p}(\mathbf{c})} = \mathcal{A}_{\mathbf{r}\times\mathbf{c}}$. Thus $\mathbf{p}$-symmetry is equivalent to the matrix equation $P\mathcal{A}_{\mathbf{r}\times\mathbf{c}} = \mathcal{A}_{\mathbf{r}\times\mathbf{c}}$.

By Lemma 4.3 we can choose the singular value decomposition $U^T\mathcal{A}_{\mathbf{r}\times\mathbf{c}}V = \Sigma$ such that $PU = U$. Furthermore, since $\mathrm{vec}(\mathcal{B}_i) = (\mathcal{B}_i)_{\mathbf{r}\times\emptyset} = U(:,i)$ then

$$(\mathcal{B}_i)_{\mathbf{p}(\mathbf{r})\times\emptyset} \;=\; P(\mathcal{B}_i)_{\mathbf{r}\times\emptyset} \;=\; (\mathcal{B}_i)_{\mathbf{r}\times\emptyset}$$

which is equivalent to $\mathcal{B}_i = \mathcal{B}_i^{<\mathbf{p}(1:d_1)>}$, proving (1).

The proofs of parts (2) and (3) are similar and are omitted.

For part (4), note that if $\mathbf{p}(\mathbf{r}) = \mathbf{c}$, $\mathbf{p}(\mathbf{c}) = \mathbf{r}$ and $\mathcal{A} = \mathcal{A}^{<\mathbf{p}>}$ then

$$\begin{aligned}\mathcal{A}_{\mathbf{r}\times\mathbf{c}} \;&=\; \mathcal{A}_{\mathbf{p}(\mathbf{r})\times\mathbf{p}(\mathbf{c})}\\ &=\; \mathcal{A}_{\mathbf{c}\times\mathbf{r}}\\ &=\; (\mathcal{A}_{\mathbf{r}\times\mathbf{c}})^T.\end{aligned}$$

Hence by Lemma 4.3 we can choose the SVD of $\mathcal{A}_{\mathbf{r}\times\mathbf{c}}$ such that $U(:,i) = \pm V(:,i)$ for all i. Since $\mathrm{vec}(\mathcal{B}_i) = U(:,i)$ and $\mathrm{vec}(\mathcal{C}_i) = V(:,i)$ this implies $\mathcal{B}_i = \pm\mathcal{C}_i$. □

Example: Suppose the order-4 tensor $\mathcal{H} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$ has the symmetry and skew-symmetry properties

$$\mathcal{H}(i_1, i_2, i_3, i_4) \;=\; \begin{cases} \mathcal{H}(i_2, i_1, i_4, i_3) \\ \mathcal{H}(i_3, i_4, i_1, i_2) \\ -\mathcal{H}(i_2, i_1, i_3, i_4) \end{cases} \tag{4.13}$$

In other words, if $\mathbf{p}_1 = [\,2\;1\;4\;3\,], \mathbf{p}_2 = [\,3\;4\;1\;2\,]$ and $\mathbf{p}_3 = [\,2\;1\;3\;4\,]$ then $\mathcal{H}$ is $\mathbf{p}_1$- and $\mathbf{p}_2$-symmetric and $\mathbf{p}_3$-skew-symmetric. Tensors with these types of symmetries frequently arise in ab initio quantum chemistry calculations [77, 29].

If we define $d_1 = d_2 = 2$ we see that $\mathbf{p}_1$ satisfies property (3) of Theorem 4.4, $\mathbf{p}_2$ satisfies (4) and $\mathbf{p}_3$ satisfies a skew-version of (1). Hence, if $k > 0$ and the matrices $B_1, \ldots, B_k \in \mathbb{R}^{n_1 \times n_2}$ and $C_1, \ldots, C_k \in \mathbb{R}^{n_3 \times n_4}$ solve

$$\min_{B_i, C_i} \left\| \mathcal{H} - \sum_{i=1}^{k} B_i \circ C_i \right\|,$$

then for $i = 1, \ldots, k$ we have that $B_i = \pm C_i$ and B_i is skew-symmetric.

4.3 The Tensor Kronecker Product

In order to establish a higher-order generalization of the KSVD we require a new tensor operation which behaves in many ways similar to the matrix Kronecker product. In particular, it is closely associated with block tensors.

In §4.3.1 we state the definition of the Tensor Kronecker Product (TKP) in terms of block tensors, and in §4.3.2–§4.3.8 we show how it interacts with some standard tensor operations.

4.3.1 Definition

In tensor computations, one often sees expressions such as "$\mathcal{B} \otimes \mathcal{C}$" for higher-order tensors $\mathcal{B}$ and $\mathcal{C}$ but there "$\otimes$" usually refers to the tensor outer product which we denote with $\mathcal{B} \circ \mathcal{C}$. Here the notation $\mathcal{B} \otimes \mathcal{C}$ has a nonstandard meaning, making use of block tensors:

Definition 4.5. *The* Tensor Kronecker product (TKP) *of two order-d tensors* $\mathcal{B} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ *and* $\mathcal{C} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ *is the block tensor* $\mathcal{A} \equiv \mathcal{B} \otimes \mathcal{C} \in \mathbb{R}^{m_1 n_1 \times \cdots \times m_d n_d}$ *whose* $\mathbf{i}$*th block is*

$$\mathcal{A}_{\mathbf{i}} \;=\; \mathcal{B}(\mathbf{i}) \cdot \mathcal{C}, \qquad \mathbf{1} \leq \mathbf{i} \leq \mathbf{n}.$$

As an example, suppose $\mathcal{B}$ and $\mathcal{C}$ are both 3rd order tensors and that $\mathcal{B}$ has size $3 \times 3 \times 3$. Then $\mathcal{A}$ can be visualized as the $3 \times 3 \times 3$ block tensor shown in Figure 4.1.

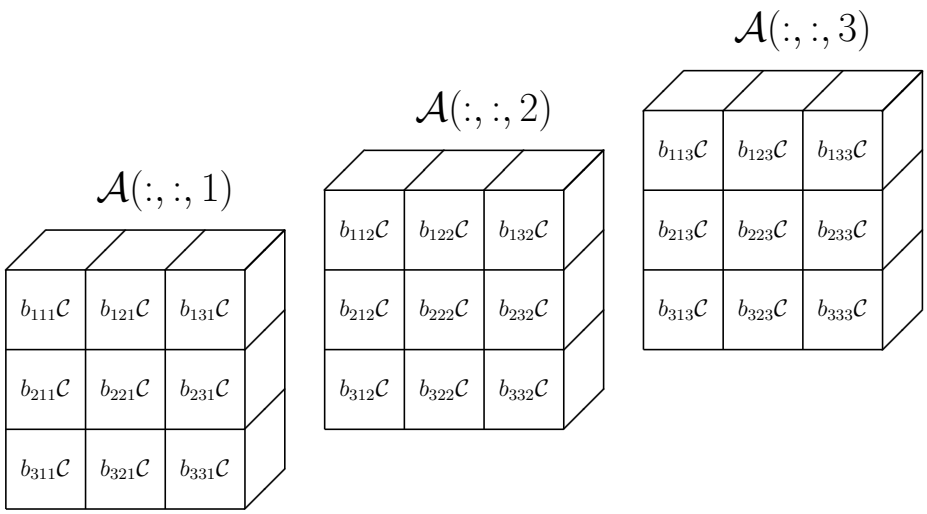


Figure 4.1: The tensor Kronecker product of two 3rd order tensors $\mathcal{B}$ and $\mathcal{C}$.

If $\mathcal{B}$ and $\mathcal{C}$ are order-2 tensors, i.e. matrices, then their tensor Kronecker product

is the same as the matrix Kronecker product. This can be readily established by comparing the above definition to the matrix Kronecker product definition (4.1).

To prevent any misunderstanding, we reemphasize that our use of the "$\otimes$" symbol for tensors is nonstandard and exclusively used for the matrix and tensor Kronecker products.

Note that the order-d tensor $\mathcal{B} \otimes \mathcal{C}$ and the order-$2d$ tensor $\mathcal{B} \circ \mathcal{C}$ actually have the same elements, but arranged differently. Indeed, if $\mathcal{A} = \mathcal{B} \circ \mathcal{C}$ and $\mathcal{A}' = \mathcal{B} \otimes \mathcal{C}$ where $\mathcal{B}$ is $n_1 \times \cdots \times n_d$ and $\mathcal{C}$ is $m_1 \times \cdots \times m_d$ then

$$\mathcal{A}(j_1, \ldots, j_d, i_1, \ldots, i_d) \;=\; \mathcal{A}'(i_1 + m_1(j_1 - 1), \ldots, i_d + m_d(j_d - 1)).$$

The block tensor extension of the Matlab Tensor Toolbox discussed in Chapter 3 provides the function `tkron` which implements the tensor Kronecker product. If $\mathcal{B}$ and $\mathcal{C}$ are `tensor` objects or multidimensional arrays of equal order then `tkron(B,C)` is the tensor $\mathcal{B} \otimes \mathcal{C}$. For example, if

```
B = tensor(rand(3,3,3));
C = tensor(rand(7,4,5));
A = tkron(B,C);
```

then $\mathcal{A} \in \mathbb{R}^{21 \times 16 \times 15}$ is a $3 \times 3 \times 3$ block tensor that we can visualize by Figure 4.1. For more details on `tkron`, see Appendix A.

The definition of the tensor Kronecker product assumes that both tensors $\mathcal{B}$ and $\mathcal{C}$ have the same order. At first this might seem too restrictive; after all, we can take the Kronecker product of a vector with a matrix. However, this is accomplished by treating the vector as a "thin matrix", either a $1 \times n$ row vector or

a $n \times 1$ column vector. In a similar way we can take the tensor Kronecker product of lower order tensors with higher order ones, but the lower order tensor needs to be "upgraded" to match the higher order by inserting modes of size 1. For example, if $\mathcal{B}$ is a $n_1 \times n_2$ matrix and $\mathcal{C}$ is a $m_1 \times m_2 \times m_3$ order-3 tensor then we can treat $\mathcal{B}$ as an order-3 tensor of size $1 \times n_1 \times n_2$, $n_1 \times 1 \times n_2$ or $n_1 \times n_2 \times 1$. This allows us to take its tensor Kronecker product with $\mathcal{C}$. Removing size-1 modes is often referred to as "squeezing" a tensor, so this reverse process of inserting such modes could be called "unsqueezing". This unsqueezing is clearly not unique, however, and must be carefully selected to get the desired result. Indeed, there are $\binom{d+k-1}{d-1}$ ways of unsqueezing an order-d tensor to an order-$(d+k)$ tensor.

For matrices there are several connections between $B \otimes C$ and B and C in terms of rank, eigenvalues, singular values and other properties. This is what makes working with Kronecker products appealing: computations can be done efficiently. Many of these connections extend to the tensor Kronecker product and will be important when we discuss the TKSVD in §4.4.

4.3.2 Connection to Block Unfoldings

As mentioned in earlier chapters, the standard tensor unfoldings defined by (1.12)-(1.16) do not respect block structure; block elements are usually noncontiguous in an unfolding. To preserve block structure we use the block unfoldings introduced in Chapter 3.

The following theorem shows an important property of the tensor Kronecker product that connects it to block unfoldings.

TKP Property 1. *Let* $\mathcal{B} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ *and* $\mathcal{C} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$. *Then for any* $\mathbf{r}$ *and* $\mathbf{c}$ *we have*

$$(\mathcal{B} \otimes \mathcal{C})_{\mathbf{R} \times \mathbf{C}} \;=\; \mathcal{B}_{\mathbf{r} \times \mathbf{c}} \otimes \mathcal{C}_{\mathbf{r} \times \mathbf{c}}. \tag{4.14}$$

Note that the "$\otimes$" on the left hand side in (4.14) is the tensor Kronecker product while the "$\otimes$" on the right hand side is the matrix Kronecker product.

Proof. By definition, for any $\mathbf{1} \leq \mathbf{k} \leq \mathbf{n}$ the $\mathbf{k}$th block of $\mathcal{B} \otimes \mathcal{C}$ is

$$(\mathcal{B} \otimes \mathcal{C})_{\mathbf{k}} \;=\; \mathcal{B}(\mathbf{k}) \cdot \mathcal{C}.$$

Hence, by equation (3.31) on the blocks of a block unfolding, we have

$$\begin{aligned} \mathcal{B}(\mathbf{k}) \cdot \mathcal{C}_{\mathbf{r} \times \mathbf{c}} \;&=\; ((\mathcal{B} \otimes \mathcal{C})_{\mathbf{k}})_{\mathbf{r} \times \mathbf{c}} \\ &=\; ((\mathcal{B} \otimes \mathcal{C})_{\mathbf{R} \times \mathbf{C}})_{\mathbf{k}(\mathbf{r}), \mathbf{k}(\mathbf{c})} \end{aligned}$$

i.e. block $(\mu, \tau) = (ivec(\mathbf{k}(\mathbf{r}), \mathbf{n}(\mathbf{r})), ivec(\mathbf{k}(\mathbf{c}), \mathbf{n}(\mathbf{c})))$ in the block matrix $(\mathcal{B} \otimes \mathcal{C})_{\mathbf{R} \times \mathbf{C}}$ is $\mathcal{B}(\mathbf{k}) \cdot \mathcal{C}_{\mathbf{r} \times \mathbf{c}}$.

On the other hand, by (1.14) block (μ, τ) in the block matrix $\mathcal{B}_{\mathbf{r} \times \mathbf{c}} \otimes \mathcal{C}_{\mathbf{r} \times \mathbf{c}}$ is given by

$$\begin{aligned} \mathcal{B}_{\mathbf{r} \times \mathbf{c}}(\mu, \tau) \cdot \mathcal{C}_{\mathbf{r} \times \mathbf{c}} \;&=\; \mathcal{B}_{\mathbf{r} \times \mathbf{c}}(ivec(\mathbf{k}(\mathbf{r}), \mathbf{n}(\mathbf{r})), ivec(\mathbf{k}(\mathbf{c}), \mathbf{n}(\mathbf{c}))) \cdot \mathcal{C}_{\mathbf{r} \times \mathbf{c}} \\ &=\; \mathcal{B}(\mathbf{k}) \cdot \mathcal{C}_{\mathbf{r} \times \mathbf{c}}. \end{aligned}$$

Since the block matrices $(\mathcal{B} \otimes \mathcal{C})_{\mathbf{R} \times \mathbf{C}}$ and $\mathcal{B}_{\mathbf{r} \times \mathbf{c}} \otimes \mathcal{C}_{\mathbf{r} \times \mathbf{c}}$ have the same blocks they are the same. $\square$

4.3.3 Rank of $\mathcal{B} \otimes \mathcal{C}$

For matrices B and C we have that $\operatorname{rank}(B \otimes C) = \operatorname{rank}(B) \cdot \operatorname{rank}(C)$. There are several distinct definitions of rank for higher-order tensors, but we will restrict our attention to the so-called multilinear rank and the outer product rank as defined in §2.5.1 and §2.5.2, respectively.

For the tensor multilinear rank, note that by (4.14) and properties of the matrix Kronecker product, we have

$$\begin{aligned} \operatorname{rank}((\mathcal{B} \otimes \mathcal{C})_{\mathbf{R}\times\mathbf{C}}) &= \operatorname{rank}(\mathcal{B}_{\mathbf{r}\times\mathbf{c}} \otimes \mathcal{C}_{\mathbf{r}\times\mathbf{c}}) \\ &= \operatorname{rank}(\mathcal{B}_{\mathbf{r}\times\mathbf{c}}) \cdot \operatorname{rank}(\mathcal{C}_{\mathbf{r}\times\mathbf{c}}). \end{aligned}$$

This includes the modal unfoldings, i.e. where $\mathbf{r} = [\, k \,]$ and $\mathbf{c} = [\, 1{:}k{-}1,\ k{+}1{:}d \,]$. From Theorem 3.4 also have that for any block tensor $\mathcal{A}$ that the matrices $\mathcal{A}_{\mathbf{R}\times\mathbf{C}}$ and $\mathcal{A}_{\mathbf{r}\times\mathbf{c}}$ are similar and thus have the same rank.

Hence, if $\operatorname{rank}_{\boxplus}(\mathcal{B}) = (r_1(\mathcal{B}), \ldots, r_d(\mathcal{B}))$ and $\operatorname{rank}_{\boxplus}(\mathcal{C}) = (r_1(\mathcal{C}), \ldots, r_d(\mathcal{C}))$ then

$$\operatorname{rank}_{\boxplus}(\mathcal{B} \otimes \mathcal{C}) \;=\; \operatorname{rank}_{\boxplus}(\mathcal{B}) * \operatorname{rank}_{\boxplus}(\mathcal{C}) \;=\; (r_1(\mathcal{B})r_1(\mathcal{C}), \ldots, r_d(\mathcal{B})r_d(\mathcal{C}))$$

where "$*$" denotes the Hadamard product defined in (1.30).

For the outer product rank, the following theorem gives an upper bound. Recall the definition of the CP-decomposition from §1.4.1.

TKP Property 2. *Suppose the CP-decompositions of $\mathcal{B}$ and $\mathcal{C}$ are*

$$\mathcal{B} \;=\; \sum_{i=1}^{R_{\mathcal{B}}} b_1^{(i)} \circ \cdots \circ b_d^{(i)}, \qquad \mathcal{C} \;=\; \sum_{j=1}^{R_{\mathcal{C}}} c_1^{(j)} \circ \cdots \circ c_d^{(j)}$$

where $R_{\mathcal{B}} = \operatorname{rank}_{\circledcirc}(\mathcal{B})$ and $R_{\mathcal{C}} = \operatorname{rank}_{\circledcirc}(\mathcal{C})$. Then

$$\mathcal{B} \otimes \mathcal{C} \;=\; \sum_{i=1}^{R_{\mathcal{B}}R_{\mathcal{C}}} (b_1^{(i)} \otimes c_1^{(j)}) \circ \cdots \circ (b_d^{(i)} \otimes c_d^{(j)})$$

which means that

$$\text{rank}_{\circledcirc}(\mathcal{B} \otimes \mathcal{C}) \;\leq\; \text{rank}_{\circledcirc}(\mathcal{B}) \cdot \text{rank}_{\circledcirc}(\mathcal{C}). \tag{4.15}$$

Proof. For any $\mathbf{1} \leq \mathbf{k} \leq \mathbf{m} * \mathbf{n}$ there exist $\mathbf{1} \leq \mathbf{i} \leq \mathbf{n}$ and $\mathbf{1} \leq \mathbf{j} \leq \mathbf{m}$ such that $\mathbf{k} = (\mathbf{m} - \mathbf{1}_d) * \mathbf{i} + \mathbf{j}$. This means $(\mathcal{B} \otimes \mathcal{C})(\mathbf{k})$ is entry $\mathbf{i}$ in tensor block $\mathbf{j}$ and thus

$$\begin{aligned}
(\mathcal{B} \otimes \mathcal{C})(\mathbf{k}) &= \mathcal{B}(\mathbf{i})\mathcal{C}(\mathbf{j}) \\
&= \left(\sum_{s=1}^{R_{\mathcal{B}}} b_1^{(s)}(i_1) \cdots b_d^{(s)}(i_d) \right) \left(\sum_{t=1}^{R_{\mathcal{C}}} c_1^{(t)}(j_1) \cdots c_d^{(t)}(j_d) \right) \\
&= \sum_{s=1}^{R_{\mathcal{B}}} \sum_{t=1}^{R_{\mathcal{C}}} \left(b_1^{(s)}(i_1) c_1^{(t)}(j_1) \right) \cdots \left(b_d^{(s)}(i_d) c_d^{(t)}(j_d) \right) \\
&= \sum_{s=1}^{R_{\mathcal{B}}} \sum_{t=1}^{R_{\mathcal{C}}} \left(b_1^{(s)} \otimes c_1^{(t)} \right)(k_1) \cdots \left(b_d^{(s)} \otimes c_d^{(t)} \right)(k_d) \\
&= \mathcal{T}(\mathbf{k})
\end{aligned}$$

where

$$\mathcal{T} \;\equiv\; \sum_{s=1}^{R_{\mathcal{B}}} \sum_{t=1}^{R_{\mathcal{C}}} \left(b_1^{(s)} \otimes c_1^{(t)} \right) \circ \cdots \circ \left(b_d^{(s)} \otimes c_d^{(t)} \right).$$

From this we conclude that $\mathcal{B} \otimes \mathcal{C} = \mathcal{T}$. □

4.3.4 Multilinear Product Properties

For matrices, we have

$$(u \otimes x)^T (B \otimes C)(v \otimes y) \;=\; (u^T B v)(x^T C y).$$

A similar result holds for the multilinear product of tensor Kronecker products as the following theorem shows.

TKP Property 3. *If $\mathcal{B}$ is an $n_1 \times \cdots \times n_d$ tensor, $\mathcal{C}$ is $m_1 \times \cdots \times m_d$ and $X_i \in \mathbb{R}^{n_i \times p_i}$ and $Y_i \in \mathbb{R}^{m_i \times q_i}$ for $i = 1, 2, \ldots, d$ then*

$$(\mathcal{B} \otimes \mathcal{C}) \cdot (X_1 \otimes Y_1, \ldots, X_d \otimes Y_d) \;=\; (\mathcal{B} \cdot (X_1, \ldots, X_d)) \otimes (\mathcal{C} \cdot (Y_1, \ldots, Y_d)) \,. \quad (4.16)$$

Note that this is an $p_1 q_1 \times \cdots \times p_d q_d$ tensor.

Proof. First, for any integer vector $\mathbf{v} \in \mathbb{R}^\ell$ we define the functions $\varphi_{\mathbf{v}}, \psi_{\mathbf{v}} : \mathbb{R}^\ell \to \mathbb{R}^\ell$ by the formula

$$(\varphi_{\mathbf{v}}(\mathbf{u}))_i = \left\lfloor \frac{u_i}{v_i} \right\rfloor + 1 \quad \text{and} \quad (\psi_{\mathbf{v}}(\mathbf{v}))_i = \begin{cases} \operatorname{mod}(u_i, v_i) & \text{if } \operatorname{mod}(u_i, v_i) \neq 0 \\ v_i & \text{if } \operatorname{mod}(u_i, v_i) = 0 \end{cases}$$

for all $i = 1, 2, \ldots, \ell$. Then it is easy to verify that

$$(\mathcal{B} \otimes \mathcal{C})(\mathbf{j}) \;=\; \mathcal{B}(\varphi_{\mathbf{m}}(\mathbf{j})) \cdot \mathcal{C}(\psi_{\mathbf{m}}(\mathbf{j})).$$

Define the tensors

$$\begin{aligned} \mathcal{T} \;&=\; (\mathcal{B} \otimes \mathcal{C}) \cdot (X_1 \otimes Y_1, \ldots, X_d \otimes Y_d) \;\in \mathbb{R}^{p_1 q_1 \times \cdots \times p_d q_d}, \\ \mathcal{S} \;&=\; (\mathcal{B} \cdot (X_1, \ldots, X_d)) \otimes (\mathcal{C} \cdot (Y_1, \ldots, Y_d)) \;\in \mathbb{R}^{p_1 q_1 \times \cdots \times p_d q_d}. \end{aligned}$$

Then for any $\mathbf{1} \leq \mathbf{i} \leq \mathbf{p} * \mathbf{q}$ we have

$$\begin{aligned} \mathcal{T}(\mathbf{i}) \;&=\; \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{n}*\mathbf{m}} (\mathcal{B} \otimes \mathcal{C})(\mathbf{j}) (X_1 \otimes Y_1)(j_1, i_1) \cdots (X_d \otimes Y_d)(j_d, i_d) \\ &=\; \sum_{\mathbf{j}=\mathbf{1}}^{\mathbf{n}*\mathbf{m}} \mathcal{B}\left(\varphi_{\mathbf{m}}(\mathbf{j})\right) \mathcal{C}(\psi_{\mathbf{m}}(\mathbf{j})) X_1\left(\varphi_{m_1}(j_1), \varphi_{q_1}(i_1)\right) Y_1\left(\psi_{m_1}(j_1), \psi_{q_1}(i_1)\right) \\ &\qquad\qquad \cdots X_d\left(\varphi_{m_d}(j_d), \varphi_{q_d}(i_d)\right) Y_d\left(\psi_{m_d}(j_d), \psi_{q_d}(i_d)\right). \end{aligned}$$

Note that as $\mathbf{j}$ goes from $\mathbf{1}$ to $\mathbf{n} * \mathbf{m}$, the functions $\varphi_{\mathbf{m}}(\mathbf{j})$ and $\psi_{\mathbf{m}}(\mathbf{j})$ span the ranges $\mathbf{1}$ to $\mathbf{n}$ and $\mathbf{1}$ to $\mathbf{m}$, respectively. Thus if we define $\boldsymbol{\ell} \equiv \varphi_{\mathbf{m}}(\mathbf{j})$ and $\mathbf{k} \equiv \psi_{\mathbf{m}}(\mathbf{j})$ then

we get

$$\begin{aligned}\mathcal{T}(\mathbf{i}) =& \sum_{\boldsymbol{\ell}=\mathbf{1}}^{\mathbf{n}} \sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{m}} \mathcal{B}(\boldsymbol{\ell})\mathcal{C}(\mathbf{k}) X_1(\ell_1, \varphi_{q_1}(i_1)) Y_1(k_1, \psi_{q_1}(i_1)) \cdots X_d(\ell_d, \varphi_{q_d}(i_d)) Y_d(k_d, \psi_{q_d}(i_d)) \\ =& \left(\sum_{\boldsymbol{\ell}=\mathbf{1}}^{\mathbf{n}} \mathcal{B}(\boldsymbol{\ell}) X_1(\ell_1, \varphi_{q_1}(i_1)) \cdots X_d(\ell_d, \varphi_{q_d}(i_d)) \right) * \\ & \qquad \left(\sum_{\mathbf{k}=\mathbf{1}}^{\mathbf{m}} \mathcal{C}(\mathbf{k}) Y_1(k_1, \psi_{q_1}(i_1)) \cdots Y_d(k_d, \psi_{q_d}(i_d)) \right) \\ =& \mathcal{S}(\mathbf{i})\end{aligned}$$

which completes the proof. □

The special case where each of the matrices X_i and Y_i is a vector is important enough to be included on its own.

TKP Property 4. *If $\mathcal{B} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, $\mathcal{C} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ and $x_i \in \mathbb{R}^{n_i}$, $y_i \in \mathbb{R}^{m_i}$ for all $i = 1, 2, \ldots, d$ then*

$$(\mathcal{B} \otimes \mathcal{C}) \cdot (x_1 \otimes y_1, \ldots, x_d \otimes y_d) \;=\; (\mathcal{B} \cdot (x_1, \ldots, x_d))(\mathcal{C} \cdot (y_1, \ldots, y_d)) \in \mathbb{R}. \quad (4.17)$$

4.3.5 Norms and Inner Product Properties

Recall from (1.28) that the tensor inner product is defined by

$$\mathcal{X}, \mathcal{Y} \in \mathbb{R}^{k_1 \times \cdots \times k_d} \;\Rightarrow\; \langle \mathcal{X}, \mathcal{Y} \rangle = \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{k}} \mathcal{X}(\mathbf{i}) \mathcal{Y}(\mathbf{i}) \in \mathbb{R}. \quad (4.18)$$

If x, y, u and v are vectors of suitable sizes then it is easy to show the identity $(x \otimes y)^T (u \otimes v) = (x^T u)(y^T v)$. The same result holds for the tensor Kronecker product:

TKP Property 5. *If* $\mathcal{X},\mathcal{U} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ *and* $\mathcal{Y},\mathcal{V} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ *then*

$$\langle \mathcal{X} \otimes \mathcal{Y}\,,\, \mathcal{U} \otimes \mathcal{V} \rangle \;=\; \langle \mathcal{X},\mathcal{U} \rangle \cdot \langle \mathcal{Y},\mathcal{V} \rangle.$$

Proof. Using Definition 4.5 and (1.28) we have

$$\begin{aligned}
\langle \mathcal{X} \otimes \mathcal{Y},\mathcal{U} \otimes \mathcal{V} \rangle \;&=\; \sum_{\mathbf{k=1}}^{\mathbf{n*m}} (\mathcal{X} \otimes \mathcal{Y})(\mathbf{k}) \cdot (\mathcal{U} \otimes \mathcal{V})(\mathbf{k}) \\
&=\; \sum_{\mathbf{i=1}}^{\mathbf{m}} \sum_{\mathbf{j=1}}^{\mathbf{n}} (\mathcal{X} \otimes \mathcal{Y})(\mathbf{i}+\mathbf{m}*(\mathbf{j}-\mathbf{1}))(\mathcal{U} \otimes \mathbf{V})(\mathbf{i}+\mathbf{m}*(\mathbf{j}-\mathbf{1})) \\
&=\; \sum_{\mathbf{i=1}}^{\mathbf{m}} \sum_{\mathbf{j=1}}^{\mathbf{n}} \mathcal{X}(\mathbf{i})\mathcal{Y}(\mathbf{j})\mathcal{U}(\mathbf{i})\mathcal{V}(\mathbf{j}) \\
&=\; \langle \mathcal{X},\mathcal{U} \rangle \cdot \langle \mathcal{Y},\mathcal{V} \rangle
\end{aligned}$$

which completes the proof. □

Note that if $\mathcal{X} = \mathcal{U}$ and $\mathcal{Y} = \mathcal{V}$ then this implies $\|\mathcal{X} \otimes \mathcal{Y}\| = \|\mathcal{X}\| \cdot \|\mathcal{Y}\|$.

4.3.6 Singular Value and Eigenvalue Properties

Recall from §2.3.1 that a Z-singular value of a tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is a scalar $\sigma \in \mathbb{R}$ such that there exist unit vectors $u_1, \ldots, u_d$, where $u_i \in \mathbb{R}^{n_i}$, called Z-singular vectors, that have the property that for every $k = 1, \ldots, d$ we have

$$\mathcal{A} \cdot (u_1, \ldots, u_{k-1}, I_{n_k}, u_{k+1}, \ldots, u_d) \;=\; \sigma u_k.$$

Since this definition is in terms of the multilinear product, we can use (4.17) to get the following result.

TKP Property 6. *If* σ *is a Z-singular value of* $\mathcal{B}$ *with Z-singular vectors* $u_1, \ldots, u_d$ *and* μ *is a Z-singular value of* $\mathcal{C}$ *with Z-singular vectors* $v_1, \ldots, v_d$ *then* $\sigma\mu$ *is a Z-singular value of* $\mathcal{B} \otimes \mathcal{C}$ *with Z-singular vectors* $u_1 \otimes v_1, \ldots, u_d \otimes v_d$.

Proof. We have $u_i \in \mathbb{R}^{n_i}$ and $v_i \in \mathbb{R}^{m_i}$ such that

$$\mathcal{B} \cdot (u_1, \ldots, u_{i-1}, I_{n_i}, u_{i+1}, \ldots, u_d) \;=\; \sigma u_i$$

and

$$\mathcal{C} \cdot (v_1, \ldots, v_{i-1}, I_{m_i}, v_{i+1}, \ldots, v_d) \;=\; \mu v_i$$

for all $i = 1, \ldots, d$. Then

$$\begin{aligned}
&(\mathcal{B} \otimes \mathcal{C}) \cdot (u_1 \otimes v_1, \ldots u_{i-1} \otimes v_{i-1}, I_{n_i m_i}, u_{i+1} \otimes v_{i+1}, \ldots, u_d \otimes v_d) \\
&= (\mathcal{B} \cdot (u_1, \ldots, u_{i-1}, I_{n_i}, u_{i+1}, \ldots, u_d)) \otimes (\mathcal{C} \cdot (v_1, \ldots, v_{i-1}, I_{m_i}, v_{i+1}, \ldots, v_d)) \\
&= (\sigma u_i) \otimes (\mu v_i) = \sigma\mu (u_i \otimes v_i)
\end{aligned}$$

which completes the proof. □

In §2.3.2 we defined $\lambda \in \mathbb{R}$ to be a Z-eigenvalue of a symmetric order-d tensor $\mathcal{A} \in \mathbb{R}^{N \times \cdots \times N}$ if there exists a unit vector $x \in \mathbb{R}^N$ such that

$$\mathcal{A} \cdot (I_N, x, \ldots, x) \;=\; \lambda x.$$

Note that if $\mathcal{B}$ and $\mathcal{C}$ are symmetric tensors, then $\mathcal{B} \otimes \mathcal{C}$ is also symmetric. We therefore get the following result.

TKP Property 7. *Let $\mathcal{B}$ and $\mathcal{C}$ be symmetric tensors. If λ is a Z-eigenvalue of $\mathcal{B}$ with Z-eigenvector x and τ is an Z-eigenvalue of $\mathcal{C}$ with Z-eigenvector y. Then $\mathcal{B} \otimes \mathcal{C}$ has Z-eigenvalue $\lambda\tau$ with Z-eigenvector $x \otimes y$.*

Proof. We have

$$\mathcal{B} \cdot (I_n, x, \ldots, x) \;=\; \lambda x \quad \text{and} \quad \mathcal{C} \cdot (I_m, y, \ldots, y) \;=\; \tau y.$$

Then by Theorem TKP 3

$$\begin{aligned}(\mathcal{B}\otimes\mathcal{C})\cdot(I_{nm},x\otimes y,\ldots,x\otimes y) &= (\mathcal{B}\cdot(I_n,x,\ldots,x))\otimes(\mathcal{C}\cdot(I_m,y,\ldots,y))\\ &= (\lambda x)\otimes(\tau y)\\ &= \lambda\tau(x\otimes y).\end{aligned}$$

which completes the proof. □

We note that TKP Properties 6 and 5 can be readily modified for any of the several other tensor singular value and eigenvalue definitions mentioned in §2.6.

4.3.7 Tensor Transposition of $\mathcal{B}\otimes\mathcal{C}$

For Kronecker products of matrices we have formula

$$(B\otimes C)^T = B^T\otimes C^T.$$

A higher-order analogue of this using tensor transposition holds as well.

TKP Property 8. *Let $\mathcal{B}$ and $\mathcal{C}$ be order-d tensors and $\mathbf{p}$ be a permutation of $1{:}d$. Then*

$$(\mathcal{B}\otimes\mathcal{C})^{<\mathbf{p}>} = \mathcal{B}^{<\mathbf{p}>}\otimes\mathcal{C}^{<\mathbf{p}>}.$$

Proof. This is straightforward result of Lemma 2.1 and Definition 4.5. □

4.3.8 The HOSVD of $\mathcal{B}\otimes\mathcal{C}$

It is well known that the singular value decomposition of a matrix $A = B\otimes C$ can be formed by only computing the SVDs of the smaller matrices B and C. Indeed,

if $B = U\Sigma V^T$ and $C = X\Delta Y^T$ are the SVDs of B and C then $A = W\Gamma Z^T$ is the SVD of A, where

$$W \;=\; (U \otimes X)P^T, \qquad \Gamma \;=\; P(\Sigma \otimes \Delta)Q^T, \qquad Z \;=\; (V \otimes Y)Q^T$$

and P and Q are permutation matrices that sort the diagonal entries of the matrix $\Sigma \otimes \Delta$ in descending order.

As the HOSVD is a generalization of the SVD and tensor Kronecker products are generalizations of matrix Kronecker products, it is reasonable to ask if the HOSVD of $\mathcal{A} = \mathcal{B} \otimes \mathcal{C}$ can be generated by only computing the HOSVDs of the small tensors $\mathcal{B}$ an $\mathcal{C}$. This is indeed the case.

Suppose that $\mathcal{A} = \mathcal{B} \otimes \mathcal{C}$ is an order-d $N_1 \times \cdots \times N_d$ tensor where $N_i = n_i m_i$, $\mathcal{B}$ is $n_1 \times \cdots \times n_d$ and $\mathcal{C}$ is $m_1 \times \cdots \times m_d$. Furthermore, let $\mathcal{B} = (U_1^{(\mathcal{B})}, \ldots, U_d^{(\mathcal{B})}) \cdot \mathcal{S}_\mathcal{B}$ and $\mathcal{C} = (U_1^{(\mathcal{C})}, \ldots, U_d^{(\mathcal{C})}) \cdot \mathcal{S}_\mathcal{C}$ be the HOSVDs. From Theorem TKP 3 we know that

$$\begin{aligned}
\mathcal{A} \;&=\; \mathcal{B} \otimes \mathcal{C} \\
&=\; \Big((U_1^{(\mathcal{B})}, \ldots, U_d^{(\mathcal{B})}) \cdot \mathcal{S}_\mathcal{B}\Big) \otimes \Big((U_1^{(\mathcal{C})}, \ldots, U_d^{(\mathcal{C})}) \cdot \mathcal{S}_\mathcal{C}\Big) \\
&=\; (U_1^{(\mathcal{B})} \otimes U_1^{(\mathcal{C})}, \ldots, U_d^{(\mathcal{B})} \otimes U_d^{(\mathcal{C})}) \cdot (\mathcal{S}_\mathcal{B} \otimes \mathcal{S}_\mathcal{C}).
\end{aligned}$$

Clearly each $U_i^{(\mathcal{B})} \otimes U_i^{(\mathcal{C})}$ is an orthogonal matrix. We thus need to determine the properties of $\mathcal{S}_\mathcal{B} \otimes \mathcal{S}_\mathcal{C}$. Recall that a $p_1 \times \cdots \times p_d$ tensor $\mathcal{S}$ is called *all-orthogonal* if for every $k = 1, 2, \ldots, d$ and $\alpha, \beta = 1, 2, \ldots, p_k$, $\alpha \neq \beta$ we have

$$\langle \mathcal{S}_{i_k=\alpha}, \mathcal{S}_{i_k=\beta} \rangle \;=\; 0.$$

where $\mathcal{S}_{i_k=\alpha}$ is the order-$(d-1)$ subtensor of $\mathcal{S}$ we get by fixing the kth index to be α.

Using TKP Property 5 we can show that $\mathcal{S}_\mathcal{B} \otimes \mathcal{S}_\mathcal{C}$ satisfies one of the requirements of the HOSVD core tensor.

TKP Property 9. *If $\mathcal{S}_\mathcal{B}$ and $\mathcal{S}_\mathcal{C}$ are all all-orthogonal then so is $\mathcal{S}_\mathcal{B} \otimes \mathcal{S}_\mathcal{C}$.*

Proof. Define $\mathcal{R} \equiv \mathcal{S}_\mathcal{B} \otimes \mathcal{S}_\mathcal{C}$. Fix $k \in \{1, 2, \ldots, d\}$. For every $\gamma = 1, 2, \ldots, n_k m_k$ there exist $\alpha \in \{1, 2, \ldots, n_k\}$ and $\beta \in \{1, 2, \ldots, m_k\}$ such that

$$\mathcal{R}_{i_k=\gamma} \;=\; ((\mathcal{S}_\mathcal{B})_{i_k=\alpha}) \otimes ((\mathcal{S}_\mathcal{B})_{i_k=\beta}).$$

Hence for $\gamma_1 \neq \gamma_2$ there exist $\alpha_1, \alpha_2, \beta_1, \beta_2$ such that

$$\begin{aligned}\langle \mathcal{R}_{i_k=\gamma_1}, \mathcal{R}_{i_k=\gamma_2} \rangle \;&=\; \langle ((\mathcal{S}_\mathcal{B})_{i_k=\alpha_1}) \otimes ((\mathcal{S}_\mathcal{B})_{i_k=\beta_1}), ((\mathcal{S}_\mathcal{B})_{i_k=\alpha_2}) \otimes ((\mathcal{S}_\mathcal{B})_{i_k=\beta_2}) \rangle \\ &=\; \langle (\mathcal{S}_\mathcal{B})_{i_k=\alpha_1}, (\mathcal{S}_\mathcal{B})_{i_k=\alpha_2} \rangle \langle (\mathcal{S}_\mathcal{C})_{i_k=\beta_1}, (\mathcal{S}_\mathcal{C})_{i_k=\beta_2} \rangle\end{aligned}$$

Since $\gamma_1 \neq \gamma_2$ then at least one of the statements $\alpha_1 \neq \alpha_2$ or $\beta_1 \neq \beta_2$ must be true. But then by the all-orthogonality of $\mathcal{S}_\mathcal{B}$ and $\mathcal{S}_\mathcal{C}$ at least one of $\langle (\mathcal{S}_\mathcal{B})_{i_k=\alpha_1}, (\mathcal{S}_\mathcal{B})_{i_k=\alpha_2} \rangle$ and $\langle (\mathcal{S}_\mathcal{C})_{i_k=\beta_1}, (\mathcal{S}_\mathcal{C})_{i_k=\beta_2} \rangle$ must be zero. □

Another requirement is the norm grading, i.e. that for any $k = 1, 2, \ldots, d$ we have

$$\|\mathcal{S}_{i_k=1}\| \;\geq\; \|\mathcal{S}_{i_k=2}\| \;\geq\; \cdots \;\geq\; \|\mathcal{S}_{i_k=n_k}\|.$$

However, for any tensor $\mathcal{S}$ there exist permutation matrices $P_1, \ldots, P_d$ such that $(P_1, \ldots, P_d) \cdot \mathcal{S}$ has this property.

Hence, there exist permutation matrices such that

$$\begin{aligned}\mathcal{A} \;&=\; \left((U_1^{(\mathcal{B})} \otimes U_1^{(\mathcal{C})}) P_1^T, \ldots, (U_d^{(\mathcal{B})} \otimes U_d^{(\mathcal{C})}) P_d^T \right) \cdot ((P_1, \ldots, P_d) \cdot \mathcal{S}_\mathcal{B} \otimes \mathcal{S}_\mathcal{C}) \\ &\equiv\; (U_1^{(\mathcal{A})}, \ldots, U_d^{(\mathcal{A})}) \cdot \mathcal{S}_\mathcal{A}\end{aligned}$$

is the HOSVD of $\mathcal{A}$.

4.4 The Tensor Kronecker Product SVD

Having firmly established the tensor Kronecker product and its properties we are now ready to introduce the main contribution of this chapter, the Tensor Kronecker Product SVD (TKSVD). The TKSVD itself is defined in §4.4.1. In §4.4.2 the TKSVD is contrasted with other SVD-based tensor decompositions and compared to the HOSVD in terms of degrees of freedom and low-rank approximation. The interactions of the TKSVD with common types of tensor symmetries are then explored in §4.4.3.

4.4.1 Definition

Suppose $\mathcal{A}$ is an order-d tensor of size $N_1 \times \cdots \times N_d$ and $N_i = m_i n_i$ for $i = 1, 2, \ldots, d$. Then we can view $\mathcal{A}$ as a $m_1 \times \cdots \times m_d$ block tensor whose blocks are all of size $n_1 \times \cdots \times n_d$. Let $\mathbf{M}$ be this uniform blocking.

Let $\mathbf{p}$ be a permutation of the vector $1{:}d$, integer e be such that $1 \leq e \leq d$ and define

$$\mathbf{r} \;=\; \mathbf{p}(1{:}e), \quad \mathbf{c} \;=\; \mathbf{p}(e+1{:}d) \quad \text{and} \quad A \;=\; \mathcal{A}_{\mathbf{R}\times\mathbf{C}}$$

where $\mathbf{R} = \mathbf{M}(\mathbf{r})$ and $\mathbf{C} = \mathbf{M}(\mathbf{c})$. Each block in the $(n_{r_1} \cdots n_{r_e}) \times (n_{c_1} \cdots n_{c_{d-e}})$ block matrix A is an $\mathbf{r} \times \mathbf{c}$ unfolding of a tensor block in $\mathcal{A}$.

We can then compute the Kronecker singular value decomposition of A, regarded as a $(n_{r_1} \cdots n_{r_e}) \times (n_{c_1} \cdots n_{c_{d-e}})$ block matrix with $(m_{r_1} \cdots m_{r_e}) \times (m_{c_1} \cdots m_{c_{d-e}})$ blocks. Using the KSVD, we solve the least squares problem

$$\min_{B,C} \| A - B \otimes C \|_F. \tag{4.19}$$

Note that there is a scalar indeterminacy here: if B and C solve (4.19) then so do αB and $\frac{1}{\alpha}C$ for any $\alpha \neq 0$. To resolve this we add the constraint that $\| B \|_F = \| C \|_F$. Now regard B and C as $\mathbf{r} \times \mathbf{c}$ tensor unfoldings and reshape them appropriately to the tensors $\mathcal{B} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $\mathcal{C} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$, respectively.

Theorem 4.6. *The tensors $\mathcal{B}$ and $\mathcal{C}$ described above are solutions to the minimization problem*

$$\min_{\mathcal{B},\mathcal{C}} \| \mathcal{A} - \mathcal{B} \otimes \mathcal{C} \|. \tag{4.20}$$

Proof. Using the definitions of block unfoldings, the KSVD and the tensor Kronecker product, we have

$$\begin{aligned} \|A - B \otimes C\|_F^2 &= \sum_{\mathbf{i=1}}^{\mathbf{n}} \|A_{\mathbf{i(r)},\mathbf{i(c)}} - B\left(ivec(\mathbf{i}(\mathbf{r}), \mathbf{n}(\mathbf{c})), ivec(\mathbf{i}(\mathbf{c}), \mathbf{n}(\mathbf{c}))\right) C\|_F^2 \\ &= \sum_{\mathbf{i=1}}^{\mathbf{n}} \|A_{\mathbf{i(r)},\mathbf{i(c)}} - \mathcal{B}(\mathbf{i})C\|_F^2 \\ &= \sum_{\mathbf{i=1}}^{\mathbf{n}} \|\mathcal{A}_{\mathbf{i}} - \mathcal{B}(\mathbf{i})\mathcal{C}\|^2 \;=\; \|\mathcal{A} - \mathcal{B} \otimes \mathcal{C}\|^2 \end{aligned}$$

which completes the proof. □

Using the full KSVD expansion (4.4) for the block unfolding $\mathcal{A}_{\mathbf{R}\times\mathbf{C}}$ we can get an exact representation.

Definition 4.7. *If $\mathcal{A} \in \mathbb{R}^{N_1 \times \cdots \times N_d}$ where $N_i = m_i n_i$ for all $i = 1, 2, \ldots, d$ and $\mathcal{B}_j \in \mathbb{R}^{m_1 \times \cdots \times m_d}, \mathcal{C}_j \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $R \in \mathbb{N}$ are solutions to*

$$\min \left\{ K \in \mathbb{N} \;\middle|\; \exists X_j \in \mathbb{R}^{m_1 \times \cdots \times m_d}, Y_j \in \mathbb{R}^{n_1 \times \cdots \times n_d} \text{ such that } \mathcal{A} = \sum_{j=1}^{K} X_j \otimes Y_j \right\}$$

with $\| \mathcal{B}_j \| = \| \mathcal{C}_j \|$ for $j = 1, 2, \ldots, R$, then

$$\mathcal{A} \;=\; \sum_{i=1}^{R} \mathcal{B}_i \otimes \mathcal{C}_i \tag{4.21}$$

is the tensor Kronecker product SVD (TKSVD) *of* $\mathcal{A}$ *and* R *is the* tensor Kronecker rank *of* $\mathcal{A}$*, denoted* $R = \text{rank}_{\otimes}(\mathcal{A})$.

Note that the k-term truncation $\sum_{i=1}^{k} \mathcal{B}_i \otimes \mathcal{C}_i$ of (4.21) is the solution to the constrained least squares problem

$$\min_{\text{rank}_{\otimes}(\mathcal{M}) \leq k} \| \mathcal{A} - \mathcal{M} \|^2. \tag{4.22}$$

We now can compute a data-sparse approximation of a tensor with block structure using (4.20). This is especially important if $\mathcal{A}$ has block-level structure, such as block sparsity. By the results of §4.3 we see that computations involving tensor Kronecker products can be performed efficiently and as we show in §4.5, the Kronecker factor tensors can be chosen to reflect many common types of structure in the original tensor. Hence, if a good TKSVD approximation can be achieved by using a small k in (4.22) then we can reap significant savings in subsequent computations.

For clarity, we state the steps necessary to compute a k-term TKSVD approximation in Algorithm 4.1. Let "fold" be the reshape operation that undoes a matrix unfolding. That is, for any tensor $\mathcal{T} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ we have

$$\mathcal{T} \;=\; \text{fold}(\mathcal{T}_{\mathbf{r} \times \mathbf{c}}, \mathbf{r}, \mathbf{c}, \mathbf{n}). \tag{4.23}$$

Note that the matrix $\mathscr{R}(A)$ has the size $\left(\prod_{i=1}^{d} n_i\right) \times \left(\prod_{i=1}^{d} m_i\right)$ regardless of which modes $\mathbf{r}$ and $\mathbf{c}$ are chosen. The number of flops required by this SVD calculation can then be estimated by standard methods. See [26, §5.4.5]. If k is small it can be advantageous to use Lanczos-based algorithms to compute only the first k singular values and singular vectors of $\mathscr{R}(A)$.

Algorithm 4.1 The Tensor Kronecker SVD (TKSVD)

Require: $\mathcal{A} \in \mathbb{R}^{N_1 \times \cdots \times N_d}$ with $N_i = n_i m_i, \forall i$.

Require: $\mathbf{M}$ is the uniform blocking of $\mathcal{A}$ and $\mathbf{R} = \mathbf{M}(\mathbf{r})$, $\mathbf{C} = \mathbf{M}(\mathbf{c})$.

1: $A \leftarrow \mathcal{A}_{\mathbf{R} \times \mathbf{C}}$

2: Compute the SVD $\mathscr{R}(A) = U \Sigma V^T$

3: **for** $i = 1, \ldots, k$ **do**

4: $\quad \mathcal{B}_i \leftarrow \sqrt{\Sigma(i,i)} \cdot \text{fold}(U(:,i), [\mathbf{r}\ \mathbf{c}], \emptyset, \mathbf{n})$

5: $\quad \mathcal{C}_i \leftarrow \sqrt{\Sigma(i,i)} \cdot \text{fold}(V(:,i), [\mathbf{r}\ \mathbf{c}], \emptyset, \mathbf{m})$

6: **end for**

Example: If $\mathcal{D}_i \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ for $i = 1, \ldots, k$ we define the *block diagonal tensor* $\mathcal{D} = \text{blkdiag}(\mathcal{D}_1, \ldots, \mathcal{D}_k) \in \mathbb{R}^{kn_1 \times \cdots \times kn_d}$ to be the $k \times \cdots \times k$ block tensor where

$$\mathcal{D}_{\mathbf{i}} \;=\; \begin{cases} \mathcal{D}_j & \text{if } \ \mathbf{i} = j \cdot \mathbf{1}_d \ \text{ for some } \ j = 1, \ldots, k \\ 0 & \text{else} \end{cases} \tag{4.24}$$

for all $\mathbf{i} = \mathbf{1}, \ldots, \mathbf{k}$. If $\mathcal{B} \in \mathbb{R}^{k \times \cdots \times k}$ and $\mathcal{C} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ minimize $\|\mathcal{D} - \mathcal{B} \otimes \mathcal{C}\|_F$ then it is straightforward to show that $\mathcal{B} = \text{diag}(b_1, \ldots, b_k)$ is a diagonal tensor, i.e.

$$\mathcal{B}(\mathbf{i}) \;=\; \begin{cases} b_j & \text{if } \ \mathbf{i} = j \cdot \mathbf{1}_d \ \text{ for some } \ j = 1, \ldots, k \\ 0 & \text{else} \end{cases} \tag{4.25}$$

for all $\mathbf{i} = \mathbf{1}, \ldots, \mathbf{k}$, and

$$\mathcal{C} \;=\; \left(\sum_{i=1}^{k} b_i \mathcal{D}_i \right) \Big/ \sum_{i=1}^{k} b_i^2.$$

Furthermore, if we require $\| C \|_F = 1$ then the b_i solve the equation

$$\sum_{i=1}^{k} b_i^2 \;=\; \Big\| \sum_{i=1}^{k} b_i \mathcal{D}_i \Big\|^2.$$

The Matlab function `tksvd` computes the TKSVD. If $\mathcal{A}$ is an $n_1 \times \cdots \times n_d$ block tensor with $m_1 \times \cdots \times m_d$ blocks then the command

```
[B,C] = tksvd(A,n,k);
```

computes the k-term TKSVD approximation (4.22), where `B = {B{1},...,B{k}}` and `C = {C{1},...,C{k}}` are cell arrays that contain the tensors $\mathcal{B}_i \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $\mathcal{C}_i \in \mathbb{R}^{m_1 \times \cdots \times m_d}$. For example, if $\mathcal{A} \in \mathbb{R}^{15 \times 20 \times 10}$ is randomly generated with $\mathbf{n} = [\,3\ 4\ 2\,]$ and $\mathbf{m} = [\,5\ 5\ 5\,]$ and we want a two-term Kronecker approximation we use the following commands:

```
A = tensor(rand(15,20,10));
n = [3 4 2];
[B,C] = tksvd(A,n,2);
```

See also Appendix A for more detail on `tksvd`.

4.4.2 Comparison to Other Decompositions

The TKSVD is similar in some ways to the HOSVD, as described in §1.4.3, and the two-factor outer product expansion (TFOPE) from §4.2. Table 4.2 summarizes the details of these decompositions.

Table 4.2: Comparison of different tensor decompositions.

	HOSVD	TKSVD	TFOPE
Unfolding(s)	Modal $\mathcal{A}_{(1)}, \ldots, \mathcal{A}_{(d)}$	Blocked $\mathcal{A}_{\mathbf{R} \times \mathbf{C}}$	Non-blocked $\mathcal{A}_{[1:d_1] \times [d_1+1:d]}$
Computation	SVD	KSVD	SVD
Result	$\mathcal{A} = (U_1, \ldots, U_d) \cdot \mathcal{S}$	$\mathcal{A} = \sum_i \mathcal{B}_i \otimes \mathcal{C}_i$	$\mathcal{A} = \sum_i \mathcal{B}_i \circ \mathcal{C}_i$

As we see, each decomposition is computed by taking the SVD or KSVD of one or more tensor unfoldings and then reassembling the results.

An important application of many tensor decompositions is to obtain "low rank" (or data-sparse) approximations of large tensors. One way of obtaining a low rank approximation of a tensor $\mathcal{A}$ is by truncating the HOSVD: a k-term HOSVD approximation of $\mathcal{A}$ is of the form

$$\mathcal{A} \;\approx\; (U_1^{(k)}, \ldots, U_d^{(k)}) \cdot \mathcal{S}^{(k)}$$

where $U_i^{(k)} = U_i(:, 1{:}k)$ and $\mathcal{S}^{(k)} = \mathcal{S}(1{:}k, \ldots, 1{:}k)$. Note that this approximation has multilinear rank $(k, k, \ldots, k)$.

By using a truncated TKSVD of the form (4.22) we can now compute another type of low rank/data-sparse tensor approximation.

To make a roughly fair comparison between an ℓ-term TKSVD approximation and and k-term HOSVD approximation in terms of degrees of freedom, let us assume for simplicity that $\mathcal{A}$ is an $N \times \cdots \times N$ tensor where $N = nm$, $\mathcal{B}$ is $n \times \cdots \times n$ and $\mathcal{C}$ is $m \times \cdots \times m$ and that N is so large that we can ignore lower-order terms in our complexity analysis. Then an ℓ-term TKSVD approximation requires about $\ell(n^d + m^d)$ parameters, whereas a k-term HOSVD approximation requires about k^d parameters. Thus we need

$$\ell\,(n^d + m^d) \;\approx\; k^d$$

for approximations with roughly the same degrees of freedom. Simplifying further, assume that $n = m = \sqrt{N}$. Then, solving for ℓ, we get

$$\ell \;\approx\; \frac{1}{2}\left(\frac{k}{\sqrt{N}}\right)^d.$$

For $\ell = 1$, i.e. a one-term TKSVD approximation, we need $k \approx \sqrt[d]{2}\sqrt{N}$ terms in a truncated HOSVD.

As an example, if $d = 4$ and $N = 100$, i.e. $\mathcal{A}$ is a $100 \times 100 \times 100 \times 100$ tensor and we approximate $\mathcal{A} \approx \mathcal{B} \otimes \mathcal{C}$ using (4.20) where $\mathcal{B}$ and $\mathcal{C}$ are $10 \times 10 \times 10 \times 10$ then we need at least $k = 12 > \sqrt[4]{2}\sqrt{100} \approx 11.8$ terms in a truncated HOSVD approximation to get a similar number of degrees of freedom.

4.4.3 The TKSVD and Tensor Symmetries

Having earlier established the fundamental properties of the tensor Kronecker product, we have the tools we need to show that the TKSVD has many of the same properties as the matrix KSVD. For example, the following lemma is a direct result of the non-negativity properties of the KSVD as shown in [58].

Theorem 4.8. *If $\mathcal{A}$ is a nonnegative tensor then $\mathcal{B}$ and $\mathcal{C}$ that solve* (4.20) *can be chosen to be nonnegative.*

Here we will focus on tensors with certain types of symmetries. If tensor $\mathcal{A}$ has symmetry properties, e.g. $\mathcal{A}(i, j, k) = \mathcal{A}(j, i, k)$, we would like to know what this implies for the tensors $\mathcal{B}_i$ and $\mathcal{C}_i$ in the expansion (4.22). Ideally $\mathcal{B}_i$ and $\mathcal{C}_i$ would have the same symmetries as $\mathcal{A}$.

We first define explicitly what kind of symmetries we will consider.

Definition 4.9. *For any permutation* $\mathbf{p}$ *of* 1:d*, we say that the order-d tensor $\mathcal{A}$ is* $\mathbf{p}$-symmetric *if $\mathcal{A}(\mathbf{i}) = \mathcal{A}(\mathbf{i}(\mathbf{p}))$ for any $\mathbf{i} = \mathbf{1}, ..., \mathbf{n}$.*

It turns out that if $\mathcal{A}$ is $\mathbf{p}$-symmetric then the $\mathcal{B}_i$ and $\mathcal{C}_i$ tensors that solve k-term Kronecker approximation problem (4.22) can be chosen to be either both $\mathbf{p}$-

symmetric or both **p**-skew symmetric. To establish this fact, we need the following lemma.

Lemma 4.10. *Let $\mathcal{A}$ be a block tensor. Then $\mathcal{A}$ is* **p***-symmetric if and only if*

$$\mathcal{A}_{\mathbf{i}} \;=\; (\mathcal{A}_{\mathbf{i}(\mathbf{p})})^{<\mathbf{p}>} \tag{4.26}$$

for every block $\mathcal{A}_{\mathbf{i}}$.

Proof. This is a straightforward application of Lemma 2.1. □

Note that Lemma 4.10 is a direct generalization of the matrix case, where a block matrix $A = (A_{ij})$ is symmetric if and only if $A_{ij} = A_{ji}^T$ for all i and j.

If we fix $\mathcal{C}$ then the one-term TKSVD approximation (4.20) is a linear least squares problem with unknowns $\mathcal{B}(\mathbf{i})$. Similarly, if we fix $\mathcal{B}$ then (4.20) is a linear least squares problem in the $\mathcal{C}(\mathbf{j})$. The following lemma gives the solutions to these least squares problems

Lemma 4.11. *Suppose $\mathcal{A}$ is an $N_1 \times \cdots \times N_d$ tensor where $N_i = m_i n_i$. If $\mathcal{C} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ is fixed then the tensor $\mathcal{B} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ defined by*

$$\mathcal{B}(\mathbf{i}) \;=\; \frac{\langle \mathcal{A}_{\mathbf{i}}, \mathcal{C} \rangle}{\langle \mathcal{C}, \mathcal{C} \rangle}, \quad \mathbf{1} \le \mathbf{i} \le \mathbf{n} \tag{4.27}$$

minimizes $\|\mathcal{A} - \mathcal{B} \otimes \mathcal{C}\|$. Likewise, if $\mathcal{B}$ is fixed, then the tensor $\mathcal{C}$ defined by

$$\mathcal{C}(\mathbf{i}) \;=\; \frac{\langle \hat{\mathcal{A}}_{\mathbf{i}}, \mathcal{B} \rangle}{\langle \mathcal{B}, \mathcal{B} \rangle}, \quad \mathbf{1} \le \mathbf{i} \le \mathbf{m} \tag{4.28}$$

minimizes $\|\mathcal{A} - \mathcal{B} \otimes \mathcal{C}\|$ where $\hat{\mathcal{A}}_{\mathbf{i}}$ is the **i***th block in the tensor*

$$\hat{\mathcal{A}} \;=\; (\Pi_{n_1,m_1}, \ldots, \Pi_{n_d,m_d}) \cdot \mathcal{A}$$

regarded as an $m_1 \times \cdots \times m_d$ block tensor with $n_1 \times \cdots \times n_d$ blocks and $\Pi_{q,r}$ is the (q, r) perfect shuffle matrix, i.e. $\hat{\mathcal{A}}_{\mathbf{i}} = \mathcal{A}(i_1{:}m_1{:}N_1, \ldots, i_d{:}m_d{:}N_d)$.

Proof. As the derivations of (4.27) and (4.28) are almost identical we will only prove the former. We have

$$\begin{aligned} \| \mathcal{A} - \mathcal{B} \otimes \mathcal{C} \|^2 &= \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \| \mathcal{A}_{\mathbf{i}} - \mathcal{B}(\mathbf{i})\mathcal{C} \|^2 \\ &= \sum_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \| \operatorname{vec}(\mathcal{A}_{\mathbf{i}}) - \mathcal{B}(\mathbf{i})\operatorname{vec}(\mathcal{C}) \|_2^2. \end{aligned}$$

It is easy to verify that

$$\min_{\alpha} \| \operatorname{vec}(\mathcal{A}_{\mathbf{i}}) - \alpha \operatorname{vec}(\mathcal{C}) \|_2$$

has the solution

$$\alpha = \frac{\operatorname{vec}(\mathcal{A}_{\mathbf{i}})^T \operatorname{vec}(\mathcal{C})}{\operatorname{vec}(\mathcal{C})^T \operatorname{vec}(\mathcal{C})} = \frac{\langle \mathcal{A}_{\mathbf{i}}, \mathcal{C} \rangle}{\langle \mathcal{C}, \mathcal{C} \rangle}$$

which completes the proof. □

Using these facts, we show that if $\mathcal{A}$ and $\mathcal{C}$ are fixed and **p**-symmetric, then a **p**-symmetric $\mathcal{B}$ can be found to minimize $\|\mathcal{A} - \mathcal{B} \otimes \mathcal{C}\|$.

Theorem 4.12. *If $N_i = m_i n_i$, $\mathcal{A} \in \mathbb{R}^{N_1 \times \cdots \times N_d}$ and $\mathcal{C} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ are* **p***-symmetric then there exists a* **p***-symmetric $\mathcal{B} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ that minimizes $\|\mathcal{A} - \mathcal{B} \otimes \mathcal{C}\|$.*

Proof. Since $\mathcal{A}$ is symmetric we have $\mathcal{A}_{\mathbf{i}} = (\mathcal{A}_{\mathbf{i}(\mathbf{p})})^{<\mathbf{p}>}$ for any $\mathbf{1} \leq \mathbf{i} \leq \mathbf{n}$ by Lemma 4.10. Using properties of the tensor inner product we have

$$\mathcal{B}(\mathbf{i}(\mathbf{p})) = \frac{\langle \mathcal{A}_{\mathbf{i}(\mathbf{p})}, \mathcal{C} \rangle}{\langle \mathcal{C}, \mathcal{C} \rangle} = \frac{\langle (\mathcal{A}_{\mathbf{i}(\mathbf{p})})^{<\mathbf{p}>}, \mathcal{C}^{<\mathbf{p}>} \rangle}{\langle \mathcal{C}, \mathcal{C} \rangle} = \frac{\langle \mathcal{A}_{\mathbf{i}}, \mathcal{C} \rangle}{\langle \mathcal{C}, \mathcal{C} \rangle} = \mathcal{B}(\mathbf{i})$$

for all $\mathbf{1} \leq \mathbf{i} \leq \mathbf{n}$. It follows that $\mathcal{B}$ is **p**-symmetric. □

In the same way we can show that a **p**-symmetric $\mathcal{C}$ can be found that minimizes $\|\mathcal{A} - \mathcal{B} \otimes \mathcal{C}\|$ if $\mathcal{A}$ and $\mathcal{B}$ are fixed and **p**-symmetric.

Analogous results hold if the factor held constant is **p**-skew symmetric.

Corollary 4.13. *If $N_i = m_i n_i, \mathcal{A} \in \mathbb{R}^{N_1 \times \cdots \times N_d}$ is $\mathbf{p}$-symmetric and $\mathcal{C} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is $\mathbf{p}$-skew symmetric then there exists a $\mathbf{p}$-skew symmetric $\mathcal{B} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ that minimizes $\|\mathcal{A} - \mathcal{B} \otimes \mathcal{C}\|$.*

Proof.

$$\mathcal{B}(\mathbf{i}(\mathbf{p})) \;=\; \frac{\langle \mathcal{A}_{\mathbf{i}(\mathbf{p})}, \mathcal{C} \rangle}{\langle \mathcal{C}, \mathcal{C} \rangle} \;=\; \frac{\langle (\mathcal{A}_{\mathbf{i}(\mathbf{p})})^{<\mathbf{p}>}, \mathcal{C}^{<\mathbf{p}>} \rangle}{\langle \mathcal{C}, \mathcal{C} \rangle} \;=\; -\frac{\langle \mathcal{A}_{\mathbf{i}}, \mathcal{C} \rangle}{\langle \mathcal{C}, \mathcal{C} \rangle} \;=\; -\mathcal{B}(\mathbf{i})$$

which means that $\mathcal{B}$ is $\mathbf{p}$-skew symmetric. □

The following definition is useful for proving the next few lemmas.

Definition 4.14. *Let $P_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{k}}$ be the $\mathtt{prod}(\mathbf{k}) \times \mathtt{prod}(\mathbf{k})$ permutation matrix that takes $\mathrm{vec}(\mathcal{X}_{\mathbf{r}\times\mathbf{c}})$ to $\mathrm{vec}((\mathcal{X}^{<\mathbf{p}>})_{\mathbf{r}\times\mathbf{c}})$ for any $k_1 \times \cdots \times k_d$ tensor $\mathcal{X}$, i.e.*

$$P_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{k}} \mathrm{vec}(\mathcal{X}_{\mathbf{r}\times\mathbf{c}}) \;=\; \mathrm{vec}((\mathcal{X}^{<\mathbf{p}>})_{\mathbf{r}\times\mathbf{c}}). \tag{4.29}$$

It is clear that this matrix always exists, since $\mathcal{X}$ and $\mathcal{X}^{<\mathbf{p}>}$ have the same entries. The exact structure of $P_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{k}}$ is not important for our purposes, but it can be derived from Lemma 3.1.

We next prove a tensor generalization of Theorem 4.1, which stated how symmetry is reflected in the matrix KSVD.

Theorem 4.15. *Suppose $\mathbf{N} = \mathbf{m} * \mathbf{n}$ and $\mathcal{A} \in \mathbb{R}^{N_1 \times \cdots \times N_d}$ is $\mathbf{p}$-symmetric for some permutation $\mathbf{p}$ of $1{:}d$. If $\mathcal{B}_1, \ldots, \mathcal{B}_k \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ and $\mathcal{C}_1, \ldots, \mathcal{C}_k \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ solve (4.22) then for every $i = 1, \ldots, k$ the tensors $\mathcal{B}_i$ and $\mathcal{C}_i$ can be chosen to be either both $\mathbf{p}$-symmetric or both $\mathbf{p}$-skew-symmetric.*

Proof. Let $A = \mathcal{A}_{\mathbf{R}\times\mathbf{C}}$, where $\mathbf{R}$ and $\mathbf{C}$ are (uniform) blockings of modes $\mathbf{r}$ and $\mathbf{c}$, respectively, such that $\mathcal{A}$ is an $m_1 \times \cdots \times m_d$ tensor with $n_1 \times \cdots \times n_d$ blocks.

Note that A is a block matrix with blocks of size $\mathtt{prod}(\mathbf{n}(\mathbf{r})) \times \mathtt{prod}(\mathbf{n}(\mathbf{c}))$ and let $\mathscr{R}(A)$ be the KSVD rearrangement matrix, i.e.

$$\mathscr{R}(A) = \begin{bmatrix} \mathrm{vec}((\mathcal{A}_{\mathbf{1}})_{\mathbf{r}\times\mathbf{c}})^T \\ \vdots \\ \mathrm{vec}((\mathcal{A}_{\mathbf{m}})_{\mathbf{r}\times\mathbf{c}})^T \end{bmatrix}.$$

Recall that by Lemma 4.10 the $\mathbf{p}$-symmetry of $\mathcal{A}$ is equivalent to $\mathcal{A}_{\mathbf{i}} = (\mathcal{A}_{\mathbf{i}(\mathbf{p})})^{<\mathbf{p}>}$ for all blocks. Using (4.29) we see that

$$P_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{m}} \begin{bmatrix} \mathrm{vec}((\mathcal{A}_{\mathbf{1}})_{\mathbf{r}\times\mathbf{c}})^T P^T_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{n}} \\ \vdots \\ \mathrm{vec}((\mathcal{A}_{\mathbf{m}})_{\mathbf{r}\times\mathbf{c}})^T P^T_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{n}} \end{bmatrix} = \begin{bmatrix} \mathrm{vec}((\mathcal{A}^{<\mathbf{p}>}_{\mathbf{1}(\mathbf{p})})_{\mathbf{r}\times\mathbf{c}})^T \\ \vdots \\ \mathrm{vec}((\mathcal{A}^{<\mathbf{p}>}_{\mathbf{m}(\mathbf{p})})_{\mathbf{r}\times\mathbf{c}})^T \end{bmatrix} = \begin{bmatrix} \mathrm{vec}((\mathcal{A}_{\mathbf{1}})_{\mathbf{r}\times\mathbf{c}})^T \\ \vdots \\ \mathrm{vec}((\mathcal{A}_{\mathbf{m}})_{\mathbf{r}\times\mathbf{c}})^T \end{bmatrix}$$

i.e.

$$P_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{m}} \mathscr{R}(A) P^T_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{n}} \;=\; \mathscr{R}(A).$$

Now let b'_i and c'_i be the ith left and right singular vectors of $\mathscr{R}(A)$, respectively, with associate singular value σ_i. Let $\mu_i\tau_i = \sigma_i$ and $b_i = \mu_i b'_i$ and $c_i = \tau_i c'_i$. By the properties of the matrix KSVD and tensor unfoldings, it follows that $b_i = \mathrm{vec}((\mathcal{B}_i)_{\mathbf{r}\times\mathbf{c}})$ and $c_i = \mathrm{vec}((\mathcal{C}_i)_{\mathbf{r}\times\mathbf{c}})$. Furthermore, by Lemma 4.3 we have

$$\begin{aligned} P_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{m}} b_i &= \pm b_i \\ P_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{n}} c_i &= \pm c_i \end{aligned}$$

and thus

$$\begin{aligned} P_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{m}} \mathrm{vec}((\mathcal{B}_i)_{\mathbf{r}\times\mathbf{c}}) &= \mathrm{vec}((\mathcal{B}_i^{<\mathbf{p}>})_{\mathbf{r}\times\mathbf{c}}) = \pm\mathrm{vec}((\mathcal{B}_i)_{\mathbf{r}\times\mathbf{c}}) \\ P_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{n}} \mathrm{vec}((\mathcal{C}_i)_{\mathbf{r}\times\mathbf{c}}) &= \mathrm{vec}((\mathcal{C}_i^{<\mathbf{p}>})_{\mathbf{r}\times\mathbf{c}}) = \pm\mathrm{vec}((\mathcal{C}_i)_{\mathbf{r}\times\mathbf{c}}) \end{aligned}$$

so

$$\begin{aligned} \mathcal{B}_i^{<\mathbf{p}>} &= \pm\mathcal{B}_i \\ \mathcal{C}_i^{<\mathbf{p}>} &= \pm\mathcal{C}_i \end{aligned}$$

which means that $\mathcal{B}_i$ and $\mathcal{C}_i$ are either both $\mathbf{p}$-symmetric or both $\mathbf{p}$-skew-symmetric.

□

Corollary 4.16. *If $\mathcal{A}$ is $\mathbf{p}$-skew-symmetric and $\mathcal{B}_1, \ldots, \mathcal{B}_k$ and $\mathcal{C}_1, \ldots, \mathcal{C}_k$ solve (4.22), then for all $i = 1, \ldots, k$ exactly one of of $\mathcal{B}_i$ and $\mathcal{C}_i$ can be chosen to be $\mathbf{p}$-symmetric and the other $\mathbf{p}$-skew-symmetric.*

Proof. We get that

$$P_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{m}}\, \mathscr{R}(A)\, P^T_{\mathbf{p},\mathbf{r}\times\mathbf{c},\mathbf{n}} \;=\; -\mathscr{R}(A).$$

and thus

$$\begin{aligned} \mathcal{B}_i^{<\mathbf{p}>} &= \mp\mathcal{B}_i \\ \mathcal{C}_i^{<\mathbf{p}>} &= \pm\mathcal{C}_i \end{aligned}$$

which means that either $\mathcal{B}_i$ is $\mathbf{p}$-skew and $\mathcal{C}_i$ is $\mathbf{p}$-symmetric or vice versa. □

Higher dimensional tensor often have several symmetries or skew-symmetries at once, one prominent example are symmetric tensors, i.e. such that $\mathcal{A} = \mathcal{A}^{<\mathbf{p}>}$ for *any* permutation $\mathbf{p}$. The above results generalize to this case, that is if $\mathcal{A}$ is $\mathbf{p}_1, \mathbf{p}_2, \ldots, \mathbf{p}_\ell$-symmetric then each $\mathcal{B}_i$ and $\mathcal{C}_i$ are both either $\mathbf{p}_j$-symmetric or $\mathbf{p}_j$-skew-symmetric for all $i = 1, \ldots, k$ and $j = 1, \ldots, \ell$.

4.5 Multilevel Tensors

The types of tensor structures, such as $\mathbf{p}$-symmetry, considered in §4.4.3 and earlier in this chapter are not specific to block tensors or the tensor Kronecker product.

Indeed, this is why such structure is only partially reflected in the TKSVD. Now we shall consider *multilevel* structure that can only be described at the block level, and correspondingly has a much stronger connection to the TKSVD.

Example: Suppose

$$\mathcal{A} \in \mathbb{R}^{2k\times\cdots\times 2k} = \text{blkdiag}(\mathcal{D}_1, \ldots, \mathcal{D}_k) \tag{4.30}$$

is a block-diagonal order-d tensor with $2 \times \cdots \times 2$ blocks, where each block $\mathcal{D}_i$ is $\mathbf{p}$-symmetric for some permutation $\mathbf{p}$ of $1{:}d$. Then $\mathcal{A}$ is a multilevel tensor: it is diagonal at the block level and $\mathbf{p}$-symmetric within each block. Note that if $\mathcal{B} \in \mathbb{R}^{2k\times\cdots\times 2k}$ has the same structure as $\mathcal{A}$ then so does any linear combination of $\mathcal{A}$ and $\mathcal{B}$.

4.5.1 Definition

We will for the most part use the notation of [51], adapted to tensors. For any $\mathbf{n} \in \mathbb{N}_+^d$ denote $\mathbb{R}^{\mathbf{n}} \equiv \mathbb{R}^{n_1\times\cdots\times n_d}$, i.e. the vector space of all real-valued $n_1 \times \cdots \times n_d$ tensors and let $\mathbb{S}^{\mathbf{n}}$ be a linear subspace of $\mathbb{R}^{\mathbf{n}}$. If $\mathcal{T} \in \mathbb{S}^{\mathbf{n}}$ we say that the tensor $\mathcal{T}$ has $\mathbb{S}$-*structure.* We differentiate between subspaces by using subscripts, e.g. $\mathbb{S}^{\mathbf{n}}_\alpha$ and $\mathbb{S}^{\mathbf{n}}_\beta$.

Let $\mathbf{N} = \mathbf{n} * \mathbf{m}$, $\mathcal{A} \in \mathbb{R}^{\mathbf{N}}$ and consider $\mathcal{A}$ as an $n_1 \times \cdots \times n_d$ block tensor with $m_1 \times \cdots \times m_d$ blocks. We say that $\mathcal{A} \in \mathbb{S}^{\mathbf{n}}_\alpha \otimes \mathbb{S}^{\mathbf{m}}_\beta$ if and only if

$$\mathcal{A}_{\mathbf{i}} \in \mathbb{S}^{\mathbf{m}}_\beta \quad \text{for all} \quad \mathbf{1} \le \mathbf{i} \le \mathbf{n},$$

and

$$\hat{\mathcal{A}}_{\mathbf{j}} \equiv (\mathcal{A}_{\mathbf{i}}(\mathbf{j}))_{\mathbf{i}=\mathbf{1}}^{\mathbf{n}} \in \mathbb{S}_{\alpha}^{\mathbf{n}} \quad \text{for all} \quad \mathbf{1} \leq \mathbf{j} \leq \mathbf{m}.$$

In other words, $\mathcal{A}$ has $\mathbb{S}_\alpha$ block structure and $\mathbb{S}_\beta$-structure within each block. We call tensors with this type of structure *multilevel tensors*.

Example: In the previous example (4.30) we have $\mathcal{A} \in \mathbb{D}^{\mathbf{k}} \otimes \mathbb{P}^{\mathbf{2}}$ where $\mathbb{D}^{\mathbf{k}}$ is the space of all $k \times \cdots \times k$ diagonal tensors and $\mathbb{P}^{\mathbf{2}}$ is the space of all $\mathbf{p}$-symmetric $2 \times \cdots \times 2$ tensors.

4.5.2 Properties

In [51] the case of multilevel matrices, i.e. matrices with Kronecker structure, is considered. There it is shown that, in an optimal low Kronecker rank approximation, each of the component matrices inherit the structure of the original matrix.

Theorem 4.17 (Olshevsky et al. [51]). *Let* $A \in \mathbb{S}_{\alpha}^{[n_1\ n_2]} \otimes \mathbb{S}_{\beta}^{[m_1\ m_2]}$ *and assume*

$$\left\| A - \sum_{i=1}^{r} B_i \otimes C_i \right\|_F = \min_{\text{rank}_\otimes(A_r) \leq r} \| A - A_r \|_F. \tag{4.31}$$

Then $B_i \in \mathbb{S}_{\alpha}^{[n_1\ n_2]}$ *and* $C_i \in \mathbb{S}_{\beta}^{[m_1\ m_2]}$ *for all* $i = 1, 2, \ldots, r$.

Because of the connection between the TKSVD and the matrix KSVD through block unfoldings we get the following extension of the above result. The crucial observation is that *a block unfolding of a multilevel tensor is a multilevel matrix.*

Corollary 4.18. *Let* $\mathcal{A} \in \mathbb{S}^{\mathbf{n}}_{\alpha} \otimes \mathbb{S}^{\mathbf{m}}_{\beta}$ *and assume*

$$\left\| \mathcal{A} - \sum_{i=1}^{r} \mathcal{B}_i \otimes \mathcal{C}_i \right\| = \min_{\operatorname{rank}_{\otimes}(\mathcal{A}_r) \leq r} \| \mathcal{A} - \mathcal{A}_r \|, \tag{4.32}$$

i.e. the $\mathcal{B}_i$ *and* $\mathcal{C}_i$ *are given by the TKSVD. Then* $\mathcal{B}_i \in \mathbb{S}^{\mathbf{n}}_{\alpha}$ *and* $\mathcal{C}_i \in \mathbb{S}^{\mathbf{m}}_{\beta}$ *for all* $i = 1, \ldots, r$.

Proof. Consider $\mathcal{A}$ as a uniform $n_1 \times \cdots \times n_d$ block tensor with $m_1 \times \cdots \times m_d$ blocks. Let $\mathbf{p} = [\,\mathbf{r}\ \mathbf{c}\,]$ be a permutation of $1{:}d$ and define the matrices $A \equiv \mathcal{A}_{\mathbf{R}\times\mathbf{C}}$ and $B_i \equiv (\mathcal{B}_i)_{\mathbf{r}\times\mathbf{c}}, C_i \equiv (\mathcal{C}_i)_{\mathbf{r}\times\mathbf{c}}$ for $i = 1, \ldots, r$.

Since $\mathcal{A} \in \mathbb{S}^{\mathbf{n}}_{\alpha} \otimes \mathbb{S}^{\mathbf{m}}_{\beta}$ we have that $A \in \hat{\mathbb{S}}^{[N_1\ N_2]}_{\alpha} \otimes \hat{\mathbb{S}}^{[M_1\ M_2]}_{\beta}$ where $N_1 = \mathtt{prod}(\mathbf{n}(\mathbf{r}))$, $N_2 = \mathtt{prod}(\mathbf{n}(\mathbf{c}))$, $M_1 = \mathtt{prod}(\mathbf{m}(\mathbf{r})), M_2 = \mathtt{prod}(\mathbf{m}(\mathbf{c}))$ and the matrix structure classes $\hat{\mathbb{S}}^{[N_1\ N_2]}_{\alpha}$ and $\hat{\mathbb{S}}^{[M_1\ M_2]}_{\beta}$ are defined by

$$\mathcal{T} \in \mathbb{S}^{\mathbf{k}} \iff \mathcal{T}_{\mathbf{r}\times\mathbf{c}} \in \hat{\mathbb{S}}^{[K_1\ K_2]}. \tag{4.33}$$

By Theorem 4.6 the matrix $\sum_{i=1}^{r} B_i \otimes C_i$ solves

$$\min_{\operatorname{rank}_{\otimes}(A_r) \leq r} \| A - A_r \|_F$$

and thus by Theorem 4.17 we have $B_i \in \hat{\mathbb{S}}^{[N_1\ N_2]}_{\alpha}$ and $C_i \in \hat{\mathbb{S}}^{[M_1\ M_2]}_{\beta}$. By the equivalence (4.33) this means that $\mathcal{B}_i \in \mathbb{S}^{\mathbf{n}}_{\alpha}$ and $\mathcal{C}_i \in \mathbb{S}^{\mathbf{m}}_{\beta}$ for all $i = 1, \ldots, r$. □

Note that, in general, an unfolding $\mathcal{T}_{\mathbf{r}\times\mathbf{c}}$ of a structured tensor $\mathcal{T}$ does not necessarily have the corresponding matrix structure, i.e. in the definition (4.33) we usually have $\hat{\mathbb{S}}^{[K_1\ K_2]} \neq \mathbb{S}^{[K_1\ K_2]}$. For examples, see [1].

The singular value decompositions needed to compute the tensors $\mathcal{B}_i$ and $\mathcal{C}_i$ can be computed efficiently by exploiting the $\hat{\mathbb{S}}^{[N_1\ N_2]}_{\alpha}$ and $\hat{\mathbb{S}}^{[M_1\ M_2]}_{\beta}$ structures, as described in [51]. This process is essentially equivalent to choosing bases for these

structured classes and doing computations with a reduced number of variables. The results are then projected back to the full space $\mathbb{R}^{\mathbf{N}}$. Suppose for simplicity that $\mathbf{n} = [n_1\ n_2]$ and $\mathbf{m} = [m_1\ m_2]$, i.e. $A \in \mathbb{S}_\alpha^{[n_1\ n_2]} \otimes \mathbb{S}_\beta^{[m_1\ m_2]}$ where

$$\dim(\mathbb{S}_\alpha^{[n_1\ n_2]}) = p \quad \text{and} \quad \dim(\mathbb{S}_\beta^{[m_1\ m_2]}) = q.$$

Let

$$\mathscr{G}_\alpha = \left\{G_\alpha^{(1)}, \ldots, G_\alpha^{(p)}\right\} \quad \text{and} \quad \mathscr{G}_\beta = \left\{G_\beta^{(1)}, \ldots, G_\beta^{(q)}\right\}$$

be bases for $\mathbb{S}_\alpha^{[n_1\ n_2]}$ and $\mathbb{S}_\beta^{[m_1\ m_2]}$, respectively. Let $a_\alpha \in \mathbb{R}^p$ be the block-coordinate vector of A with respect to $\mathscr{G}_\alpha$ and $a_\beta^1, \ldots, a_\beta^p \in \mathbb{R}^q$ be the coordinates of the p unique blocks of A with respect to the basis $\mathscr{G}_\beta$.

Define the matrices

$$G_\alpha = \left[\operatorname{vec}(G_\alpha^{(1)}) \ \cdots \ \operatorname{vec}(G_\alpha^{(p)})\right] \in \mathbb{R}^{n_1 n_2 \times p}$$
$$G_\beta = \left[\operatorname{vec}(G_\beta^{(1)}) \ \cdots \ \operatorname{vec}(G_\beta^{(q)})\right] \in \mathbb{R}^{m_1 m_2 \times q}$$

and the matrix $\tilde{A} \in \mathbb{R}^{p \times q}$ defined by

$$\tilde{A}(i,:) \ = \ a_\alpha(i)\, a_\beta^i$$

is the reduced-coordinate representation of A with respect to $\mathscr{G}_\alpha$ and $\mathscr{G}_\beta$. It is shown in [51] that

$$G_\alpha \tilde{A} G_\beta^T \ = \ \mathscr{R}(A) \tag{4.34}$$

where $\mathscr{R}(A)$ is the KSVD rearrangement matrix given by (4.3). If $\mathcal{A}$ is a higher-order tensor then the procedure is the same, the only difference being that the basis elements in $\mathscr{G}_\alpha$ and $\mathscr{G}_\beta$ are tensors. Note that $\tilde{\mathcal{A}}$ will still be a matrix.

Thus Algorithm 4.2 gives an efficient way to compute the k-term TKSVD approximation of a multilevel tensor $\mathcal{A}$. Recall the definition of the "fold" operation from (4.23).

Algorithm 4.2 Computing the k-term TKSVD approximation of a multilevel tensor.

Require: $\mathcal{A} \in \mathbb{S}^{\mathbf{n}}_{\alpha} \otimes \mathbb{S}^{\mathbf{m}}_{\beta}$.

Require: $\mathscr{G}_{\alpha} = \{\mathcal{G}^{(1)}_{\alpha}, \ldots, \mathcal{G}^{(p)}_{\alpha}\}$ is a basis for $\mathbb{S}^{\mathbf{n}}_{\alpha}$ and $G_{\alpha} = [\mathrm{vec}(\mathcal{G}^{(1)}_{\alpha}) \cdots \mathrm{vec}(\mathcal{G}^{(p)}_{\alpha})]$.

Require: $\mathscr{G}_{\beta} = \{\mathcal{G}^{(1)}_{\beta}, \ldots, \mathcal{G}^{(p)}_{\beta}\}$ is a basis for $\mathbb{S}^{\mathbf{m}}_{\beta}$ and $G_{\beta} = [\mathrm{vec}(\mathcal{G}^{(1)}_{\beta}) \cdots \mathrm{vec}(\mathcal{G}^{(p)}_{\beta})]$.

1: Compute the thin QR decompositions $G_{\alpha} = Q_{\alpha}R_{\alpha}$ and $G_{\beta} = Q_{\beta}R_{\beta}$

2: Compute the singular value decomposition $R_{\alpha}\tilde{A}R^{T}_{\beta} = U\Sigma V^{T}$

3: $\tilde{U} \leftarrow Q_{\alpha}U$

4: $\tilde{V} \leftarrow Q_{\beta}V$

5: **for** $i = 1, \ldots, k$ **do**

6: $\quad \mathcal{B}_i \leftarrow \mathrm{fold}(\tilde{U}(:, i), 1{:}d, \emptyset, \mathbf{n})$

7: $\quad \mathcal{C}_i \leftarrow \mathrm{fold}(\tilde{V}(:, i), 1{:}d, \emptyset, \mathbf{m})$

8: **end for**

We note that Algorithm 4.2 is very similar to the matrix algorithm of Olshevsky et al. [51]. This is excellent demonstration of how block unfoldings can make generalizations of matrix algorithms to tensors very intuitive.

4.5.3 Examples

The following examples will help to make the ideas of this section and their implementation clearer.

Let $A \in \mathbb{R}^{6\times 6}$ be the following 3×3 block symmetric Toeplitz matrix with 2×2

upper triangular blocks:

$$A = \left[\begin{array}{cc|cc|cc} 6 & 9 & -1 & 2 & 4 & 16 \\ & 3 & & 3 & & 8 \\ \hline -1 & 2 & 6 & 9 & -1 & 2 \\ & 3 & & 3 & & 3 \\ \hline 4 & 16 & -1 & 2 & 6 & 9 \\ & 8 & & 3 & & 3 \end{array}\right].$$

Then the bases $\mathscr{G}_\alpha$ and $\mathscr{G}_\beta$ are

$$\mathscr{G}_\alpha = \left\{ \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix} \right\}$$

$$\mathscr{G}_\beta = \left\{ \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}, \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \right\}$$

and

$$a_\alpha = [\,3\ \ 1\ \ 4\,]$$

$$a_\beta^1 = [\,2\ \ 1\ \ 3\,], \quad a_\beta^2 = [\,-1\ \ 3\ \ 2\,], \quad a_\beta^3 = [\,1\ \ 2\ \ 4\,].$$

Therefore

$$\tilde{A} = \begin{bmatrix} 6 & 3 & 9 \\ -1 & 3 & 2 \\ 4 & 8 & 16 \end{bmatrix}.$$

It is now easy to show that $G_\alpha \tilde{A} G_\beta^T = \mathscr{R}(A)$.

As an example of a tensor structure class that frequently arises [55, 1], a tensor $\mathcal{T} \in \mathbb{R}^{\mathbf{n}}$ is called a *Toeplitz tensor* if

$$\mathcal{T}(\mathbf{i}) = \mathcal{T}(\mathbf{i} + k) \tag{4.35}$$

for all k such that $1 \leq k \leq \min(\mathbf{n} - \mathbf{i})$. We denote this as $\mathcal{T} \in \mathbb{T}^{\mathbf{n}}$.

We also say that $\mathcal{T}$ is a *Hankel tensor* if the value of $\mathcal{T}(\mathbf{i})$ depends only on $i_1 + i_2 + \cdots + i_d$ and denote this as $\mathcal{T} \in \mathbb{H}^{\mathbf{n}}$.

As an example, the $3 \times 3 \times 3$ tensor $\mathcal{A}$ given by

$$\mathcal{A}(:,:,1) = \begin{bmatrix} 5 & -2 & 8 \\ 7 & 0 & 3 \\ 9 & 12 & -1 \end{bmatrix}, \quad \mathcal{A}(:,:,2) = \begin{bmatrix} 1 & 13 & 23 \\ 4 & 5 & -2 \\ -8 & 7 & 0 \end{bmatrix},$$

$$\mathcal{A}(:,:,3) = \begin{bmatrix} -6 & 21 & -3 \\ 11 & 1 & 13 \\ 15 & 4 & 5 \end{bmatrix},$$

is a Toeplitz tensor. For example $\mathcal{A}(1,1,1) = \mathcal{A}(2,2,2) = \mathcal{A}(3,3,3) = 5$ and $\mathcal{A}(1,2,1) = \mathcal{A}(2,3,2) = -2$. Note that the matrix

$$\mathcal{A}_{(1)} = [\, \mathcal{A}(:,:,1) \mid \mathcal{A}(:,:,2) \mid \mathcal{A}(:,:,3) \,]$$

is not a Toeplitz matrix.

As another example, suppose $\mathcal{A}$ is a $6 \times 4 \times 10$ tensor such that

$$\mathcal{A} \in \mathbb{T}^{[3\ 2\ 5]} \otimes \mathbb{H}^{[2\ 2\ 2]},$$

i.e. $\mathcal{A}$ is a $3 \times 2 \times 5$ block-Toeplitz tensor with $2 \times 2 \times 2$ Hankel blocks. Let the tensors $\mathcal{B}_1, \mathcal{B}_2 \in \mathbb{R}^{3\times 2\times 5}$ and $\mathcal{C}_1, \mathcal{C}_2 \in \mathbb{R}^{2\times 2\times 2}$ solve (4.32) for $r = 2$. Then

$$\mathcal{B}_1, \mathcal{B}_2 \in \mathbb{T}^{[3\ 2\ 5]} \quad \text{and} \quad \mathcal{C}_1, \mathcal{C}_2 \in \mathbb{H}^{[2\ 2\ 2]}.$$

4.6 Conclusions

The tensor Kronecker product and tensor Kronecker SVD developed in this chapter have many of the same properties as their matrix counterparts. In particular, the TKSVD can be used to approximate a tensor with a tensor Kronecker product or a sum of such terms in an optimal way and the terms in this expansion have many of the same structure and symmetries as the original tensor. Thus we can form a data-sparse Kronecker approximation of a large, structured tensor and many important operations, such as computing norms, multilinear products or the HOSVD, can be efficiently performed on this approximation.

CHAPTER 5

ADDITIONAL TOPICS AND FUTURE RESEARCH

In this chapter we cover a few topics which do not fit in other chapters but are related to the theme of this thesis of extending matrix methods to tensors and exploiting structure. In §5.1 we define new a tensor decomposition which is similar to the HOSVD and is based on the matrix QR decomposition and its uses for low-rank tensor approximations are investigated. In §5.2 we explore some of the properties of partially symmetric tensors and propose a higher-order power method that exploits their symmetries.

5.1 The Higher-Order QR Decomposition

In matrix computations, the QR decomposition with partial pivoting can often be used instead of the full singular value decomposition for such things as rank determination and low-rank approximation [66]. Here we consider one way of extending the QR decomposition to tensors that is similar to how the HOSVD extends the matrix SVD and we compare their effectiveness for low-rank tensor approximation.

5.1.1 QR with Partial Pivoting

We first briefly review some of the properties of the QR decomposition with partial pivoting. See [26, Chapter 5.4] for a more detailed discussion.

For an $m \times n$ matrix A the QR factorization with partial pivoting has the form

$$A \;=\; QRP^T,$$

where $Q \in \mathbb{R}^{m\times m}$ is orthogonal, $P \in \mathbb{R}^{n\times n}$ is a permutation matrix and $R \in \mathbb{R}^{m\times n}$ is upper triangular. Additionally, we have the property that for every $j \leq \min(m,n)$

$$|\, R(j,j)\,|^2 \;\geq\; \|\, R(j{:}m,\ell)\,\|^2, \quad \forall \ell > j \tag{5.1}$$

which means that if we partition R into

$$\begin{bmatrix} R_{11}^{(j)} & q^{(j)} & R_{12}^{(j)} \\ 0 & r_{jj} & w^{(j)} \\ 0 & 0 & R_{22}^{(j)} \end{bmatrix}$$

and define

$$Z^{(j)} \;=\; \begin{bmatrix} w^{(j)} \\ R_{22}^{(j)} \end{bmatrix}$$

then $|r_{jj}|$ is greater than or equal the norms of the colums in $Z^{(j)}$.

5.1.2 Definition and Properties

The *Higher-Order QR Decomposition* (HOQRD) is a Tucker decomposition which for a tensor $\mathcal{A} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ is

$$\mathcal{A} \;=\; (Q_1,\ldots,Q_d)\cdot\mathcal{R} \tag{5.2}$$

where $\mathcal{R} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ is the *core tensor* and the matrices $Q_i \in \mathbb{R}^{n_i\times n_i}$ are orthogonal. Note that the core tensor $\mathcal{R}$ is neither triangular nor "supertriangular" and neither are its modal unfoldings $\mathcal{R}_{(k)}$. However, due to the pivoting strategy, the $\mathcal{R}$ tensor has a lot of its "mass" concentrated near the $(1,1,\ldots,1)$ entry.

For $k = 1, \ldots, d$, the matrix Q_k is defined as Q-matrix of the QR decomposition with partial pivoting of the modal unfolding $\mathcal{A}_{(k)}$, i.e.

$$\mathcal{A}_{(k)} \;=\; Q_k R_k P_k^T \tag{5.3}$$

where P_k is a permutation matrix and R_k is upper triangular.

We have written a Matlab function called `hoqrd` which computes the HOQRD. If $\mathcal{A}$ is stored in the `tensor` object `A` then the command

```
[R,Q] = hoqrd(A);
```

returns a `tensor` object `R` which stores the tensor $\mathcal{R}$ and a cell array `Q` such that `Q{i}`$= Q_i$. See Appendix A for more details.

Similar to equation (1.40) for the HOSVD, it can be shown that

$$\mathcal{R}_{(k)} \;=\; R_k P_k^T (Q_d \otimes Q_{d-1} \otimes \cdots \otimes Q_{k+1} \otimes Q_{k-1} \otimes \cdots \otimes Q_1) \tag{5.4}$$

where $R_k \in \mathbb{R}^{n_k \times N_k}$ has property (5.1), and $N_k = \prod_{i \neq k} n_i$. Note that (5.4) can be written as

$$\mathcal{R}_{(k)} \;=\; R_k X$$

where X is an $N_k \times N_k$ orthogonal matrix. Therefore the row-norms of the matrices $\mathcal{R}_{(k)}$ and R_k are equal.

By equation (5.1) we have that for any $1 \leq j \leq n_k$ and $\ell \geq j+1$

$$|\, R_k(j,j) \,|^2 \;\geq\; \|\, R_k(j\!:\!n_k, \ell) \,\|_2^2$$

and thus

$$(N_k - j)\, |\, R_k(j,j) \,|^2 \;\geq\; \|\, R_k(j\!:\!n_k, j\!+\!1\!:\!N_k) \,\|_F^2.$$

By the row-norm equivalence of R_k and $\mathcal{R}_{(k)}$, and the tridiagonality of R_k, we know that

$$\| \mathcal{R}_{(k)}(j{+}1 : n_k, :) \|_F^2 \;=\; \| R_k(j{+}1 : n_k, j{+}1 : N_k) \|_F^2$$

and

$$| R_k(j,j) |^2 \;\leq\; \| R_k(j, j : N_k) \|_2^2 \;=\; \| R_k(j, :) \|_2^2 \;=\; \| \mathcal{R}_{(k)}(j, :) \|_2^2.$$

Combining these facts, we have

$$\begin{aligned}
\| \mathcal{R}_{(k)}(j, :) \|_2^2 \;&\geq\; | R_k(j,j) |^2 \\
&\geq\; \frac{1}{N_k - j} \| R_k(j : n_k, j{+}1 : N_k) \|_F^2 \\
&\geq\; \frac{1}{N_k - j} \| R_k(j{+}1 : n_k, j{+}1 : N_k) \|_F^2 \\
&\geq\; \frac{1}{N_k - j} \| \mathcal{R}_{(k)}(j{+}1 : n_k, :) \|_F^2
\end{aligned}$$

i.e.

$$\| \mathcal{R}_{(k)}(j, :) \|_2 \;\geq\; \frac{1}{\sqrt{N_k - j}} \| \mathcal{R}_{(k)}(j{+}1 : n_k, :) \|_F \tag{5.5}$$

which has the weaker form

$$\| \mathcal{R}_{(k)}(j, :) \|_2 \;\geq\; \frac{1}{\sqrt{N_k - j}} \| \mathcal{R}_{(k)}(\ell, :) \|_2, \quad \forall \ell > j. \tag{5.6}$$

Equation (5.5) is a general inequality that holds for any $n \times N$ matrix B with $N > n$ that has the form $B = RX$ where $R \in \mathbb{R}^{n \times N}$ is upper triangular and satisfies (5.1) and $X \in \mathbb{R}^{N \times N}$ is orthogonal, i.e. for any $j = 1, 2, \ldots, n-1$

$$\| B(j, :) \|_2 \;\geq\; \frac{1}{\sqrt{N - j}} \| B(j{+}1 : n, :) \|_F.$$

In fact, this inequality is tight. Consider the case where X is any orthogonal matrix and

$$R \;=\; \begin{bmatrix} 1 & 0 \\ 0 & u^T \end{bmatrix}$$

where $u = \mathbf{1}_{N-1}$ is the column vector of length $N-1$ with 1 in all entries. Then for $j = 1$ we have $\| B(1,:) \|_2 = 1$ and $\| B(2,:) \|_F = \sqrt{N-1}$.

5.1.3 The case $d = 2$

Suppose $\mathcal{A}$ is a matrix. The HOQRD in this case is

$$\begin{aligned} \mathcal{A} &= (Q_1, Q_2) \cdot \mathcal{R} \\ &= Q_1 \mathcal{R} Q_2^T \end{aligned}$$

where

$$\begin{aligned} A_{(1)} &= \mathcal{A} = Q_1 R_1 P_1^T \\ A_{(2)} &= \mathcal{A}^T = Q_2 R_2 P_2^T \end{aligned}$$

and

$$\begin{aligned} \mathcal{R}_{(1)} &= \mathcal{R} = R_1 P_1^T Q_2 \\ \mathcal{R}_{(2)} &= \mathcal{R}^T = R_2 P_2^T Q_1 \end{aligned}$$

Note that these equations are consistent, since $Q_1 R_1 P_1^T = P_2 R_2^T Q_2^T$.

We can now replace $\mathcal{R}$ in the equation $\mathcal{A} = Q_1 \mathcal{R} Q_2^T$ with $R_1 P_1^T Q_2$ so the HOQRD reduces to

$$\mathcal{A} = Q_1 (R_1 P_1^T Q_2) Q_2^T = Q_1 R_1 P_1^T$$

which is the regular matrix QR with column pivoting. But the core tensor $\mathcal{R}$ did *not* reduce to the upper-triangular matrix R. This is in contrast to the HOSVD where the core tensor $\mathcal{S}$ reduces to the diagonal matrix Σ.

5.1.4 Truncated HOQRD and the HOPM

We have established in (5.5) and (5.6) that a "norm grading" is in effect on the rows of $\mathcal{R}_{(k)}$; the rows tend to have smaller norms as we go down the matrix. As this holds for any mode k, the entries in the tensor $\mathcal{R}$ tend to generally become smaller in absolute value the further away from the $(1, 1, \ldots, 1)$ position they are.

With this grading in mind it is not unreasonable to expect that truncating the HOQRD would give a decent low-rank approximation to $\mathcal{A}$ in the same way as the HOSVD does. Indeed, the matrix QR decomposition with pivoting can be used to compute a rank-j approximation to an $m \times n$ matrix A with the formula

$$\hat{A} \;=\; Q(:, 1{:}j)\, R(1{:}j, :)\, P^T. \tag{5.7}$$

Then we have [26, 66]

$$\| A - \hat{A} \|_F^2 \;=\; \| R(j{+}1{:}m, j{+}1{:}n) \|_F^2 \;\leq\; (n-j)\, |R(j{+}1, j{+}1)|^2. \tag{5.8}$$

If $| R(j{+}1, j{+}1) |$ is small then the rank-j approximation $\hat{A}$ will be reasonably close to A in the Frobenius norm.

The inequality (5.8) can be generalized to the HOQRD. Consider the multilinear rank-$(m_1, m_2, \ldots, m_d)$ truncation

$$\hat{\mathcal{A}} \;=\; (Q'_1, Q'_2, \ldots, Q'_d) \cdot \mathcal{R}' \tag{5.9}$$

where $\mathcal{R}' = \mathcal{R}(1{:}m_1, 1{:}m_2, \ldots, 1{:}m_d)$ and $Q'_i = Q_i(:, 1 : m_i)$ for $i = 1, \ldots, d$. Then by (5.6) we have

$$\begin{aligned} \| \mathcal{A} - \hat{\mathcal{A}} \|^2 \;&\leq\; \sum_{j=1}^{d} (n_j - m_j)\, | R_j(m_j{+}1, m_j{+}1) |^2 \\ &\leq\; \sum_{j=1}^{d} (n_j - m_j)\, \| \mathcal{R}_{(j)}(m_j{+}1, :) \|_2^2 \end{aligned}$$

Approximations such as (5.9) are usually used as initial guesses for iterative algorithm such as the HOPM or HOOI [21]. The most widely used initial guess is a truncated HOSVD, but a truncated HOQRD is also possible as the examples below show.

Example: (From [21].) Consider a tensor $\mathcal{B} \in \mathbb{R}^{2\times2\times2\times2}$, which has the nonzero entries

$$\begin{cases} b_{1111} = 25.1, & b_{1212} = 25.6, \\ b_{2121} = 24.8, & b_{2222} = 23. \end{cases}$$

Because of the symmetries $b_{ijk\ell} = b_{kji\ell}$ and $b_{ijk\ell} = b_{i\ell kj}$ the best rank-1 approximation of $\mathcal{B}$ is of the form $\lambda u \circ w \circ u \circ w$ where $u, w \in \mathbb{R}^2$ are unit vectors. Write $u = (\cos\alpha,\ \sin\alpha)^T$ and $w = (\cos\beta,\ \sin\beta)^T$. The function $f_1 = \mathcal{B} \cdot (u, w, u, w)$ can be expanded to

$$\begin{aligned} f_1(\alpha,\beta) \ = \ & 25.1\cos^2\alpha\cos^2\beta + 25.6\cos^2\alpha\sin^2\beta \\ & + 24.8\sin^2\alpha\cos^2\beta + 23\sin^2\alpha\sin^2\beta. \end{aligned}$$

The global maximum is 25.6, when $\alpha = 0, \beta = \pi/2$. Thus the best rank-1 approximation has $\lambda = 25.6$ and $u = (1,\ 0)^T, w = (0,\ 1)^T$. Note that

$$\mathcal{B}_{(1)} = \begin{bmatrix} 25.1 & 0 & 0 & 0 & 0 & 25.6 & 0 & 0 \\ 0 & 0 & 24.8 & 0 & 0 & 0 & 0 & 23 \end{bmatrix},$$

and

$$\mathcal{B}_{(2)} = \begin{bmatrix} 25.1 & 0 & 0 & 0 & 0 & 24.8 & 0 & 0 \\ 0 & 0 & 25.6 & 0 & 0 & 0 & 0 & 23 \end{bmatrix}.$$

so it is easy to show that the truncated HOSVD yields the approximation with $u = w = (1,\ 0)^T$, which does belong the HOPM attraction region of the global

maximum. However, the truncated HOQRD will give $u = (1,\ 0)^T$ and $w = (0,\ 1)^T$ as an initial guess, which happens to be the global maximum.

Next consider the tensor $\mathcal{A}$ which is equal to $\mathcal{B}$ except

$$a_{1121} \ = \ 0.3, \quad a_{2111} \ = \ 0.3,$$

which has similar symmetries as $\mathcal{B}$. The function $f_2 = \mathcal{A} \cdot (u, w, u, w)$ is given by

$$f_2(\alpha, \beta) \ = \ f_1(\alpha, \beta) + 0.6 \cos^2 \beta \cos \alpha \sin \alpha$$

and has the same global maximum as f_1 but now also has a local maximum at $(0.5536,\ 0)$. Truncation of the HOSVD yields vectors u and w that are very close to $(1,\ 0)^T$ and the HOPM with this initial guess will converge to the local maximum and not the global. The truncated HOQRD will again yield $u = (1,\ 0)^T, w = (0,\ 1)^T$, i.e. the global maximum.

Example: To test numerically how the truncated HOQRD compares to the truncated HOSVD and against randomly generated guesses for generic tensors, we created 1000 tensors of size $10 \times 10 \times 10$ with entries randomly chosen from a normal distribution with mean zero and standard deviation 1. We then computed rank-1 approximations to each of these tensors using truncated HOQRD and HOSVD decompositions and also a randomly generated rank-1 approximation, whose entries are from the same normal distribution.

The results show that the truncated HOSVD beat the truncated HOQRD in about 70% of the cases, reflecting its stronger norm grading properties. On the other hand, the truncated HOQRD was better than a random guess in 83% of cases.

5.2 Partially Symmetric Tensors

If tensor $\mathcal{A}$ has one or several $\mathbf{p}$-symmetries as described in Definition 4.9 then many decompositions and properties of $\mathcal{A}$ reflect those symmetries. In addition, it is possible to exploit the symmetry structure in algorithms, such as higher-order power methods, to save on computational cost.

If $\mathcal{A}$ has one or more $\mathbf{p}$-symmetries but is not necessarily fully symmetric we say that $\mathcal{A}$ is *partially symmetric*.

5.2.1 HOSVD and Symmetries

The HOSVD has a special structure when $\mathcal{A}$ is partially symmetric. The following lemma originally appeared in [15].

Lemma 5.1. *If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is $\mathbf{p}$-symmetric and $\mathcal{A} = (U_1, \ldots, U_d) \cdot \mathcal{S}$ is the HOSVD then $\mathcal{S}$ is $\mathbf{p}$-symmetric and if i and j are in the same cycle in the permutation $\mathbf{p}$ then $U_i = U_j$.*

Proof. First note that $\mathcal{A} = (U_1, \ldots, U_d) \cdot S$ is equivalent to

$$\mathcal{A}(\mathbf{i}) \;=\; \sum_{\mathbf{j=1}}^{\mathbf{n}} \mathcal{S}(\mathbf{i}) U_1(i_1, j_1) \cdots U_d(i_d, j_d)$$

for all $\mathbf{1} \leq \mathbf{i} \leq \mathbf{n}$. Hence

$$\begin{aligned}
(\mathcal{A}^{<\mathbf{p}>})(\mathbf{i}) \;&=\; \mathcal{A}(\mathbf{i}(\mathbf{p})) \\
&=\; \sum_{\mathbf{j=1}}^{\mathbf{n}} \mathcal{S}(\mathbf{j}) U_1(i_{p_1}, j_1) \cdots U_d(i_{p_d}, j_d) \\
&=\; \sum_{\mathbf{k=1}}^{\mathbf{n}} \mathcal{S}^{<\mathbf{p}>}(\mathbf{k}) U_{p_1}(i_1, k_1) \cdots U_{p_d}(i_d, k_d)
\end{aligned}$$

which is also the **i**th entry in the tensor $(U_{p_1}, \ldots, U_{p_d}) \cdot \mathcal{S}^{<\mathbf{p}>}$. Hence

$$((U_1, \ldots, U_d) \cdot \mathcal{S})^{<\mathbf{p}>} \;=\; (U_{p_1}, \ldots, U_{p_d}) \cdot \mathcal{S}^{<\mathbf{p}>}. \tag{5.10}$$

Now, suppose i and j are in the same cycle in $\mathbf{p}$. Then there exists a power $\mathbf{q}$ of $\mathbf{p}$, i.e. $\mathbf{q} = \mathbf{p}(\mathbf{p}(\cdots(\mathbf{p})))$, such that $\mathbf{q}(i) = j$. Note that $\mathcal{A}^{<\mathbf{q}>} = \mathcal{A}$. Thus $\mathcal{A}_{(i)}$ is a column permutation of $\mathcal{A}_{(j)}$ which implies $U_i = U_j$. This means that $U_{p_k} = U_k$ for all $k = 1, 2, \ldots, d$. Combined with (5.10) this completes the proof. □

This result generalizes readily to the case where $\mathcal{A}$ has multiple symmetries.

Corollary 5.2. *Suppose $\mathcal{A}$ is $\mathbf{p}_1$-,$\mathbf{p}_2$-,... and $\mathbf{p}_k$-symmetric and the HOSVD of $\mathcal{A}$ is given by $\mathcal{A} = (U_1, \ldots, U_d) \cdot \mathcal{S}$. Then the core tensor $\mathcal{S}$ is also $\mathbf{p}_1$-,$\mathbf{p}_2$-,... and $\mathbf{p}_k$-symmetric. Furthermore, if i and j are in the same cycle in any of $\mathbf{p}_1, \ldots, \mathbf{p}_k$ or are in cycles that intersect, then $U_i = U_j$.*

Example: Suppose $\mathcal{A} \in \mathbb{R}^{n_1 \times n_2 \times n_3 \times n_4}$ where for all $\mathbf{i} = \mathbf{1}, \ldots, \mathbf{n}$

$$\mathcal{A}(i_1, i_2, i_3, i_4) \;=\; \mathcal{A}(i_2, i_1, i_4, i_3) \;=\; \mathcal{A}(i_3, i_4, i_1, i_2),$$

i.e. $\mathcal{A}$ has $\mathbf{p}_1 = [\,2\ 1\ 4\ 3\,] = (1\ 2)(3\ 4)$ and $\mathbf{p}_2 = [\,3\ 4\ 1\ 2\,] = (1\ 3)(2\ 4)$ symmetries. If $\mathcal{A} = (U_1, U_2, U_3, U_4) \cdot \mathcal{S}$ is the HOSVD then by Corollary 5.2 we have $U_1 = U_2$, $U_3 = U_4$ and $U_1 = U_3$ and $U_2 = U_4$, which of course means $U_1 = U_2 = U_3 = U_4$. Furthermore, the core tensor $\mathcal{S}$ is $\mathbf{p}_1$- and $\mathbf{p}_2$-symmetric. Note that $\mathcal{S}$ is *not* a fully symmetric tensor.

5.2.2 Singular Vectors

We hypothesize that results similar to Lemma 5.1 hold for the singular vectors of a partially symmetric tensor $\mathcal{A}$.

Conjecture 5.3. *If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is* **p***-symmetric and σ is a Z-singular value of $\mathcal{A}$ then there exist corresponding unit 2-norm singular vectors $u_1, \ldots, u_d$ such that $u_i = u_{p_i}$ for all $i = 1, 2, \ldots, d$.*

The condition $u_i = u_{p_i}, \forall i$ is equivalent to $u_i = u_j$ if i and j are in the same cycle in $\mathbf{p}$. Furthermore, it is easy to show that if $u_1, \ldots, u_d$ are singular vectors corresponding to σ, then so are $v_1, \ldots, v_d$ where $v_i = u_{p_i}$.

Rephrasing Conjecture 5.3, it states that at any critical value σ of the Rayleigh quotient

$$\phi_{\mathcal{A}}(x_1, \ldots, x_d) = \frac{\mathcal{A} \cdot (x_1, \ldots, x_d)}{\| x_1 \|_2 \cdots \| x_d \|_2}$$

we can choose corresponding critical points $u_1, \ldots, u_d$, normalized to unit 2-norm, such that $u_i = u_{p_i}$ for all i.

Note that since Z-singular values are not normalized to be nonnegative as matrix singular values are, the possible sign difference of the left and right singular vectors of a symmetric matrix, i.e. $u_i = \pm v_i$, is not a problem in the tensor case.

5.2.3 A Power Method for Partially Symmetric Tensors

Conjecture 5.3 suggests that in a power method for computing optimal rank-1 approximations of a **p**-symmetric tensor we only need to compute one vector for

each cycle in $\mathbf{p}$. We propose one such algorithm: the *Shifted Partially Symmetric Higher-Order Power method* (SPS-HOPM) which is shown in Algorithm 5.1. This algorithm can be viewed as a kind of halfway point between the HOPM and the and SS-HOPM discussed in Chapter 2.

In terms of flops, if $\mathcal{A}$ is a cubical order-d tensor of size $n \times n \times \cdots \times n$ then a single iteration of the HOPM requires approximately $2d(d-1)n^d$ flops. If $\mathcal{A}$ is fully symmetric, then a single iteration of the S-HOPM is $2(d-1)n^d$ flops. If $\mathcal{A}$ is $\mathbf{p}$-symmetric and $\mathbf{p}$ has k cycles then one iteration of the SPS-HOPM requires $2k(d-1)n^d$ flops.

Algorithm 5.1 Shifted partially symmetric HOPM (SPS-HOPM)

Given a $\mathbf{p}$-symmetric, order-d tensor $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$, where $\mathbf{p}$ is a permutation of $1{:}d$ with k distinct cycles $\mathbf{c}_1, \ldots, \mathbf{c}_k$ where $\mathbf{c}_i$ is of length ℓ_i. Let $\mathbf{s}$ be an index vector of length k such that $s_j \in \mathbf{c}_j$.

Require: $u_j^{(0)} \in \mathbb{R}^{n_j}$ with $\|u_j^{(0)}\|_2 = 1$ and $u_j^{(0)} = u_{p_j}^{(0)}$. Set $\sigma^{(0)} = \mathcal{A} \cdot (u_1^{(0)}, \ldots, u_d^{(0)})$.

Require: $\alpha > 0$.

for $i = 0, 1, \ldots$ **do**
  for $j = 1, 2, \ldots, k$ **do**
    $\hat{u}_{s_j}^{(i+1)} \leftarrow \mathcal{A} \cdot (u_1^{(i)}, \ldots, u_{s_j-1}^{(i)}, I_{n_{s_j}}, u_{s_j+1}^{(i)}, \ldots, u_d^{(i)}) + \alpha u_{s_j}^{(i)}$
    for $t = 1, 2, \ldots, \ell_j$ **do**
      $u_{\mathbf{c}_j(t)}^{(i+1)} \leftarrow \hat{u}_{s_j}^{(i+1)} / \|\hat{u}_{s_j}^{(i+1)}\|_2$
    end for
  end for
  $\sigma^{(i+1)} \leftarrow \mathcal{A} \cdot (u_1^{(i+1)}, \ldots, u_d^{(i+1)})$
end for

The SPS-HOPM as stated in Algorithm 5.1 is not guaranteed to converge when $\alpha = 0$ as the following example shows.

If $\mathcal{A} \in \mathbb{R}^{2\times2\times2\times2}$ is the $[\,3\ 1\ 2\ 4\,]$-symmetric tensor given by

$$\mathcal{A}_{[1\ 2]\times[3\ 4]} = \begin{bmatrix} 0.3649 & 0.7760 & 1.0487 & 0.5099 \\ 0.7760 & -0.3934 & 0.5099 & 0.4080 \\ 0.7760 & -0.3934 & 0.5099 & 0.4080 \\ -0.3934 & -1.5691 & 0.4080 & -0.6190 \end{bmatrix}$$

and the initial guess is given by a one-term truncation of the HOSVD then the SPS-HOPM with $\alpha = 0$ will eventually start to oscillate between $\sigma = 0.8095$ and $\sigma = 0.7959$.

We have created a Matlab function `spshopm` that implements the SPS-HOPM. If the $\mathbf{p}$-symmetric tensor $\mathcal{A} \in \mathbb{R}^{n_1\times\cdots\times n_d}$ is stored in the `tensor` object `A` then the `spshopm` function computes a Z-singular value and associate singular vectors of $\mathcal{A}$ with the following commands.

```
[sigma,U] = spshopm(A,p);
%Specify initial guess U0:
[sigma,U] = spshopm(A,p,'Start',U0);
%Specify shift alpha:
[sigma,U] = spshopm(A,p,'Shift',alpha);
```

If $\mathbf{p}$ has k distinct cycles then the output `sigma` is a scalar and `U` is a cell array of length k that contains the unique u-vectors. Note that if a starting value `U0` is specified, then `U0` is also a length-k cell array containing the unique u_0 starting vectors. See Appendix A for further details.

It is an open question if choosing a sufficiently large shift for SPS-HOPM, or modifying it in some other way, can guarantee convergence. We note that the SPS-HOPM bears some resemblance to the the algorithm of [7] for finding the

unique positive singular value of a nonnegative irreducible rectangular tensor $\mathcal{A}$. Specifically, $\mathcal{A} \in \mathbb{R}^{k_1 \times \cdots \times k_d}$ is said to be a *rectangular tensor* if $d = p + q$ and $k_1 = \cdots = k_p = m$, $k_{p+1} = \cdots = k_{p+q} = n$ and furthermore $\mathcal{A}$ is unchanged under any permutation of the first p modes and under any permutation of the last q modes. Rectangular tensors are thus partially symmetric. The singular value definition used by [7, 8, 75] is similar to, yet different from C-singular values. However if Conjecture 5.3 holds for C-singular values then the two definitions would be essentially equivalent.

APPENDIX A

MATLAB CODE DETAILS

Here we will go into the details of the Matlab software used throughout this thesis. The Matlab Tensor Toolbox is available at [40] and the specialized Matlab code we use is available at `http://www.cam.cornell.edu/~stefan/code`.

A.1 The Tensor Toolbox

First we will review the basic properties of the Matlab Tensor Toolbox by Bader and Kolda [40] which we have found indispensable for tensor computations in the Matlab programming language. For a more in-depth introduction we refer to [39].

A.1.1 Classes

The Tensor Toolbox provides several new Matlab classes that represent various types of tensors, the basic one being `tensor`. These classes and their purpose are listed in Table A.1.

Table A.1: Tensor Toolbox classes for representing tensors

Class	Description
`tensor`	Represents a tensor.
`sptensor`	Represents a sparse tensor.
`ktensor`	Represents a CP-decomposed tensor.
`ttensor`	Represents a Tucker-decomposed tensor.

The `ktensor` and `ttensor` classes store the tensor in decomposed form, `ktensor` stores the CP-decomposition parameters λ_i and $b_j^{(i)}$ defined in (1.36) and `ttensor` stores the Tucker decomposition parameters $\mathcal{C}$ and X_i defined in (1.37).

The class `tensor` is the one we shall use most. As such, we list the fields of this class in Table A.2.

Table A.2: Fields of the `tensor` class

Field	Description
`data`	The multidimensional array that the `tensor` class encapsulates.
`size`	The size vector.

A.1.2 Tensor Operations

Each of the classes in Table A.1 provides optimized versions of several important operations that are not possible for standard Matlab multidimensional arrays. Such operations include computing norms, contractions, outer products and multilinear products in addition to such elementary operations as addition, subtraction and scalar multiplication. We list some of the most important tensor operations in Table A.3. Note that some operations might not be possible for tensors stored in decomposed form.

Table A.3: Tensor operations provided by the Tensor Toolbox.

Operation	Description
`innerprod`	Computes the inner product of two tensors.
`norm`	Computes the norm of a single tensor.
`ttm`	Perform the multilinear product of a tensor with matrices in one or more modes.
`ttv`	Perform the multilinear product of a tensor with vectors in one or more modes.
`ttsv`	Perform the multilinear product of a tensor with a single vector in one or more modes.
`ttt`	Contracted product of two tensors along one or more modes, or perform a tensor outer product (if no modes specified).

As an example, the following code creates tensors $\mathcal{A} \in \mathbb{R}^{2\times2\times2}, \mathcal{B} \in \mathbb{R}^{2\times4\times2}$, matrices $B_1, B_2 \in \mathbb{R}^{2\times2}$ and vectors $u, v, w \in \mathbb{R}^2$ with randomly generated entries

between 0 and 1. It then computes the contracted tensor $\mathcal{T} \in \mathbb{R}^{2\times2\times4\times2}$ such that

$$\mathcal{T}(i_1, i_2, i_3, i_4) = \sum_{k_1=1}^{2} \mathcal{A}(i_1, k_1, i_2) \cdot \mathcal{B}(k_1, i_3, i_5)$$

and the quantities

$$\mathcal{C} = \mathcal{A} \circ \mathcal{B}, \qquad \mathcal{E} = \mathcal{A} \cdot (B_1, B_2, I_2), \qquad \alpha = \mathcal{A} \cdot (u, v, w), \qquad \beta = \mathcal{A} \cdot (u, u, u).$$

```
A = tensor(rand(2,2,2));
B = tensor(rand(2,4,2));
B_1 = rand(2,2); B_2 = rand(2,2);
u = rand(2,1); v = rand(2,1); w = rand(2,1);
T = ttt(A,B,2,1);
C = ttt(A,B);
E = ttm(A,{B_1,B_2},[1 2]);
alpha = ttv(A,{u,v,w});
beta = ttsv(A,u);
```

A.1.3 Unfoldings

The Tensor Toolbox has two classes that form tensor unfoldings and we list them in Table A.4

Table A.4: Tensor Toolbox classes for unfoldings

Class	Description
`tenmat`	Forms and represents an unfolding of a `tensor` object.
`sptenmat`	Forms and represents an unfolding of a `sptensor` object.

Specifically, if `A` is a tensor object and `r` and `c` are vectors then `tenmat(A,r,c)` forms the unfolding $\mathcal{A}_{\mathbf{r}\times\mathbf{c}}$ and if `i` is an integer then `tenmat(A,i)` forms the mode-i unfolding $\mathcal{A}_{(i)}$.

As an example, the following code creates a randomly generated tensor $\mathcal{A} \in \mathbb{R}^{4\times2\times5\times3}$ with entries chosen from a $N(0,1)$ distribution and then forms the unfoldings $M_1 = \mathcal{A}_{[3\ 1]\times[2\ 4]}$ and $M_2 = \mathcal{A}_{(3)}$.

```
A = tensor(randn(4,2,5,3));
M_1 = tenmat(A,[3 1],[2 4]);
M_2 = tenmat(A,3);
```

We list fields of the `tenmat` class in Table A.5.

Table A.5: Fields of the `tenmat` class

Field	Description
`data`	The matrix data.
`tsize`	The size vector of the original tensor.
`rindices`	The vector **r** of modes mapped to rows.
`cindices`	The vector **c** of modes mapped to columns.

A.2 Block Tensor Classes

To facilitate fast prototyping of block tensor algorithms and simplify block-based algorithms we have created two new classes that extend the Tensor Toolbox and are listed in Table A.6. These classes provide all the same functionality as their non-blocked counterparts from the Tensor Toolbox.

Table A.6: Block tensor classes

Class	Description
`bltensor`	Represents a blocked tensor.
`bltenmat`	Forms and represents a block unfolding of a `bltensor` object.

A.2.1 bltensor

There are several differences between the `bltensor` and `tensor` classes. A `bltensor` object has a field `M` which is a cell array storing all the blocking vectors

of the tensor. Therefore, if `T` is a `bltensor` its blocking can be accessed by `T.M`. This blocking is automatically updated when operations such as `ttt` or `ttm` are performed on the tensor. A list of all the fields of a `bltensor` object is in Table A.7.

Table A.7: Fields of the `bltensor` class

Field	Description
`data`	A `tensor` object that the `bltensor` class encapsulates.
`M`	A cell array containing the modal blocking vectors.

There are several different ways of creating a `bltensor` object. The typical call would be

```
A_blocked = bltensor(A, M);
```

where `A` is a `tensor` object and `M` is a cell array containing the modal blocking vectors. For example, if $\mathcal{A} \in \mathbb{R}^{4\times5\times3}$ is blocked by $\mathbf{M} = \{[2\ 2], [1\ 3\ 1], [2\ 1]\}$ then we create the corresponding `bltensor` object with the following commands.

```
A = tensor(rand(4,5,3));
M = {[2 2],[1 3 1],[2 1]};
A_blocked = bltensor(A,M);
```

If the tensor blocked uniformly, i.e. every block has the same size, then it is sufficient to supply only the block size vector. For example, if $\mathcal{A} \in \mathbb{R}^{12\times10\times10}$ is a $3 \times 2 \times 2$ block tensor with $4 \times 5 \times 5$ blocks then we use the following commands.

```
   A = tensor(rand(12,10,10));
   m = [4 5 5];
   A_blocked = bltensor(A,m);
```

If the constructor is called without any blocking parameters, i.e. of the form `bltensor(A)`, then `M` is an empty cell array and each mode is considered to be a single block.

As mentioned earlier, the blocking vectors in `M` are automatically updated when tensor operations are performed. As an example, consider the case where $\mathcal{A} \in \mathbb{R}^{8\times 9\times 4}$ with blocking $\mathbf{M}_{\mathcal{A}} = \{[4\ \ 4], [3\ \ 2\ \ 4], [2\ \ 2]\}$ and $\mathcal{B} \in \mathbb{R}^{9\times 3\times 3}$ with blocking $\mathbf{M}_{\mathcal{B}} = \{[3\ \ 2\ \ 4], [2\ \ 1], [1\ \ 1\ \ 1]\}$. If $\mathcal{C} \in \mathbb{R}^{8\times 4\times 3\times 3}$ is defined by

$$\mathcal{C}(i_1, i_2, i_3, i_4) \ = \ \sum_{k_1} \mathcal{A}(i_1, k_1, i_2) \cdot \mathcal{B}(k_1, i_3, i_4)$$

then $\mathcal{C}$ inherits the blocking $\mathbf{M}_{\mathcal{C}} = \{[4\ 4], [2\ 2], [2\ 1], [1\ 1\ 1]\}$ from the corresponding modes in $\mathcal{A}$ and $\mathcal{B}$. This contraction is performed by `ttt` and the blocking is automatically created for $\mathcal{C}$. The following code demonstrates this.

```
   M_A = {[4 4],[3 2 4],[2 2]};
   A = bltensor(rand(8,9,4),M_A);
   M_B = {[3 2 4],[2 1],[1 1 1]};
   B = bltensor(rand(9,3,3),M_B);
   C = ttt(A,B,2,1);
>> cellfun(@disp,C.M)
     4     4
     2     2
     2     1
     1     1     1
```

A significant difference between `tensor` and `bltensor` is indexing. If $\mathbf{i} \in \mathbb{R}^d$ is an index vector and `A` is a `tensor` object, then `A(i)` returns the entry $\mathcal{A}(\mathbf{i})$. However, if `A` is a `bltensor` object, then `A(i)` returns the $\mathbf{i}$th block, $\mathcal{A}_{\mathbf{i}}$.

If we wish to access the $\mathbf{i}$th entry of the `bltensor` object `T` we use the command `T.data(i)`. The field `data` of a `bltensor` object is the underlying, unblocked `tensor` object.

The following example illustrates the different indexing schemes on a $4 \times 4 \times 4$ tensor that is also a $2 \times 2 \times 2$ block tensor.

```
   A = tensor(rand(4,4,4));
   A_blocked = bltensor(A,[2 2 2]);
>> A(1,1,1)
   ans =
       0.5921
>> A_blocked(1,1,1)
   ans is a tensor of size 2 x 2 x 2
    ans(:,:,1) =
         0.5921    0.7359
         0.3253    0.1566
     ans(:,:,2) =
         0.1012    0.9505
         0.2016    0.5321
>> A_blocked.data(1,1,1)
   ans =
       0.5921
```

A.2.2 bltenmat

Just as the `tenmat` class forms an unfolding of a `tensor` object, the `bltenmat` class forms block unfoldings of `bltensor` objects. The `bltensor` class is used in a very similar way to how `tenmat` is used. If the order-d tensor $\mathcal{A}$ has blocking $\mathbf{M}$ and $[\mathbf{r}\ \ \mathbf{c}]$ is a permutation of $1{:}d$ then

```
   A_blunfold = bltenmat(A,r,c);
```

is the $\mathcal{A}_{\mathbf{R}\times\mathbf{C}}$ block unfolding of $\mathcal{A}$, where $\mathbf{R} = \mathbf{M}(\mathbf{r})$ and $\mathbf{C} = \mathbf{M}(\mathbf{c})$. If $1 \leq i \leq d$ then the command

```
   A_blunfold = bltenmat(A,i);
```

forms the mode-i block unfolding, i.e. it is equivalent to the unfolding `bltenmat(A,[i],[1:i-1 i+1:d])`.

Similar to `bltensor`, the indexing of `bltenmat` is block-level. Thus `A_blunfold(i,j)` returns the (i, j) block of the block matrix $\mathcal{A}_{\mathbf{R}\times\mathbf{C}}$.

The fields of the `bltenmat` class are listed in Table A.8.

Table A.8: Fields of the `bltenmat` class

Field	Description
`data`	The matrix data.
`tsize`	The size vector of the original tensor.
`blsize`	Cell array containing the modal blocking vectors.
`rindices`	The vector $\mathbf{r}$ of modes mapped to rows.
`cindices`	The vector $\mathbf{r}$ of modes mapped to columns.

A.3 Power Methods

The various power methods described in Chapters 2 and 5 can be readily implemented in Matlab by using the Tensor Toolbox. Note that since the code for these methods is fairly long, we will sometimes not show unimportant parts of the code, mostly relating to displaying results, adding a `%[Code not shown]` comment in the appropriate places.

A.3.1 SS-HOPM

Provided with the Tensor Toolbox is an implementation of the SSHOPM [42] in the appropriately named Matlab function `sshopm`. This function has several options, but the following commands show some of the simplest uses, assuming $\mathcal{A}$ is a symmetric order-d size-N tensor stored in the `tensor` object `A`.

```
[lambda,x] = sshopm(A);
%Specify initial guess x0:
[lambda,x] = sshopm(A,'Start',x0);
%Specify shift alpha:
[lambda,x] = sshopm(A,'Shift',alpha);
```

The returned `lambda` is a scalar and `x` is a vector such that $\mathcal{A} \cdot (I_N, x, \ldots, x) = \lambda x$. The `sshopm` function makes effective use of Tensor Toolbox functionality such as the `ttsv` function. Note that if the shift is specified to be 0 then the SSHOPM becomes the SHOPM.

A.3.2 HOPM

Using the `sshopm` function as a baseline, we implement the HOPM [19] with the Matlab function `hopm`. If $\mathcal{A} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is stored in the `tensor` object `A` then the `hopm` computes a Z-singular value and associate singular vectors of $\mathcal{A}$ with the following commands.

```
[sigma,U] = hopm(A);
%Specify initial guess U0:
[sigma,U] = hopm(A,'Start',U0);
```

The output `sigma` is a scalar and `U` is a cell array of length d such that `U{i}`$= u_i$ for $i = 1, \ldots, d$ and together $\lambda, u_1, \ldots, u_d$ solve (2.31). Note that if a starting value is `U0` is specified that `U0` is also a length-d cell array such that `U0{i}`$= u_i^{(0)}$.

```
function [sigma,u,flag,its,u0,trace,tracex] = hopm(A,varargin)
%HOPM Power method for finding a real singular pair of a real tensor.
%
%  [SIGMA,U]=HOPM(A) finds an singular value (SIGMA) and singular vectors (U) for
%  the real tensor A such that A(u1,...,um)_{-k} = sigma*u_k.
%  [SIGMA,U]=HOPM(A,parameter,value,...) can specify
%  additional parameters as follows:
%   'MaxIts'   : Maximum power method iterations (Default: 1000)
%   'Start'    : Initial guess (Default: normal random vector)
%   'Tol'      : Tolerance on norm of change in |sigma| (Default: 1e-16)
%   'Display'  : Display every n iterations (Default: -1 for no display)
%  [SIGMA,U,FLAG]=HOPM(...) also returns a flag indicating convergence.
%      FLAG = 0  => Succesfully terminated
%      FLAG = -1 => Norm(X) = 0
%      FLAG = -2 => Maximum iterations exceeded
%  [SIGMA,U,FLAG,IT]=HOPM(...) also returns the number of  iterations.
%  [SIGMA,U,FLAG,IT,U0]=HOPM(...) also returns the initial guess.
%  [SIGMA,U,FLAG,IT,U0,TRACE]=HOPM(...) also returns a trace
%  of the sigma values at each iteration.
%
% See also SSHOPM,SJHOPM,PSHOPM.
% Stefan Ragnarsson, 2011.
% The MATLAB Tensor Toolbox is Copyright 2010, Sandia Corporation.

P = ndims(A); N = size(A);
%% Check inputs
p = inputParser;
p.addParamValue('MaxIts', 1000, @(x) x > 0);
p.addParamValue('Start', [], @(u) isequal(length(u),P));
p.addParamValue('Tol', 1.0e-16);
p.addParamValue('Display', -1, @isscalar);
p.parse(varargin{:});
%% Copy inputs
maxits  = p.Results.MaxIts;
u0      = p.Results.Start;
tol     = p.Results.Tol;
display = p.Results.Display;
%% Check starting vector
if isempty(u0)
  for i=1:P
    u0{i} = 2*rand(N(i),1)-1;
  end
end
min_u0_norm = norm(u0{1});
for i=2:P
  if norm(u0{i})<min_u0_norm
    min_u0_norm = norm(u0{i});
  end
end
if min_u0_norm < eps
  error('Zero starting vector');
end
%% Execute power method
if (display >= 0)
  %[Code not shown]
end
flag = -2;
for i=1:P
  u{i} = u0{i} / norm(u0{i});
end
sigma = ttv(A,u);
trace     = zeros(maxits,1);
trace(1)  = sigma;
tracex    = cell(maxits);
tracex{1} = u0;
```

```
for its = 1:maxits
  oldu = u;
  for i=1:P
    newu = double(ttv(A,u,-i));
    nnu = norm(newu);
    if nnu < eps,
      flag = -1;
      break;
    end
    newu = newu / nnu;
    u{i} = newu;
  end
  if flag == -1
    break;
  end
  newsigma = ttv(A,u);
  if norm(abs(newsigma-sigma)) < tol
    flag = 0;
  end
  if (display > 0) && ((flag == 0) || (mod(its,display) == 0))
   %[Code not shown]
  end
  sigma = newsigma;
  trace(its+1) = sigma;
  tracex{its+1} = u;
  if flag == 0
    break
  end
end
%% Check results
%[Code not shown]
```

A.3.3 SJ-HOPM

Algorithm 2.5, which we call the SJ-HOPM, is implemented in the Matlab function `sjhopm`. The function calls are very similar to the `sshopm` and `hopm` functions, as the following examples show.

```
    [sigma,U] = sjhopm(A);
    %Specify initial guess x0:
    [sigma,U] = sjhopm(A,'Start',U0);
    %Specify shift alpha:
    [sigma,U] = sjhopm(A,'Shift',alpha);
```

```
function [sigma,u,flag,its,u0,trace,tracex] = sjhopm(A,varargin)
%SJHOPM Shifted Jacobi power method for finding a real singular pair of a
%   real tensor.
%   [SIGMA,U]=SJHOPM(A) finds an eigenvalue (SIGMA) and eigenvectors (U)
%   for the real tensor A such that A(u1,...,um)_{-k} = sigma*u_k.
%   [SIGMA,U]=SJHOPM(A,parameter,value,...) can specify additional
%   parameters as follows:
%     'Shift'    : Shift in the singular value calculation (Default: 0)
%     'MaxIts'   : Maximum power method iterations (Default: 1000)
%     'Start'    : Initial guess (Default: normal random vector)
%     'Tol'      : Tolerance on norm of change in |sigma| (Default: 1e-16)
%     'Concave'  : Treat the problem as concave rather than convex.
%                  (Default: true for negative shift; false otherwise.)
%     'Display'  : Display every n iterations (Default: -1 for no display)
%   [SIGMA,U,FLAG]=SJHOPM(...) also returns a flag indicating convergence.
%      FLAG = 0  => Succesfully terminated
%      FLAG = -1 => Norm(X) = 0
%      FLAG = -2 => Maximum iterations exceeded
%   [SIGMA,U,FLAG,IT]=SJHOPM(...) also returns the number of iterations.
%   [SIGMA,U,FLAG,IT,X0]=SJHOPM(...) also returns the intial guess.
%   [SIGMA,U,FLAG,IT,X0,TRACE]=SJHOPM(...) also returns a trace of the
%   sigma values at each iteration.
%
%   See also SSHOPMC,SSHOPM,HOPM,SPSHOPM.
%
% Stefan Ragnarsson, 2011.
%MATLAB Tensor Toolbox. Copyright 2010, Sandia Corporation.

%% Error checking on A
d = ndims(A);
N = size(A);

%% Check inputs
p = inputParser;
p.addParamValue('Shift', 0);
p.addParamValue('MaxIts', 1000, @(x) x > 0);
p.addParamValue('Start', [], @(u) isequal(length(u),d));
p.addParamValue('Tol', 1.0e-16);
p.addParamValue('Display', -1, @isscalar);
p.addParamValue('Concave', -1);
p.parse(varargin{:});

%% Copy inputs
maxits  = p.Results.MaxIts;
u0      = p.Results.Start;
shift   = p.Results.Shift;
tol     = p.Results.Tol;
display = p.Results.Display;
concave = p.Results.Concave;

%% Check starting vector
if isempty(u0)
  for i=1:d
    u0{i} = 2*rand(N(i),1)-1;
  end
end
min_u0_norm = norm(u0{1});
for i=2:d
  if norm(u0{i})<min_u0_norm
    min_u0_norm = norm(u0{i});
  end
end
if min_u0_norm < eps
  error('Zero starting vector');
end
```

```
%% Check concavity
if concave == -1
  concave = (shift < 0);
end

%% Execute power method
if (display >= 0)
  %[Code not shown]
end

flag = -2;
normu0sq = 0;
for i=1:d
  normu0sq = normu0sq + norm(u0{i})^2;
end
for i=1:d
  u{i} = u0{i} / sqrt(normu0sq);
end
sigma = ttv(A,u);
trace     = zeros(maxits,1);
trace(1)  = sigma;
tracex    = cell(maxits);
tracex{1} = u0;

for its = 1:maxits
  for i=1:d
    newu{i} = double(ttv(A,u,-i) + shift * u{i});
    if (concave)
      newu{i} = -newu{i};
    end
  end
  for i=1:d
    nu(i,1) = norm(newu{i});
  end
  if min(nu) < eps,
    flag = -1;
    break;
  end
  nu_norm_sq = nu' * nu;
  for i=1:d
    newu{i} = newu{i} / sqrt(nu_norm_sq);
  end
  newsigma = ttv(A,newu);
  if norm(abs(newsigma-sigma)) < tol
    flag = 0;
  end
  if (display > 0) && ((flag == 0) || (mod(its,display) == 0))
   %[Code not shown]
  end
  u = newu;
  sigma = newsigma;
  trace(its+1) = sigma;
  tracex{its+1} = u;
  if flag == 0
    % Renormalize
    for i=1:d
      sigma = sigma*norm(u{i});
      u{i}  = u{i}/norm(u{i});
    end
    sigma = d^d * sigma;
    break
  end
end
%% Check results
%[Code not shown]
```

A.3.4 SPS-HOPM

Tthe SPS-HOPM listed in Algorithm 5.1 is implemented in the Matlab function `spshopm`. It uses the `permutation` class included with the Matgraph Toolbox [64] to decompose a permutation $\mathbf{p}$ into a product of cycles.

```
function [sigma,u_full,flag,its,u0,trace,tracex] = spshopm(A,p,varargin)
% SPSHOPM  Shifted p-Symmetric Higher Order Power Method.
% Use a shifted HOPM (a la the SSHOPM) for a p-symmetric tensor A. The
% symmetries are given as input. Only one vector per cycle of p is
% computed.
%
%   [SIGMA,U]=SPSHOPM(A,p) finds an eigenvalue (SIGMA) and eigenvectors (U)
%   for the real p-symmetric tensor A such that
%                A*(u1,...,um)_{-k} = sigma*u_k
%   [SIGMA,U]=SPSHOPM(A,p,parameter,value,...) can specify additional
%   parameters as follows:
%     'Shift'    : Shift in the eigenvalue calculation (Default: 0)
%     'MaxIts'   : Maximum power method iterations (Default: 1000)
%     'Start'    : Initial guess (Default: normal random vector)
%     'Tol'      : Tolerance on norm of change in |sigma| (Default: 1e-16)
%     'Display'  : Display every n iterations (Default: -1 for no display)
%   [SIGMA,U,FLAG]=SPSHOPM(...) also returns a flag indicating convergence.
%      FLAG = 0  => Succesfully terminated
%      FLAG = -1 => Norm(X) = 0
%      FLAG = -2 => Maximum iterations exceeded
%   [SIGMA,U,FLAG,IT]=SPSHOPM(...) also returns the number of iterations.
%   [SIGMA,U,FLAG,IT,U0]=SPSHOPM(...) also returns the intial guess.
%   [SIGMA,U,FLAG,IT,U0,TRACE]=SPSHOPM(...) also returns a trace of the
%   sigma values at each iteration.
%
%   See also SSHOPM,SSHOPMC,SJHOPM,HOPM.
% Stefan Ragnarsson, 2011.
% The MATLAB Tensor Toolbox is Copyright 2010, Sandia Corporation.

p   = permutation(p);
cyc = cycles(p);
L   = length(cyc);
n   = size(A);
d   = length(n);

% Check inputs
q = inputParser;
q.addParamValue('Shift', 0);
q.addParamValue('MaxIts', 1000, @(x) x > 0);
q.addParamValue('Start', [], @(u) isequal(length(u),L));
q.addParamValue('Tol', 1.0e-16);
q.addParamValue('Display', -1, @isscalar);
q.parse(varargin{:});
% Copy inputs
maxits  = q.Results.MaxIts;
shift   = q.Results.Shift;
u0      = q.Results.Start;
tol     = q.Results.Tol;
display = q.Results.Display;

% Check starting vector
if isempty(u0)
  for i=1:L
    u0{i} = 2*rand(n(cyc{i}(1)),1) - 1;
```

```
  end
end
min_u0_norm = norm(u0{1});
for i=2:L
  if norm(u0{i})<min_u0_norm
    min_u0_norm = norm(u0{i});
  end
end
if min_u0_norm < eps
  error('Zero starting vector');
end

% Execute power method
if (display >= 0)
  %[Code not shown]
end

u_full = cell(d,1);
for i=1:L
  for j=1:length(cyc{i})
    u_full{cyc{i}(j)} = u0{i};
  end
end

flag = -2;
u = u0;
sigma     = double(ttv(A,u_full));
trace(1)  = sigma;
tracex{1} = u0;

for its=1:maxits
  oldu = u;
  for j=1:L
    newu = double(ttv(A,u_full,-cyc{j}(1))) + shift*u{j};
    nnu = norm(newu);
    if nnu < eps,
      flag = -1;
      break;
    end
    newu = newu / nnu;
    u{j} = newu;
    for i=1:length(cyc{j})
      u_full{cyc{j}(i)} = u{j};
    end
  end
  if flag == -1
    break;
  end
  newsigma = double(ttv(A,u_full));
  if norm(abs(newsigma-sigma)) < tol
    flag = 0;
  end
  if (display > 0) && ((flag == 0) || (mod(its,display) == 0))
    %[Code not shown]
  end
  sigma = newsigma;
  trace(its+1)  = sigma;
  tracex{its+1} = u;
  if flag == 0
    break
  end
end
% Check results
%[Code not shown]
```

A.4 TKP and TKSVD

The function `tkron` computes the tensor Kronecker product of two tensors. If $\mathcal{B}$ and $\mathcal{C}$ are order-d tensors then

```
A = tkron(B,C);
```

is the tensor $\mathcal{A} = \mathcal{B} \otimes \mathcal{C}$. Note that this function does not support Kronecker products between tensors of different orders.

```
function A = tkron(B,C)
%TKRON Create the tensor Kronecker product.
%
%  A=tkron(B,C) Forms the tensor Kronecker product of the two order-d
%  tensors B and C.

if isnumeric(B), B = tensor(B); end
if isnumeric(C), C = tensor(C); end
n = size(B); m = size(C); N = n.*m;
d = length(n);
if length(m)~=d
  error('Tensor Kronecker products of different order tensors not supported.')
end
A = bltensor(zeros(N),m);
for i=1:prod(n)
  A(i) = B(i)*C;
end
```

The optimal k-term TKSVD approximation

$$\mathcal{A} \approx \sum_{i=1}^{k} \mathcal{B}_i \otimes \mathcal{C}_i$$

where $\mathcal{B} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ and $\mathcal{C} \in \mathbb{R}^{m_1 \times \cdots \times m_d}$ can be computed with the function `tksvd`. Specifically, if

```
[B,C] = tksvd(A,m,k);
```

then `B` and `C` are cell arrays such that `B{i}`$= \mathcal{B}_i$ and `C{i}`$= \mathcal{C}_i$ for $i = 1, \ldots, k$. The function `tksvd` follows Algorithm 4.1 fairly closely, and for simplicity always uses the mode-1 block unfolding, i.e. $\mathbf{r} = [1]$ and $\mathbf{c} = 2{:}d$.

We need a function that computes the matrix KSVD. Our function `ksvd` uses the PROPACK [47] toolbox for Matlab implementation of a Lanczos SVD algorithm to compute only the first k singular values and singular vectors of a matrix.

```
function [U,sigma,V] = ksvd(A,n1,n2,m1,m2,k)
%KSVD Compute the Krockecker SVD.
%
% [U,sigma,V]=KSVD(A,n1,n2,m1,m2,k) Compute the k-term KSVD approximation.
%  A is an n1-by-n2 block matrix with m1-by-m2 blocks
%  Let p = min(n1*n2,n2*m2)
%  U is an p-by-1 cell array, U{i} is n1-by-n2, and the column
%    vectors vec(U{1}),...,vec(U{p}) are orthonormal.
%  V is an p-by-1 cell array, V{i} is m1-by-m2, and the column
%    vectors vec(V{1}),...,vec(V{p}) are orthonormal.
%  sigma(1) >= ... >= sigma(p) >= 0.
%
%  A = sigma(1)Kron(U{1},V{1}) + ... + sigma(p)Kron(U{p},V{p})

Atilde = [];
for j=1:n2
  cols = (j-1)*m2+1:j*m2;
  for i=1:n1
    rows = (i-1)*m1+1:i*m1;
    Atilde = [Atilde; vec(A(rows,cols))'];
  end
end

if nargin<6
  %%Compute full KSVD.
  [U_tilde,sigma,V_tilde] = svd(A_tilde);
  sigma = diag(sigma);
  p = length(sigma);
  U = cell(p,1);
  V = cell(p,1);
  for k=1:p
    U{k} = reshape(U_tilde(:,k),n1,n2);
    V{k} = reshape(V_tilde(:,k),m1,m2);
  end
else
  %%Compute only first k terms in the KSVD using Lanczos.
  [U_tilde,sigma,V_tilde] = lansvd(A_tilde,k,'L');
  sigma = diag(sigma);
  U = cell(k,1);
  V = cell(k,1);
  for k=1:k
    U{k} = reshape(U_tilde(:,k),n1,n2);
    V{k} = reshape(V_tilde(:,k),m1,m2);
  end
end
```

```
function [B,C] = tksvd(A,m,k)
%TKSVD Compute the tensor KSVD.
%
%  [B,C]=TKSVD(A,m) computes the one term TKSVD approximation of the
%  tensor A where size(C)=m.
%
%  [B,C]=TKSVD(A,m,k) computes the k-term TKSVD approximation.
%
% Stefan Ragnarsson, 2011.

warning off all
N = size(A);
d = length(N);
n = N./m;

if nargin<3, k=1; end

A_blocked = bltensor(A,m);
A_blunf   = bltenmat(A_blocked,1);
p1 = n(1);  p2 = prod(n(2:d));
q1 = m(1);  q2 = prod(m(2:d));
[U,S,V] = ksvd(A_blunf.data,p1,p2,q1,q2,k);

if k>1
  B = cell(k,1); C = cell(k,1);
  for i=1:k
    B{i} = sqrt(S(i))*U{i};
    B{i} = tensor(reshape(B{i},n));
    C{i} = sqrt(S(i))*V{i};
    C{i} = tensor(reshape(C{i},m));
  end
else
  B = sqrt(S)*U{1};
  B = tensor(reshape(B,n));
  C = sqrt(S)*V{1};
  C = tensor(reshape(C,m));
end
```

A.5 HOQRD

The function `hoqrd` computes the HOQRD of a given tensor. If $\mathcal{A} \in \mathbb{R}^{n_1,\ldots,n_d}$ and

```
[R,Q] = hoqrd(A);
```

then $\mathcal{R} \in \mathbb{R}^{n_1 \times \cdots \times n_d}$ is the core tensor and $\texttt{Q\{i\}} = Q_i \in \mathbb{R}^{n_i \times n_i}$ are orthogonal such that

$$\mathcal{A} \;=\; (Q_1, \ldots, Q_d) \cdot \mathcal{R}$$

is the HOQRD of $\mathcal{A}$.

```
function [R,Q] = hoqrd(A)
% HOQRD Compute the Higher-Order QR Decomposition.
%
% [R,Q]=HOQRD(A) returns the full HOQRD of the order-d tensor A. The core
%  tensor R has the same size as A and Q is a cell array of length d, where
%  each Q{i} is an orthogonal matrix and A = ttm(R,Q).
%
% Stefan Ragnarsson, 2011.

n = size(A);
d = length(n);
R = A;
Q = cell(d,1);
for i=1:d
  Ai = tenmat(A,i);
  [Q{i},~,~] = qr(Ai.data);
  temp = tenmat(R,i);
  temp = Q{i}'*temp;
  temp = reshape(temp.data,[n(i),n(1:i-1),n(i+1:d)]);
  R = permute(temp,[2:i,1,i+1:d]);
end
```

BIBLIOGRAPHY

[1] R. Badeau, R. Boyer (2008). *Fast Multilinear Singular Value Decomposition for Structured Tensors*. SIAM. J. Matrix Anal. & Appl. 30, pp. 1008–1021.

[2] G. Ballard, J. Demmel, O. Holtz, and O. Schwartz (2010). *Minimizing Communication in Numerical Linear Algebra*, UCB/EECS-2009-62.

[3] G. Baumgartner, A. Auer, D. Bernholdt, A. Bibireata, V. Choppella, D. Cociorva, X. Gao, R. Harrison, S. Hirata, S. Krishnamoorthy, S. Krishnan, C. Lam, Q. Lu, M. Nooijen, R. Pitzer, J. Ramanujam, P. Sadayappan, and A. Sibiryakov (2005). *Synthesis of High-Performance Parallel Programs for a Class of ab initio Quantum Chemistry Models,* Proceedings of the IEEE, 93, No. 2, pp. 276–292.

[4] J.D. Carroll, J.J. Chang (1970). *Analysis of individual differences in multidimensional scaling via an N-way generalization of 'Eckart-Young' decomposition*, Psychometrica, 35, pp. 283–319.

[5] D. Cartwright, B. Sturmfels (2010). *The number of eigenvalues of a tensor.* arXiv:1004.4953v1.

[6] G.K.-L. Chan, J.J. Dorando, D. Ghosh, J. Hachmann, E. Neuscamman, H. Wang, and T. Yanai (2008). *An introduction to the density matrix renormalization group ansatz in quantum chemistry.* Prog. Theor. Chem. and Phys., 18:49.

[7] K.C. Chang, L. Qi and G. Zhou (2010). *Singular values of a real rectangular tensor*, J. Math. Anal. & Appl. 370, pp. 284–294.

[8] K.C. Chang, T. Zhang (2009). *Multiplicity of singular values for tensors*, Commun. Math. Sci. 7, pp. 611–625.

[9] P. Comon (2001). *Tensor decompositions: State of the art and applications*, in Mathematics and Signal Processing V, J.G. McWhirter and I. K. Proudler, eds., Oxford University Press, pp. 1–24.

[10] P. Comon (2004). *Canonical tensor decompositions*, I3S Report RR-2004-17, Lab. I3S, 2000 route des Lucioles, B.P.121,F-06903 Sophia-Antipolis cedex, France.

[11] P. Comon (2009). *Tensors versus matrices : Usefulness and unexpected properties*, Statistical Signal Processing, SSP '09. IEEE/SP 15th Workshop on Statistical Signal Processing.

[12] P. Comon, G.H. Golub, L.-H. Lim, and B. Mourrain (2008). *Symmetric tensors and symmetric tensor rank*, SIAM J. Matrix Anal. Appl. Vol. 30, No. 3, pp. 1254–1279.

[13] P. Comon, M. Sorensen, and E. Tsigaridas (2009). *Decomposing tensors with structured matrix factors reduces to rank-1 approximations*, Research report I3S/RR-2009-11.

[14] D. Cox, J. Little, and D. O'Shea (2005). *Using Algebraic Geometry*, 2nd ed., Springer-Verlag.

[15] L. De Lathauwer (1997). *Signal Processing based on Multilinear Algebra*, PhD thesis, Faculty of Engineering, K.U.Leuven (Leuven, Belgium), 256 p., Lirias number: 179323.

[16] L. De Lathauwer (2008). *Decompositions of a higher-order tensor in block terms – Part I: Lemmas for partitioned matrices*. SIAM Journal on Matrix Analysis and Applications, 30, pp. 1022–1032.

[17] L. De Lathauwer (2008). *Decompositions of a higher-order tensor in block terms – Part II: Definitions and uniqueness*. SIAM Journal on Matrix Analysis and Applications, 30, pp. 1033–1066.

[18] L. De Lathauwer (2008). *Decompositions of a higher-order tensor in block terms – Part III: Alternating least squares algorithms*. SIAM Journal on Matrix Analysis and Applications, 30, pp. 1067–1083.

[19] L. De Lathauwer, P. Comon, B. De Moor, and J. Vandewalle (1995). *Higher-order power method–Application in independent component analysis*, in proceedings of the International Symposium on Nonlinear Theory and Its Applications (NOLTA '95), Las Vegas, NV, pp. 91–96.

[20] L. De Lathauwer, B. De Moor, and J. Vandewalle (2000). *A Multilinear Singular Value Decomposition*, SIAM J. Matrix Anal. Appl., 21, pp. 1253–1278.

[21] L. De Lathauwer, B. De Moor, and J. Vandewalle (2000). *On the best rank-1 and rank-$(R_1, R_1, \ldots, R_N)$ approximation of higher-order tensors*, SIAM J. Matrix Anal. Appl. Vol. 21, No. 4, pp. 1324–1342.

[22] V. De Silva, L.-H. Lim (2008). *Tensor rank and the ill-posedness of the best low-rank approximation problem*, SIAM Journal on Matrix Analysis and Applications, 30, No. 3, pp. 1084–1127.

[23] P. Drineas, G. Golub, L.-H. Lim, and M. Mahoney (2006). *Bridging the gap between numerical linear algebra, theoretical computer science, and data applications*, SIAM News, 39, No. 8.

[24] E. Elmroth, F. Gustavson, I. Jonsson, and B. Kågström (2005). *Recursive Blocked Algorithms and Hybrid Data Structures for Dense matrix Library Software*, SIAM Review, 46, pp. 3–45.

[25] G.H. Golub, W. Kahan (1965). *Calculating the singular values and pseudo-inverse of a matrix*, J. Soc. Indust. Appl. Math. Ser. B Numer. Anal., 2, pp. 205–224.

[26] G.H. Golub, C.F. Van Loan (1996). *Matrix Computations*, 3rd edition, Johns Hopkins University Press.

[27] L. Grasedyck (2010). *Hierarchical Singular Value Decomposition of Tensors*, SIAM J. Matrix Anal. Appl., Vol. 31, No. 4, pp. 2029–2054.

[28] R.A. Harshman (1970). *Foundation of the PARAFAC procedure: Models and conditions for an "exploratory" multi-modal factor analysis*, UCLA Working Papers in Phonetics, 16, pp. 1–84.

[29] M. Head-Gordon, Y. Shao, C. Saravanan, and C. White (2003). *Curvy Steps for Density Matrix Based Energy Minimization: Tensor Formulation and Toy Applications*, Molecular Physics, 101, pp. 37–43

[30] H.V. Henderson, S.R. Searle (1981). *The Vec-Permutation Matrix, The Vec Operator, and Kronecker Products, A Review*, Linear and Mulitilinear Algebra, 9, pp. 271–288.

[31] C.J. Hillar, L.-H. Lim (2009). *Most tensor problems are NP hard.* arXiv:0911.1393v2 [cs.CC].

[32] S. Hu, Z.-H. Huang, C. Ling, and L. Qi (2011). *E-Determinants of Tensors*, arXiv:1109.0348v3.

[33] J. JáJá (1979). *Optimal evaluation of pairs of bilinear forms*, SIAM J. Comput., 8, pp. 443–462.

[34] J. JáJá, J. Takche (1986). *On the validity of the direct sum conjecture*, SIAM J. Comput., 15, pp. 1004–1020.

[35] H.A.L. Kiers (2000). *Towards a standardized notation and terminology in multiway analysis*, J. Chemometr., 14, pp. 105–122.

[36] E. Kofidis, P.A. Regalia (2000). *The higher-order power method revisited: convergence proofs and effective initialization*. Proceedings of the Acoustics, Speech, and Signal Processing, IEEE International Conference, Vol. 5, pp. 2709–2712.

[37] E. Kofidis, P.A. Regalia (2002). *On the best rank-1 approximation of higher-order supersymmetric tensors*, SIAM J. Matrix Anal. Appl. Vol. 23, pp. 863–884.

[38] T.G. Kolda (2006). *Multilinear operators for higher-order decompositions*. Technical Report Number SAND2006-2081, Sandia National Laboratories, Albuquerque, NM and Livermore, CA.

[39] T.G. Kolda, B.W. Bader (2006). *Algorithm 862: MATLAB Tensor Classes for Fast Algorithm Prototyping*, ACM Transactions on Mathematical Software, Vol. 32, No. 4, pp. 635–653.

[40] T.G. Kolda, B.W. Bader (2006). "MATLAB Tensor Toolbox Version 2.4. `http://csmr.ca.sandia.gov/~tgkolda/TensorToolbox/`, March 2010.

[41] T.G. Kolda, B.W. Bader (2009). *Tensor Decompositions and Applications*, SIAM Review 51, pp. 455–500.

[42] T.G. Kolda, J. R. Mayo (2010). *Shifted power method for computing tensor eigenpairs.* arXiv:1007.1267v1.

[43] P.M. Kroonenberg (2008). *Applied Multiway Data Analysis*, Wiley.

[44] J.B. Kruskal (1977). *Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra Appl., 18, pp. 95–138.

[45] J.B. Kruskal (1989). *Rank, decomposition, and uniqueness for 3-way and N-way arrays*, in Coppi and Bolasco, pp. 7–18.

[46] C. Lanczos (1950). *An iteration method for the solution of the eigenvalue problem of linear differential and integral operators*, J. Res. Nat. Bur. Standards 45, pp. 255–282.

[47] R.M. Larsen (2000). *PROPACK Version 1.1 for Matlab*. `http://soi.stanford.edu/~rmunk/PROPACK/`, March 2004.

[48] L.-H. Lim (2005). *Singular values and eigenvalues of tensors : A variational approach*, Proceedings of the IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing, 1 (2005), pp. 129–132.

[49] M.W. Mahoney, L.-H. Lim, and G. Carlsson (2009). *Algorithmic and Statistical Challenges in Modern Large-Scale Data Analysis are the Focus in MMDS 2008*, SIGKDD Explorations, 10(2), pp. 57–60.

[50] J. Morton, L.-H. Lim (2009). *Principal Cumulant Component Analysis*, Technical report, Stanford University.

[51] V. Olshevsky, I.V. Oseledets, and E.E. Tyrtyshnikov (2006). *Tensor properties of multilevel Toeplitz and related matrices*. Lin. Alg. Appl. 412, pp. 1–21.

[52] I.V. Oseledets (2011). *Tensor-Train Decomposition*, SIAM J. on Scientific Computing, Vol. 33, pp. 2295–2317.

[53] I.V. Oseledets, E.E. Tyrtyshnikov (2009). *Recursive Decomposition of Multidimensional Tensors*, Doklady Mathematics, Vol. 80, No. 1, pp. 460–462.

[54] C.C. Paige (1974). *Bidiagonalization of Matrices and Solution of Linear Equations*, SIAM J. Num. Anal. 11, pp. 197–209.

[55] J. Papy, L. De Lathauwer, and S. Van Huffel (2005). *Exponential data fitting using multilinear algebra. The single-channel and the multichannel case*, Numerical Linear Algebra and Applications, 12, pp. 809–826.

[56] A. H. Phan, A. Cichocki (2009). *Advances in PARAFAC Using Parallel Block Decomposition*. ICONIP 2009, Part I, LNCS 5863, pp. 323–330.

[57] A. H. Phan, A. Cichocki (2009). *Block Decomposition for Very Large-Scale Nonnegative Tensor Factorization*, 2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing.

[58] N. Pitsianis, C.F. Van Loan (1993). *Approximation with Kronecker Products*, Linear Algebra for Large Scale and Real-Time Applications, Kluwer Publications, 1993, pp. 293–314.

[59] C. Procesi (2006). *Lie Groups: An Approach through Invariants and Representations*, Springer-Verlag.

[60] L. Qi (2005). *Eigenvalues of a real supersymmetric tensor*, Journal of Symbolic Computation 40, pp. 1302–1324.

[61] L. Qi (2006). *Rank and eigenvalues of a supersymmetric tensor, the multivariate homogeneous polynomial and the algebraic hypersurface it defines*, Journal of Symbolic Computation 41, pp. 1309–1327.

[62] L. Qi (2007). *Eigenvalues and invariants of tensors*, J. Math. Anal. Appl. 325, pp. 1363–1377.

[63] B. Savas, L.L. Elden (2009). *Krylov subspace methods for tensor computations*. Technical report, Department of Mathematics, Linkoping University. Available at `http://www.mai.liu.se/~laeld/`.

[64] E. Scheinerman (2005). *Matgraph Version 2.4 for Matlab*. `http://www.ams.jhu.edu/~ers/matgraph/`, March 2010.

[65] A. Smilde, R. Bro, and P. Geladi (2004). *Multi-Way Analysis: Applications in the Chemical Sciences*, Wiley, West Sussex, England.

[66] G. W. Stewart (1998). *Two algorithms for the efficient computation of truncated pivoted QR approximations of sparse matrices*, Tech. Report TR-98-12, University of Maryland, Computer Science Department.

[67] V. Strassen (1973). *Vermeidung yon Divisionen,*. J. Reine Angew. Math. 264, pp. 184–202.

[68] D.S. Tracy, R.P. Singh (1972). *A New Matrix Product and Its Applications in Partitioned Matrices*, Statistica Neerlandica 26, pp. 143–157.

[69] L.R. Tucker (1966). *Some mathematical notes on three-mode factor analysis*, Psychometrica, 31, pp. 279–311.

[70] C.F. Van Loan (1992). *Computational Frameworks for the Fast Fourier Transform*, SIAM Publications, Philadelphia, PA.

[71] C.F. Van Loan (2000). *The Ubiquitous Kronecker Product*. Journal of Computational and Applied Mathematics, 123, pp. 85–100.

[72] C.F. Van Loan (2009). *Future Directions in Tensor-Based Computation and Modeling*, NSF Workshop Report.

[73] Y. Wang, L. Qi (2007). *On the successive supersymmetric rank-1 decomposition of higher-order supersymmetric tensors*, Numerical Linear Algebra with Applications, Vol. 14, No. 4, pp. 503–519.

[74] H. J. Wang, F. Xu, and F. Wang (2011). *Tensor Factorization and Clustering for the Feature Extraction Based on Tucker3 with Updating Core*, Advanced Materials Research, 308-310, 2517.

[75] Y. Wang, L. Qi, X. Zhang (2009). *A practical method for computing the largest M-eigenvalue of a fourth-order partially symmetric tensor*, Numer. Linear Algebra Appl. 16, pp. 589–601.

[76] H. Weyl (1997). *The classical groups: Their Invariants and Representations*, Princeton Landmarks in Mathematics, Princeton University Press.

[77] S.R. White, R.L. Martin (1999). *Ab Initio Quantum Chemistry Using the Density Matrix Renormalization Group*, *J. Chem. Phys.*, 110, No. 9, p. 4127.

[78] Q. Wu, T. Xia, C. Chen, H.-Y. Lin, H. Wang, and Yizhou Yu (2008). *Hierarchical Tensor Approximation of Multi-Dimensional Visual Data, IEEE Transactions on Visualization and Computer Graphics*, Vol. 14, No. 1, 2008, pp. 186–199.