Master's Thesis

Analysis of $2 \times 2 \times 2$ Tensors

Ana Rovi

LiTH - MAT - INT - A - - 2010/01 - - SE

Analysis of $2 \times 2 \times 2$ Tensors

MAI Mathematics, Linköpings Universitet

Universidad Nacional de Educación a Distancia. España

Ana Rovi

LiTH - MAT - INT - A - - 2010/01 - - SE

Master's Thesis: **30 ETCS**

Supervisor: **Göran Bergqvist**,
MAI Mathematics, Linköpings Universitet

Examiner: **Göran Bergqvist**,
MAI Mathematics, Linköpings Universitet

Linköping: **June 2010**

LINKÖPINGS UNIVERSITET

Avdelning, Institution
Division, Department

Matematiska Institutionen
581 83 LINKÖPING
SWEDEN

Datum
Date

June 2010

Language

□ Svenska/Swedish
☒ Engelska/English
□ ______________

Rapporttyp
Report category

□ Licentiatavhandling
☒ Examensarbete
□ C-uppsats
□ D-uppsats
□ Övrig rapport
□ ______________

ISBN

ISRN
LiTH - MAT - INT - A - - 2010/01 - - SE

Serietitel och serienummer
Title of series, numbering

ISSN
0348-2960

URL för elektronisk version

http://urn.kb.se/resolve?urn=urn:nbn:se:liu:diva-56762

Titel
Title

Analysis of $\mathbf{2} \times \mathbf{2} \times \mathbf{2}$ Tensors

Författare
Author

Ana Rovi

Sammanfattning
Abstract

The question about how to determine the rank of a tensor has been widely studied in the literature. However the analytical methods to compute the decomposition of tensors have not been so much developed even for low-rank tensors.
In this report we present analytical methods for finding real and complex PARAFAC decompositions of $2 \times 2 \times 2$ tensors before computing the actual rank of the tensor. These methods are also implemented in MATLAB.
We also consider the question of how best lower-rank approximation gives rise to problems of degeneracy, and give some analytical explanations for these issues.

Nyckelord
Keyword

Tensor decomposition. PARAFAC decomposition. Alternating Least Squares. Tensor Rank. Typical and generic rank. Best lower-rank approximation. Tensor Toolbox for MATLAB. Degeneracy. Uniqueness. Tensor classification

To my parents

Abstract

The question about how to determine the rank of a tensor has been widely studied in the literature. However the analytical methods to compute the decomposition of tensors have not been so much developed even for low-rank tensors.

In this report we present analytical methods for finding real and complex PARAFAC decompositions of $2 \times 2 \times 2$ tensors before computing the actual rank of the tensor. These methods are also implemented in MATLAB.

We also consider the question of how best lower-rank approximation gives rise to problems of degeneracy, and give some analytical explanations for these issues.

Keywords: Tensor decomposition. PARAFAC decomposition. Alternating Least Squares. Tensor Rank. Typical and generic rank. Best lower-rank approximation. Tensor Toolbox for MATLAB. Degeneracy. Uniqueness. Tensor classification

Acknowledgments

First of all, I would like to thank my supervisor, Göran Bergqvist for his time, kindness and support. He has changed my way to approach mathematics and to understand what it means to do mathematics. Tack så mycket, Göran!

I would also like to thank Erik Aas for his thoughtful comments and interesting questions about the contents. Muchas gracias también a Carmen Verde que nos ayudó a mi hermana y a mi en la inscripción para el máster. También agradezco a Ruth Lamagrande su ayuda en todo lo relacionado con el intercambio Erasmus. I am very grateful to Milagros Izquierdo and Antonio F. Costa for organizing the Erasmus exchange programme which has made it possible for me to come to Linköpings universitet to work on this thesis and I am also grateful for their support. Finally, I would like to say that nothing would have been possible without my sister Carmen, who is the unconditional support in my life.

Contents

List of Figures

List of MATLAB codes

Introduction

This work has been written as the final thesis of the master's degree "Máster en Matemáticas Avanzadas" of the Universidad Nacional de Educación a Distancia, Spain. This master's thesis has been written at Linköpings Universitet, Sweden due to an Erasmus Exchange organized between both universities and has been supervised by Göran Bergqvist.

This chapter will provide an overview of the whole report. We give some historical background and give an outline of the different chapters, describing what topics are covered.

Historical Background

A fundamental problem in mathematics is given by the question of how to organize data so that they reveal relevant information. If we can identify unknowns and we can rearrange given data somehow so that it all fits into a problem that we know how to solve, then we are half way to the solution. - Although perhaps we still have to work through a long algorithm to find the final solution to our initial problem. It is also interesting that new problems also give rise to new algorithms to solve them. So it is easy to follow that, mathematicians of all times have thought about these questions and have always found interesting ways of approaching the same theoretical problem. The earliest recorded analysis of a study of data stored as simultaneous equations is found in the Chinese book Chiu-chang Suan-shu (Nine Chapters on Arithmetic), written around 200 B.C.

As we said before, new problems give rise to new algorithms to solve them. New problems give rise to new ways of organizing and storing data and new ways of looking for relevant information. It is interesting that these new problems to solve by mathematical means often arise in a non-mathematical field. Let it be chemistry, psychology or biology, results in these fields often relay on numerical data that need to be organized by means of mathematics. So it is interesting to remark that many of the relevant papers that we shall talk about in this report have been published in the Psychometric Society journal, Psychometrika, which contains articles on the development of quantitative models of psychological phenomena, as well as statistical methods and mathematical techniques for evaluating psychological and educational data. It is also interesting to remark that since the first papers about decomposition of multidimensional arrays were published by Hitchcock [18] around 1927, there has been a great development in the subject and the way it has been dealt with. While the first papers had a more applied approach, many of the more recent papers concentrate on more mathematical aspects of the algorithms used when working with tensors.

Purpose of this Thesis

In this thesis I will concentrate on the study of $2 \times 2 \times 2$ tensors, specially on issues concerning rank, decompositions and problems of degeneracy that arise when computing lower rank approximations to a tensor. We try to demonstrate that even such a small tensor as $2 \times 2 \times 2$, which is indeed the smallest tensor possible, has special features which make its analysis very interesting and very different from the study of matrices. We relate these features to the inner structure underlying the tensor and we give a classification of theses $2 \times 2 \times 2$ tensors according to the different features they display. Even to analyze such a small tensor, we have to develop special mathematical tools, based on linear algebra, but completely different from their counterpart in matrix analysis. We also want to point out that many problems concerning rank and decompositions are still open, waiting for the development of mathematical tools that will help solving them.

Programming Enviroments

In this report we use MATLAB to compute results and run M-files that demonstrate algorithms about tensors. We present several examples of MATLAB codes developed to solve relevant problems and questions about tensors.

We use the MATLAB Tensor Toolbox developed by Brett Bader and Tamara Kolda, see [3].

Outline of the Chapters

Chapter 1. Preliminaries

We present the necessary mathematical tools to understand tensors and to develop further work with them. We try to generalize features from one-dimensional and two-dimensional arrays to multidimensional arrays.

Chapter 2. Tensors

This long Chapter is dedicated to the study of the inner structure of tensors. We study tensors of different ranks and we try to find some analytical answers to the question of identifying its rank. We also explain how to work out the PARAFAC decomposition of tensors of different rank.

Another issue we study in this Chapter is that of computing the best lower rank approximation to a tensor. This problem is also very different from its matrix counterpart, since it can only be done using iterative methods instead of using an straightforward algorithm as given by the Eckart-Young theorem. In fact, the best lower rank approximation does not even exist in some cases.

Chapter 3. Uniqueness

In this Chapter we study the sufficient and necessary conditions for the uniqueness of a tensor decomposition. We state Kruskal's Theorem for Uniqueness and we give examples to demonstrate these conditions.

Chapter 4. Degeneracy

In this Chapter we study the special features displayed by certain tensors. We shall give examples demonstrating these features and we will relate them to the inner structure of the tensors where they arise. We will give not only a numerical approach to this issue but also an analytical explanation.

Chapter 5. Classification of $2 \times 2 \times 2$ Tensors

In this Chapter we present the classification of the tensors studied in 8 different classes according to the different features they display.

Chapter 1

Preliminaries

This chapter will give us the necessary tools to describe tensors and to work with them. It will also give an outline of the problems we attempt to discuss in the following chapters of this report.

This preliminary chapter also intends to make clear that although tensors are closely related to matrices, there are many important differences between them which makes matrix analysis and tensor analysis quite different subjects, each with their own open questions and specific applications. We will also give examples of how to compute tensors with MATLAB.

Understanding Arrays

Whenever we encounter data we must think of the best way of arranging them so that we obtain relevant information that will help us solving the given problem. We arrange words in alphabetically ordered lists so that we can find them more easily, we arrange events to organize a schedule and we arrange data in arrays so that relevant information becomes highlighted and we can describe relationships more easily as well as operate with the given data more efficiently.

1.1 Working with Vectors

An array consisting of a single column or row is called a vector. Hence we can define a vector as a 1-dimensional array. Vectors are denoted by boldface lowercase letters, e.g., $\mathbf{a}$. The i^{th} entry of a vector $\mathbf{a}$ is denoted by a_i.

Thus we can write a vector $\mathbf{a} \in \mathbb{R}^2$ as

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$$

Although vectors can be studied from a geometrical point of view, in this report we shall focus on a more arithmetical approach, studying the most relevant operations between vectors as elements of a vector space.

Vector Addition

We can add two or more vectors by adding their corresponding entries.

Let us write two vectors $\mathbf{a}, \mathbf{b} \in \mathbb{R}^2$ as

$$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

Then we can write their sum as

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \end{pmatrix}$$

Vector Products

While we can define vector addition only in one way, things change when defining vector products and we find different ways of multiplying the entries of the vectors.

- *Inner Product:*

  Let

  $$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

  Then we can write their inner product as

  $$\left\langle \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}, \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right\rangle = a_1 b_1 + a_2 b_2$$

  We can see that this product gives a scalar as a result.

- *Outer Product:*

  Let

  $$\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \text{ and } \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$$

  Then we can write the outer product of the two vectors $\mathbf{a}$, $\mathbf{b} \in \mathbb{R}^2$ as

  $$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \circ \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 b_1 & a_1 b_2 \\ a_2 b_1 & a_2 b_2 \end{pmatrix} \tag{1.1}$$

  We see that this product gives a matrix as a result.

Norm and Normalization

When considering a vector as a geometrical object, one of its most important features is length. If we take different vectors with the same direction, we can see that they are scalar multiples of each other. Hence we can choose one single vector to define a direction. We will take this vector to have length 1 unit, and we will define a vector in a given direction to be normalized if it has unit length. We define the length of a vector to be its norm. Norm and normalization will be important when computing with MATLAB.

- *Euclidean Vector Norm*

  Although different norms can be defined on vectors, we will consider here the Euclidean norm, which is closely related to the geometric length of the vector.

  For a vector $\mathbf{v} \in \mathbb{R}^n$, the *euclidean norm* of $\mathbf{v}$ is defined as,

$$\|\mathbf{v}\| = \left(\sum_{i=1}^{n} v_i^2 \right)^{1/2} = \sqrt{\langle \mathbf{v}, \mathbf{v} \rangle} \tag{1.2}$$

1.2 Working with Matrices

We define a matrix as as two-dimensional array of m rows and n columns. Matrices are denoted by boldface capital letters, e.g., $\mathbf{A}$.

The i^{th} row is denoted by $\mathbf{A}_{i*}$ and the j^{th} column is denoted by $\mathbf{A}_{*j}$

Thus we can write a $m \times n$ matrix as

$$\mathbf{A}_{m\times n} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix}$$

The first subscript on an individual entry in a matrix designates the row that the entry occupies, and the second subscript denotes the column that the entry occupies.

Adding Matrices

Proceeding in the same way we saw for vectors, we can add matrices by adding the corresponding entries of each matrix. It is easy to see that there is only one to define matrix addition.

Matrix Products

Just as we saw when considering vector multiplication, we can define different ways of multiplying matrices. In this report we will use the following four products.

- *Usual Matrix Multiplication*

  Let

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1K} \\ a_{21} & a_{22} & \cdots & a_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1} & a_{I2} & \cdots & a_{IK} \end{pmatrix} \text{and } \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1J} \\ b_{21} & b_{22} & \cdots & b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ b_{K1} & b_{K2} & \cdots & b_{KJ} \end{pmatrix}$$

  be an $I \times K$ matrix and a $K \times J$ matrix respectively.

Then we can define the matrix product $\mathbf{AB}$ to be an $I \times J$ matrix where each entry $(\mathbf{AB})_{ij}$ will be given by the result of the scalar product of the i^{th} row of $\mathbf{A}$ and the j^{th} column of $\mathbf{B}$, so that

$$\left(\begin{array}{cccc} a_{11} & a_{12} & \cdots & a_{1K} \\ \vdots & \vdots & \ddots & \vdots \\ \hline \multicolumn{1}{|c}{a_{i1}} & a_{i2} & \cdots & \multicolumn{1}{c|}{a_{iK}} \\ \hline \vdots & \vdots & \ddots & \vdots \\ a_{I1} & a_{I2} & \cdots & a_{IK} \end{array}\right) \left(\begin{array}{ccc|c|cc} b_{11} & b_{12} & \cdots & b_{1j} & \cdots & b_{1J} \\ b_{21} & b_{22} & \cdots & b_{2j} & \cdots & b_{2J} \\ \vdots & \vdots & \ddots & \vdots & \cdots & \vdots \\ b_{K1} & b_{K2} & \cdots & b_{Kj} & \cdots & b_{KJ} \end{array}\right) =$$

$$\begin{pmatrix} \cdots & \cdots & \cdots & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ \cdots & \langle \mathbf{A}_{i*}, \mathbf{B}_{*j} \rangle & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix}$$

- *Hadamard Product:* $*$

  This matrix product, first defined by the French mathematician Hadamard, is the elementwise matrix product.

  Let

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1J} \\ a_{21} & a_{22} & \cdots & a_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1} & a_{I2} & \cdots & a_{IJ} \end{pmatrix} \text{ and } \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1J} \\ b_{21} & b_{22} & \cdots & b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ b_{I1} & b_{I2} & \cdots & b_{IJ} \end{pmatrix}$$

  be two $I \times J$ matrices.

  Then we can define the Hadamard product $\mathbf{A} * \mathbf{B}$ as

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1J} \\ a_{21} & a_{22} & \cdots & a_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1} & a_{I2} & \cdots & a_{IJ} \end{pmatrix} * \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1J} \\ b_{21} & b_{22} & \cdots & b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ b_{I1} & b_{I2} & \cdots & b_{IJ} \end{pmatrix} =$$

$$\begin{pmatrix} a_{11}b_{11} & a_{12}b_{12} & \cdots & a_{1J}b_{1J} \\ a_{21}b_{21} & a_{22}b_{22} & \cdots & a_{2J}b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1}b_{I1} & a_{I2}b_{I2} & \cdots & a_{IJ}b_{IJ} \end{pmatrix}$$

  It is interesting to remark that the Hadamard product multiplies matrices of the same size and that the resulting matrix has the same size as the original matrices.

- *Kronecker Product:* $\otimes$

  The Kronecker product multiplies any two matrices of any given sizes. Let

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1J} \\ a_{21} & a_{22} & \cdots & a_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1} & a_{I2} & \cdots & a_{IJ} \end{pmatrix} \text{and } \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1L} \\ b_{21} & b_{22} & \cdots & b_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ b_{K1} & b_{K2} & \cdots & b_{KL} \end{pmatrix}$$

  be an $I \times J$ matrix and a $K \times L$ matrix respectively.

  Then the Kronecker product $\mathbf{A} \otimes \mathbf{B}$ is defined as follows,

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1J} \\ a_{21} & a_{22} & \cdots & a_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1} & a_{I2} & \cdots & a_{IJ} \end{pmatrix} \otimes \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1L} \\ b_{21} & b_{22} & \cdots & b_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ b_{K1} & b_{K2} & \cdots & b_{KL} \end{pmatrix} =$$

$$\begin{pmatrix} a_{11}\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1L} \\ b_{21} & b_{22} & \cdots & b_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ b_{K1} & b_{K2} & \cdots & b_{KL} \end{pmatrix} & \cdots & a_{1J}\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1L} \\ b_{21} & b_{22} & \cdots & b_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ b_{K1} & b_{K2} & \cdots & b_{KL} \end{pmatrix} \\ \vdots & \ddots & \vdots \\ a_{I1}\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1L} \\ b_{21} & b_{22} & \cdots & b_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ b_{K1} & b_{K2} & \cdots & b_{KL} \end{pmatrix} & \cdots & a_{IJ}\begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1L} \\ b_{21} & b_{22} & \cdots & b_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ b_{K1} & b_{K2} & \cdots & b_{KL} \end{pmatrix} \end{pmatrix}$$

  The output product is a matrix of size $(IK) \times (JL)$.

- *Khatri-Rao Product:* $\odot$

  The Khatri-Rao product multiplies matrices with the same number of columns. Hence, we deduce that this product computes the Kronecker product of the corresponding columns of each matrix of the two matrices. Let

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1K} \\ a_{21} & a_{22} & \cdots & a_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1} & a_{I2} & \cdots & a_{IK} \end{pmatrix} \text{and } \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1K} \\ b_{21} & b_{22} & \cdots & b_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ b_{J1} & b_{J2} & \cdots & b_{JK} \end{pmatrix}$$

  be an $I \times K$ matrix and a $J \times K$ matrix respectively.

Then the Khatri-Rao product $\mathbf{A} \odot \mathbf{B}$ is defined as follows,

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1K} \\ a_{21} & a_{22} & \cdots & a_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1} & a_{I2} & \cdots & a_{IK} \end{pmatrix} \odot \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1K} \\ b_{21} & b_{22} & \cdots & b_{2K} \\ \vdots & \vdots & \ddots & \vdots \\ b_{J1} & b_{J2} & \cdots & b_{JK} \end{pmatrix} =$$

$$\left(\left(\begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{I1} \end{pmatrix} \otimes \begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{J1} \end{pmatrix}\right) \cdots \left(\begin{pmatrix} a_{1K} \\ a_{2K} \\ \vdots \\ a_{IK} \end{pmatrix} \otimes \begin{pmatrix} b_{1K} \\ b_{2K} \\ \vdots \\ b_{JK} \end{pmatrix}\right)\right) =$$

$$\begin{pmatrix} a_{11}\begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{J1} \end{pmatrix} & \cdots & a_{1K}\begin{pmatrix} b_{1K} \\ b_{2K} \\ \vdots \\ b_{JK} \end{pmatrix} \\ a_{21}\begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{J1} \end{pmatrix} & \cdots & a_{2K}\begin{pmatrix} b_{1K} \\ b_{2K} \\ \vdots \\ b_{JK} \end{pmatrix} \\ \vdots & \ddots & \vdots \\ a_{I1}\begin{pmatrix} b_{11} \\ b_{21} \\ \vdots \\ b_{J1} \end{pmatrix} & \cdots & a_{IK}\begin{pmatrix} b_{1K} \\ b_{2K} \\ \vdots \\ b_{JK} \end{pmatrix} \end{pmatrix}$$

As we can see, the Khatri-Rao product produces an output matrix of size $(IJ) \times K$.

Note that the Khatri-Rao product and the Kronecker product are identical when considering vectors, i.e., $\mathbf{a} \odot \mathbf{b} = \mathbf{a} \otimes \mathbf{b}$.

- *Matrix Scalar Product*

  Let

$$\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1J} \\ a_{21} & a_{22} & \cdots & a_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ a_{I1} & a_{I2} & \cdots & a_{IJ} \end{pmatrix} \text{ and } \mathbf{B} = \begin{pmatrix} b_{11} & b_{12} & \cdots & b_{1J} \\ b_{21} & b_{22} & \cdots & b_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ b_{I1} & b_{I2} & \cdots & b_{IJ} \end{pmatrix}$$

  be two $I \times J$ matrices.

  Then we can define the scalar product $\langle \mathbf{A}, \mathbf{B} \rangle$ as

$$\langle \mathbf{A}, \mathbf{B} \rangle = \sum_{i=1}^{I} \sum_{j=1}^{J} (a_{ij} b_{ij}) = \text{Tr } (\mathbf{A}^T \mathbf{B})$$

It is interesting to remark how multiplication can create new mathematical objects from already existing ones. We can create a two-dimensional array by multiplying two one-dimensional arrays. We can create larger matrices by computing the Kronecker product or the Hadamard product of two matrices and we can also have a scalar as a result when multiplying vectors or matrices.

This idea will also apply to tensors and we will see how a tensor can be created by defining the multiplication of vectors in a multidimensional space.

Matrix Norm

In equation 1.2 we have defined the euclidean norm of a vector to be its length. Although we cannot define matrix norms in the same way, we can relate the entries of the matrix to some scalar that will provide information about the structure of the matrix. In this sense, although we can define different matrix norms, we will concentrate on the Frobenius matrix norm that is defined by the square root of the sum of the squared entries of the matrix.

For a matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$ we define its Frobenius norm as

$$\|\mathbf{A}\|_F = \left(\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2 \right)^{1/2} = \langle \mathbf{A}, \mathbf{A} \rangle \tag{1.3}$$

We will see that the matrix norm of the difference between two matrices $\mathbf{A}_{m\times n}, \mathbf{B}_{m\times n}$, given by $\|\mathbf{A} - \mathbf{B}\|_F$ will define the distance between the two matrices.

Matrix Inverses

When dealing with matrices we are often confronted with the problem of solving equations of the form,

$$\mathbf{Ax} = \mathbf{b} \tag{1.4}$$

This equation can be solved by multiplying both sides of the equation 1.4 by the inverse of the matrix $\mathbf{A}$ given by $\mathbf{A}^{-1}$ such that, $\mathbf{A^{-1}Ax} = \mathbf{A^{-1}b}$ and we can rewrite equation 1.4 as,

$$\mathbf{x} = \mathbf{A^{-1}b}$$

Unfortunately, this straightforward method can only be used when the matrix $\mathbf{A}$ is a square matrix and is non-singular, that is $\|\mathbf{A}\| \neq 0$.

In the cases where the conditions for finding an inverse of the matrix $\mathbf{A}$ do not hold, we must find another way of solving equations as given by 1.4.

We shall use the **pseudoinverse matrix** of $\mathbf{A}$

$$\mathbf{A}^{\dagger} = \begin{cases} (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T \text{ when } rank(\mathbf{A}_{m\times n}) = n \\ \mathbf{A}^T(\mathbf{A}\mathbf{A}^T)^{-1} \text{ when } rank(\mathbf{A}_{m\times n}) = m \end{cases}$$

We find that the pseudoinverse is a generalization of the idea of finding the inverse of a matrix.

We have that

- If the system given by equation 1.4 is consistent, then $\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{B}$ will be the solution of minimal euclidean norm.
- If the system given by equation 1.4 is inconsistent, the $\mathbf{x} = \mathbf{A}^{\dagger}\mathbf{B}$ will be the least squares solution of minimal euclidean norm.

However, the pseudoinverse is not a continuous function of the entries of the matrix considered which can lead to numerical errors when using it in computations.

1.3 Working with Tensors

Although we can consider tensors in a broad way as a multilinear map, in the way it is considered in physics and differential geometry, in this report we will be interested in the concept of tensor a a multilinear array that arises as a result of the growth of matrix theory and its applications to new fields. When the problems to solve are no longer supported by matrix theory because we have to deal with more variables, we have to think of organizing data as multi-dimensional arrays. This idea has proven to be an interesting approach to the solutions of many problems in chemometrics, psychometrics, statistics, data mining and other fields where large quantities of data and many variables play a role.

We define this multi-dimensional array as a tensor. Although we can define a tensor to be in N-dimensions, in this report we shall only consider 3-dimensional tensors which display the same properties as the higher-dimensional ones.

It is important to remark that although the applications of tensors to the fields of psychometrics, chemometrics or signal processing are only a few decades old, the mathematical ideas underlying this multidimensional structure were already known in the 19th century and were developed by Cayley [5] who defined hypermatrices and hyperdeterminants and to Schläfli who developed ideas about the N-dimensional space. Many of the ideas from Cayley have been revived by Gelfand, Kapranov and Zelevinsky [15].

Nowadays, there are many important open questions in this field that are being intensively researched because of their mathematical interest and also because of the very important applications that rely on the answers to these questions. Questions such as determining the rank of higher dimensional tensors or computing exact decompositions of these tensors are still open.

In this section we will give some preliminary mathematical tools to work with tensors. In this sense we will first define what a tensor is and how we can define multiplication so that we can work with arrays of different size and in different dimensions.

1.3.1 Defining Tensors

Tensors are denoted by calligraphic script letters, e.g., $\mathcal{T}$. We can visualize 3-dimensional tensors as a parallelepiped.

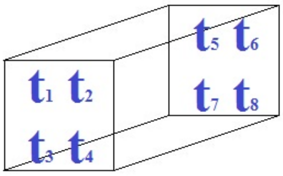

Figure 1.1: Visualization of a $2 \times 2 \times 2$ Tensor

Throughout this report we will use the representation used in Kruskal [21],

$$\mathcal{T} = \begin{array}{cccc} & & t_5 & t_6 \\ & & t_7 & t_8 \\ t_1 & t_2 & & \\ t_3 & t_4 & & \end{array}$$

While we can define a matrix by its number of rows and columns, we need three integers to define a 3-dimensional tensor. In a similar way as is done when working with matrices, we can fix 1, 2 or 3 indexes of each entry to define elements of the tensor.

- *Fibers*

  When working with tensors, columns and rows are replaced by their higher-order analogue, fibers. Hence we can define fibers in the different dimensions or *modes* of the tensor. Furthermore, we deduce that we can identify fibers by fixing two of the three indexes that define an entry of a tensor.

  - Tensor columns are mode-1 fibers, $\mathbf{t}_{:jk}$
  - Tensor rows are mode-2 fibers, $\mathbf{t}_{i:k}$
  - And we can still define a mode-3 fiber $\mathbf{t}_{ij:}$ in the remaining dimension.

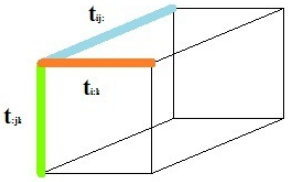

Figure 1.2: Fibers

- *Slices*

  Fixing only one of the three indexes that define tensor entries we define slices.

  - When fixing the first index of the entries of a tensor, we define the **horizontal slices** of the tensor, $\mathbf{T}_{i::}$

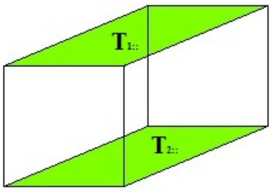

Figure 1.3: Mode-1 Slices.

  - Similarly, when fixing the second index of each entry, we define the **lateral slices** of the tensor, $\mathbf{T}_{:j:}$

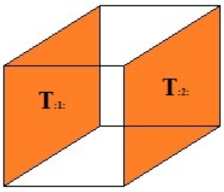

Figure 1.4: Mode-2 Slices

  - And by fixing the third index of each entry, we define the **frontal slices** of the tensor $\mathbf{T}_{::k}$ in the remaining dimension.

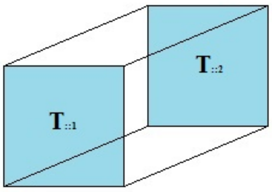

Figure 1.5: Mode-3 Slices

- *Tensor Entries*

  Hence we can write a 2 × 2 × 2 tensor as

$$\mathcal{T} = \begin{array}{cccc} & & t_{112} & t_{122} \\ & & t_{212} & t_{222} \\ t_{111} & t_{121} & & \\ t_{211} & t_{221} & & \end{array} \tag{1.5}$$

  The first subscript on an individual entry in a tensor designates the horizontal slice that the entry occupies. The second subscript denotes the lateral slice occupied by the entry. The third entry shows the frontal slice where the entry lies.

1.3.2 Matricization and Modes

Matricization is the process of rearranging the entries of a tensor so that it can be represented as a matrix. Also called *unfolding* or *flattening*, matricization will be an important tool when working with tensors.

We explained before, that the entries of a 3-dimensional tensor can be arranged in *fibers*. Fibers represent the entries of the tensor, when considered from each of its three different dimensions. Hence, for a 3-dimensional tensor, we will find mode-1, mode-2 and mode-3 fibers.

Building on this idea, we will define the mode-n matricization of a tensor $\mathcal{T}$ as the rearrangement of the entries of the tensor so that the mode-n fibers become the columns of the resulting matrix. This resulting matrix will be denoted by $\mathcal{T}_{(n)}$.

Thus we deduce that the matricization along the different modes of the tensor represented above in 1.5 will be given by the following expressions,

$$\mathcal{T}_{(1)} = \begin{bmatrix} t_{111} & t_{121} & t_{112} & t_{122} \\ t_{211} & t_{221} & t_{212} & t_{222} \end{bmatrix}$$

Mode-1 Matricization
We can see that the entries of the first row represent the entries of the upper slice of the tensor.

$$\mathcal{T}_{(2)} = \begin{bmatrix} t_{111} & t_{211} & t_{112} & t_{212} \\ t_{121} & t_{221} & t_{122} & t_{222} \end{bmatrix}$$

Mode-2 Matricization
We can see that the entries of the first row represent the entries of the left lateral slice, whereas the entries in the lower row represent the entries of the right lateral slice of the tensor.

$$\mathcal{T}_{(3)} = \begin{bmatrix} t_{111} & t_{211} & t_{121} & t_{221} \\ t_{112} & t_{212} & t_{122} & t_{222} \end{bmatrix}$$

Mode-3 Matricization
Finally, we can see that the entries in the upper row represent the entries of the front slice of the tensor.

Example 1.1

Let us take the tensor given by

$$\mathcal{T} = \begin{array}{cccc} & & 5 & 6 \\ & & 7 & 8 \\ 1 & 2 & & \\ 3 & 4 & & \end{array}$$

Then we can represent the matricizations along the three different modes as follows,

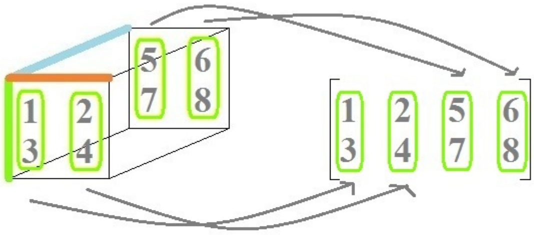

Figure 1.6: Mode-1 Matricization

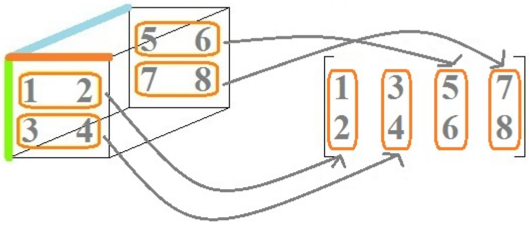

Figure 1.7: Mode-2 Matricization

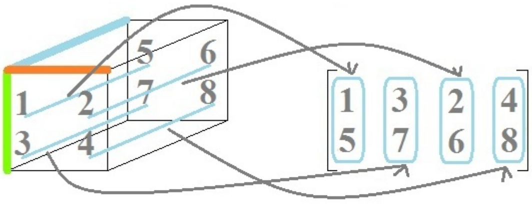

Figure 1.8: Mode-3 Matricization

□

1.4 Redefining Multiplication

1.4.1 Outer Product Revisited

In section 1.1, when studying vector products, we saw that the outer product of two vectors produces a matrix. (See equation 1.1). Taking this idea a bit further we can deduce that the outer product of three vectors produces a 3-dimensional tensor.

Taking each vector to be in a different mode, we can visualize the outer product of three vectors as follows,

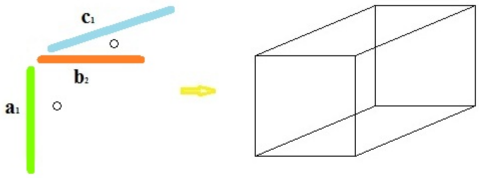

Figure 1.9: Tensor as the Outer Product of Three Vectors

Mathematically, we can write the outer product of three vectors $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^2$ as follows,

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \circ \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \circ \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{array}{cccc} & & a_1b_1c_2 & a_1b_2c_2 \\ & & a_2b_1c_2 & a_2b_2c_2 \\ a_1b_1c_1 & a_1b_2c_1 & & \\ a_2b_1c_1 & a_2b_2c_1 & & \end{array}$$

We can see that the indexes of the entries in the resulting tensor follow the same pattern as displayed by the entries of the tensor given in equation 1.5.

We can rewrite the outer product of three vectors as a matricization of the resulting tensor along the different modes in the following way,

$$\mathcal{T}_{(1)} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \left(\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \odot \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right)^T \tag{1.6}$$

$$\mathcal{T}_{(2)} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \left(\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \odot \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \right)^T \tag{1.7}$$

$$\mathcal{T}_{(3)} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \left(\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \odot \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \right)^T \tag{1.8}$$

MATLAB 1.1

The following MATLAB script performs the outer product of three given vectors and computes the matricization along the three modes of the resulting tensor.

MATLAB 1.1 Outer Product

```
a=[1;2],b=[3;4],c=[5;6]
T_1 = a*(kron(c,b))'; T = tensor(T_1, [2,2,2])
T_1 = a*(kron(c,b))'
T_2 = b*(kron(c,a))'
T_3 = c*(kron(b,a))'
```

Running through this MATLAB code we obtain the following result,

```
a =

     1
     2


b =

     3
     4


c =

     5
     6

T is a tensor of size 2 x 2 x 2
T(:,:,1) =
    15    20
    30    40
T(:,:,2) =
    18    24
    36    48

T_1 =

    15    20    18    24
    30    40    36    48


T_2 =

    15    30    18    36
    20    40    24    48


T_3 =

    15    30    20    40
    18    36    24    48

>>
```

Note that we use the `kron` command of MATLAB to compute the Khatri Rao products given in equations 1.6, 1.7 and 1.8. As we saw in section 1.2 the Kronecker and Khatri Rao products are identical when considering vectors.

1.4.2 Tensor Multiplication

Just as we have defined the multiplication of vectors and different ways of multiplying matrices, we can define tensor multiplication. We can define three different tensor products depending on whether the tensors is multiplied by another tensor of the same size, by a matrix or by a vector.

Thus we define,

- Scalar product of two tensors of the same size.
- The n-mode matrix product of a tensor with a matrix.
- The n-mode vector product of a tensor with a vector.

Tensor Inner Product

We have defined the inner product of two vectors as the sum of the products of the corresponding entries of each vector. In the same way, we can define the inner product of two same-sized tensors.

Let,

$$\mathcal{A} = \begin{array}{cccc} & & a_5 & a_6 \\ & & a_7 & a_8 \\ a_1 & a_2 & & \\ a_3 & a_4 & & \end{array} \quad \text{and} \quad \mathcal{B} = \begin{array}{cccc} & & b_5 & b_6 \\ & & b_7 & b_8 \\ b_1 & b_2 & & \\ b_3 & a_b & & \end{array}$$

Then, the inner product of both tensors is defined by,

$$\langle \mathcal{A}, \mathcal{B} \rangle = a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4 + a_5b_5 + a_6b_6 + a_7b_7 + a_8b_8$$

Tensor Times Matrix

We defined usual matrix multiplication as being the inner product of the rows of the first matrix with the columns of the second matrix.

When considering the product of a tensor times a matrix, we have to decide which dimension of the tensor we are going to take into account when computing the product to develop a similar kind of algorithm as we use for multiplying matrices. We ”decide” the dimension of the tensor we are going to consider by defining the n-mode product of a tensor $\mathcal{T}$ with a matrix $\mathbf{A}$. Hence we can take the product of a given tensor times a matrix in so many modes as dimensions of the tensor.

We denote the n-mode product of a tensor $\mathcal{T}$ with a matrix $\mathbf{A}$ as,

$$\mathcal{P} = \mathcal{T} \times_n \mathbf{A}$$

where each mode-n fiber of $\mathcal{T}$ is multiplied by the matrix $\mathbf{A}$ to compute each mode-n fiber of the resulting tensor $\mathcal{P}$.

We can also express this multiplication in terms of unfolded tensors as

$$\mathcal{P} = \mathcal{T} \times_n \mathbf{A} \Longleftrightarrow \mathcal{P}_{(n)} = \mathbf{A}\mathcal{T}_{(n)}$$

We can visualize this tensor multiplication in the following figure.

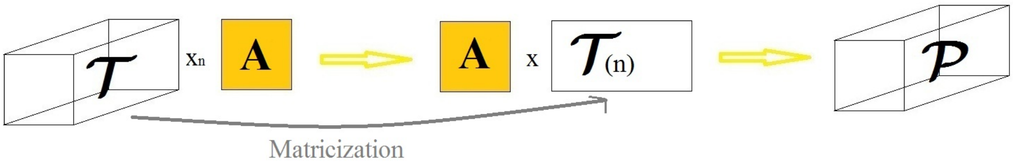

Figure 1.10: Visualization of the Tensor by Matrix Multiplication

We can see that the matricization process is vital when computing with both matrices and tensors. We will see that this is also the case when computing with vectors and tensors. In the following example we are going to compute the product of a tensor by a matrix along the first mode.

Example 1.2

Let us consider the tensor $\mathcal{T} = \begin{array}{cccc} & & 5 & 6 \\ & & 7 & 8 \\ 1 & 2 & & \\ 3 & 4 & & \end{array}$ and the matrix $\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$.

We want to find the 1-mode product of $\mathcal{T}$ with the matrix $\mathbf{A}$.

Writing the tensor $\mathcal{T}$ in its mode-1 matricization as $\mathcal{T}_{(1)} = \begin{pmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{pmatrix}$ we can work out the tensor product as follows,

$$\mathcal{P} = \mathcal{T} \times_1 \mathbf{A} \rightsquigarrow \mathcal{P}_{(1)} = \mathbf{A}\mathcal{T}_{(1)} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{pmatrix}$$

$$= \begin{pmatrix} a+3b & 2a+4b & 5a+7b & 6a+8b \\ c+3d & 2c+4d & 5c+7d & 6c+8d \end{pmatrix}$$

$$\rightsquigarrow \begin{array}{cccc} & & 5c+7d & 6c+8d \\ & & 5a+7b & 6a+8b \\ a+3b & 2a+4b & & \\ c+3d & 2c+4d & & \end{array}$$

□

MATLAB 1.2

The following MATLAB script developed by Bader and Kolda [1] performs the product of a tensor times a matrix along the different modes of the tensor.

MATLAB 1.2 Tensor times Matrix along the Different Modes

```
M = [0,2,5,6;3,4,7,8]; T = tensor(M, [2,2,2])
A = [1,2;3,4]
P1 = ttm(T,A,1)    %mode 1
P2 = ttm(T,A,2)    %mode 2
P3 = ttm(T,A,3)    %mode 3
```

running through this script we obtain,

```
T is a tensor of size 2 x 2 x 2

T(:,:,1) =
     0     2
     3     4
T(:,:,2) =
     5     6
     7     8

A =

     1     2
     3     4

P1 is a tensor of size 2 x 2 x 2
P1(:,:,1) =
     6    10
    12    22
P1(:,:,2) =
    19    22
    43    50
P2 is a tensor of size 2 x 2 x 2
P2(:,:,1) =
     4     8
    11    25
P2(:,:,2) =
    17    39
    23    53
P3 is a tensor of size 2 x 2 x 2
P3(:,:,1) =
    10    14
    17    20
P3(:,:,2) =
    20    30
    37    44
>>
```

Tensor Times Vector

Following the algorithm developed for multiplying tensors with matrices, we can take the product of a given tensor times a vector in so many modes as dimensions of the tensor.

We denote the n-mode product of a tensor $\mathcal{T}$ with a vector $\mathbf{v}$ as,

$$\mathcal{P} = \mathcal{T} \bar{\times}_n \mathbf{v}$$

where each mode-n fiber of $\mathcal{T}$ is multiplied by the vector $\mathbf{v}$ to compute the result.

Example 1.3

Let us consider the tensor $\mathcal{T} = \begin{array}{cccc} & & 5 & 6 \\ & & 7 & 8 \\ 1 & 2 & & \\ 3 & 4 & & \end{array}$ and the vector $\mathbf{v} = \begin{pmatrix} a \\ b \end{pmatrix}$.

We want to find the 1-mode product of $\mathcal{T}$ with the vector $\mathbf{v}$.

We saw in example 1.1 that the mode-1 fibers of $\mathcal{T}$ are given by the columns of its mode-1 matricization, that is, $\mathcal{T}_{(1)} = \begin{pmatrix} 1 & 2 & 5 & 6 \\ 3 & 4 & 7 & 8 \end{pmatrix}$.

Thus, we can work out the tensor product as follows,

$$\mathcal{P} = \mathcal{T} \bar{\times}_1 \mathbf{v} = \begin{pmatrix} a + 3b & 5a + 7b \\ 2a + 4b & 6a + 8b \end{pmatrix}$$

□

MATLAB 1.3

The following MATLAB script developed by Bader and Kolda [1] performs the product of a tensor times a vector along the different modes of the tensor.

MATLAB 1.3 Tensor times Vector

```
M = [1,2,5,6;3,4,7,8]; T = tensor(M, [2,2,2])
v = [1;2]
P1 = ttv(T,v,1)
P2 = ttv(T,v,2)
P3 = ttv(T,v,3)
```

Running through this script we obtain,

```
T is a tensor of size 2 x 2 x 2

T(:,:,1) =
     1     2
     3     4
T(:,:,2) =
     5     6
     7     8

v =

     1
     2

P1 is a tensor of size 2 x 2
P1(:,:) =
     7    19
    10    22


P2 is a tensor of size 2 x 2
P2(:,:) =
     5    17
    11    23
P3 is a tensor of size 2 x 2
P3(:,:) =
    11    14
    17    20
>>
```

Tensor Norm

Whereas the norm of a vector is mainly a geometrical concept that defines its length, we can also define the norm of a tensor in a similar way as it is defined for matrices.

In equations 1.2 and 1.3 we defined the euclidean norm of the vector and the Frobenius matrix norm respectively. Similarly, we can define the Frobenius norm of a tensor $\mathcal{T}$ of size $I \times J \times K$ by the equation

$$\|\mathcal{T}\|_F = \left(\sum_{i=1}^{I} \sum_{j=1}^{J} \sum_{k=1}^{K} |t_{ijk}|^{2}) \right)^{1/2} = \sqrt{\langle \mathcal{T}, \mathcal{T} \rangle} \tag{1.9}$$

where $\langle \mathcal{T}, \mathcal{T} \rangle$ is the inner product of the tensor by itself.

Example 1.4

Let us consider the tensor

$$\mathcal{T} = \begin{array}{cccc} & & 5 & 6 \\ & & 7 & 8 \\ 1 & 2 & & \\ 3 & 4 & & \end{array}$$

then the norm of the tensor will be given by

$$\|\mathcal{T}\|_F = \sqrt{\langle \mathcal{T}, \mathcal{T} \rangle} = \left(1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 + 7^2 + 8^2\right)^{1/2}$$

$$= \left(1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 + 7^2 + 8^2\right)^{1/2} = 2\sqrt{51}$$

□

1.5 Tensor Decompositions

We know that we can decompose a given matrix in different ways depending on the type of problem that we wish to solve.

If we want to solve a system of equations, we will probably choose a LU decomposition for the matrix representing the equations.

Other matrix decompositions take orthogonality as a main issue and compute decompositions with orthogonal components.

But if we want to compute operations on a matrix or discover the inner geometrical structure of the transformation given by a matrix, we will probably prefer to work with a diagonalized version of our original matrix and we will use the SVD decomposition.

Thus, we can deduce that tensor decomposition will be an important issue when analyzing tensors.

We can define different decompositions that represent different approaches to the various problems that arise when studying tensors.

In this report we will consider the generalization of the Singular Value matrix Decomposition (SVD) to higher order arrays which correspond to the Higher Order Singular Value Decomposition (HOSVD) on one hand, and the CANDECOMP/PARAFAC decomposition (canonical decomposition and parallel factor decomposition respectively) on the other hand. These two decompositions are connected with two different tensor generalizations of the concept of matrix rank.

1.5.1 CANDECOMP/PARAFAC

This tensor decomposition was first attempted by Hitchcock [18, 19] in 1927 and Eckart and Young [12] in 1936.

However it was not fully introduced until 1970 with the work of Harshman about the PARAFAC decomposition [16] and of Carroll and Chang about CANDECOMP [4]. Both papers appeared in Psychometrika and explained the same decomposition.

The CANDECOMP/PARAFAC is based on the fact that tensors can be rewritten as the sum of several other tensors.

We saw before in subsection 1.4.1 that the outer product of three vectors gives a tensor as a result. We shall denote this tensor to be of rank 1 and we will use the term "rank 1 tensor" to denote tensors that can be written as the outer product of a vector triple.

The CANDECOMP/PARAFAC decomposition rewrites a given tensor as a sum of several rank 1 tensors.

Following the argument above, we will define a tensor to be of rank 2 if it can be expressed as the sum of two rank 1 tensors. Similarly, we define a tensor to be rank 3 if it can be expressed as the sum of three rank 1 tensors.

Definition The rank of a tensor $\mathcal{T}$ is the minimal number of rank 1 tensors that yield $\mathcal{T}$ as a linear combination [21].

Since in this report we concentrate on $2\times2\times2$ tensors, we will only encounter tensors up to rank 3.

Summarizing we have,

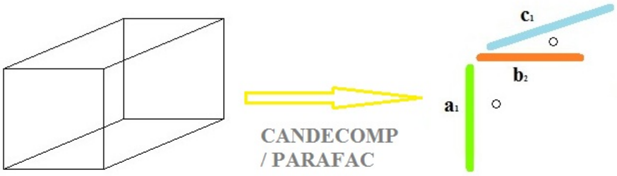

Figure 1.11: Visualization of the PARAFAC Decomposition of a Rank 1 Tensor

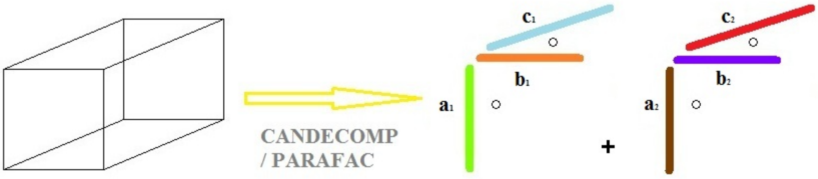

Figure 1.12: Visualization of the PARAFAC Decomposition of a Rank 2 Tensor

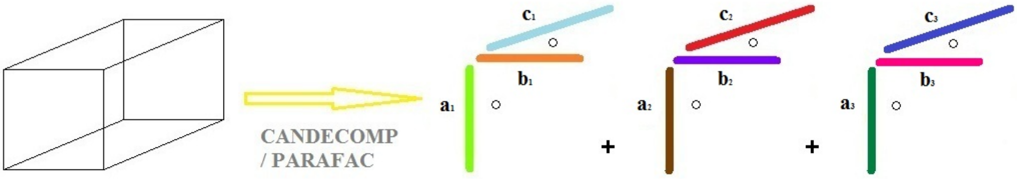

Figure 1.13: Visualization of the PARAFAC Decomposition of a Rank 3 Tensor

We can summarize these ideas mathematically as,

$$\mathcal{T} = \sum_{r=1}^{R} \mathbf{a_r} \circ \mathbf{b_r} \circ \mathbf{c_r}$$

where R is the number of vector triples that compose $\mathcal{T}$ when added up.

This decomposition will also be represented by the following expression

$$\mathcal{T} = [\![\mathbf{A}\ \mathbf{B}\ \mathbf{C}]\!]$$

where the matrices are given by $\mathbf{A} = (\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_R})$, $\mathbf{B} = (\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_R})$, $\mathbf{C} = (\mathbf{c_1}, \mathbf{c_2}, \cdots, \mathbf{c_R})$ with vectors $\mathbf{a}_i$, $\mathbf{b}_i$, and $\mathbf{c}_i$, $i = 1, \cdots, R$ as columns.

Hence we can write the PARAFAC decomposition of a rank R tensor as
$\mathcal{T} = \mathbf{a_1} \circ \mathbf{b_1} \circ \mathbf{c_1} + \mathbf{a_2} \circ \mathbf{b_2} \circ \mathbf{c_2} + \cdots + \mathbf{a_R} \circ \mathbf{b_R} \circ \mathbf{c_R}$
$= \sum_{r=1}^{R} \mathbf{a_r} \circ \mathbf{b_r} \circ \mathbf{c_r}$
$= [\![\mathbf{A}\ \mathbf{B}\ \mathbf{C}]\!]$

1.5.2 HOSVD. Higher Order Singular Value Decomposition

This tensor decomposition is based on the Tucker model, which was introduced by Tucker in 1963 [30] and refined in later articles also by Tucker [31, 32].

The Tucker model is based on the possibility of expressing a tensor as the result of the n-mode product of another tensor of equal size with several matrices.

We can represent this decomposition as shown in the following picture.

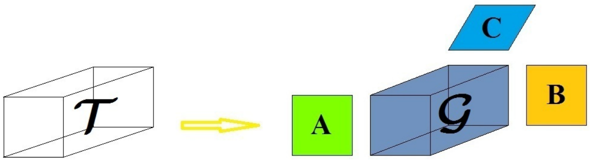

Figure 1.14: Visualization of the Tucker Decomposition of a Tensor

Mathematically, we can write the Tucker model as

$$\mathcal{T} = \mathcal{G} \times_1 \mathbf{A} \times_2 \mathbf{B} \times_3 \mathbf{C}$$

where $\mathcal{G}$ is the *core tensor.*

This approach has been recently further developed by L. De Lathauwer, B. De Moor and J. Vandewalle [8] by setting conditions of orthogonality on the slices of the matrices and on the slices of the core tensor. These developments aims to generalize the SVD matrix decomposition to tensor analysis, so that it can be also defined as Higher Order Singular Value Decomposition.

In the HOSVD of a tensor $\mathcal{T}$, matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ must be orthogonal and will be from now on represented with the letters $\left(\mathbf{U}_{(1)}, \mathbf{U}_{(2)}, \mathbf{U}_{(3)}\right)$. The Higher Order SVD also sets conditions of orthogonality on the core tensor $\mathcal{G}$. This tensor must have mutually orthogonal slices in all the three different modes of $\mathcal{T}$ so that the slices satisfy the equations $\langle \mathbf{G}_1, \mathbf{G}_2 \rangle = 0$, where the matrices are considered in all three modes of the tensor.

We can visualize this decomposition as shown in the following figure

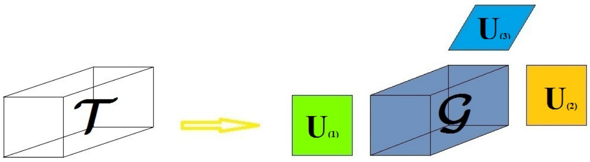

Figure 1.15: Visualization of the HOSVD of a Tensor where $\mathbf{U}_{(1)}, \mathbf{U}_{(2)}, \mathbf{U}_{(3)}$ are orthogonal matrices and $\langle \mathbf{G}_1, \mathbf{G}_2 \rangle = 0$ holds for any matrix slices G_1 and G_2 in an arbitrary mode.

We can write the Higher Order SVD model as

$$\mathcal{T} = \mathcal{G} \times_1 \mathbf{U}_{(1)} \times_2 \mathbf{U}_{(2)} \times_3 \mathbf{U}_{(3)}$$

where $\mathcal{G}$ is the *core tensor* such that the slices along the three modes of the tensor are orthogonal.

Now, we write the SVD of a matrix $\mathbf{A}$ as,

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal eigenvector matrices of $\mathbf{A}\mathbf{A}^T$ and $\mathbf{A}^T\mathbf{A}$ respectively.

We compute the HOSVD of a tensor $\mathcal{T}$ by first computing the SVD of the different matrizations along the different modes $\left(\mathbf{T}_{(1)}, \mathbf{T}_{(2)}, \mathbf{T}_{(3)}\right)$ as seen in subsection 1.3.2 and example 1.1. The result of the multiplication of the tensor $\mathcal{T}$ by the inverses of the first normalized eigenvector matrices $\left(\mathbf{U}_{(1)}, \mathbf{U}_{(2)}, \mathbf{U}_{(3)}\right)$ will produce the core tensor $\mathcal{G}$.

In this algorithm we can see the importance of the matricization process when dealing with tensors.

Example 1.5

In this example we are going to compute the HOSVD of a tensor.

Let us consider the tensor

$$\mathcal{T} = \begin{array}{cccc} & & -1 & 1 \\ & & 1 & 0 \\ 1 & 0 & & \\ 1 & -1 & & \end{array}$$

We want to rewrite the given tensor $\mathcal{T}$ as the n-mode product of a core tensor $\mathcal{G}$ with three orthogonal matrices that will be represented by $\mathbf{U}^1, \mathbf{U}^2$, and $\mathbf{U}^3$.

First we compute the matricizations of the given tensor along the different modes to find the following three different 2×4 matrices.

$$\mathbf{T}_{(1)} = \begin{bmatrix} 1 & 0 & -1 & 1 \\ 1 & -1 & 1 & 0 \end{bmatrix}$$

$$\mathbf{T}_{(2)} = \begin{bmatrix} 1 & 1 & -1 & 1 \\ 0 & -1 & 1 & 0 \end{bmatrix}$$

$$\mathbf{T}_{(3)} = \begin{bmatrix} 1 & 1 & 0 & -1 \\ -1 & 1 & 1 & 0 \end{bmatrix}$$

We can compute that the Singular Value Decompositions of $\mathbf{T}_{(1)}$, $\mathbf{T}_{(2)}$ and $\mathbf{T}_{(3)}$ are given by

$$\mathbf{T}_{(\mathbf{1})} = \mathbf{U}_{(\mathbf{1})}\boldsymbol{\Sigma}_{(\mathbf{1})}(\mathbf{V}_{(\mathbf{1})})^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 & 0 & 0 \\ 0 & \sqrt{3} & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \frac{1}{\sqrt{3}} & 0 & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{3}} & 0 \\ \frac{1}{2\sqrt{3}} & \frac{3+\sqrt{3}}{6} & \frac{1}{2} & \sqrt{\frac{2-\sqrt{3}}{6}} \\ \frac{1}{2} & \sqrt{\frac{2-\sqrt{3}}{6}} & -\frac{1}{2\sqrt{3}} & -\frac{3+\sqrt{3}}{6} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1.7321 & 0 & 0 & 0 \\ 0 & 1.7321 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.57735 & 0 & -0.57735 & 0.57735 \\ 0.57735 & -0.57735 & 0.57735 & 0 \\ 0.28868 & 0.78868 & 0.5 & 0.21132 \\ 0.5 & 0.21132 & -0.28868 & -0.78868 \end{bmatrix}$$

$$\mathbf{T}_{(\mathbf{2})} = \mathbf{U}_{(\mathbf{2})}\boldsymbol{\Sigma}_{(\mathbf{2})}(\mathbf{V}_{(\mathbf{2})})^T = \begin{bmatrix} \sqrt{\frac{\sqrt{5}+5}{10}} & \sqrt{\frac{2}{\sqrt{5}+5}} \\ -\sqrt{\frac{2}{\sqrt{5}+5}} & \sqrt{\frac{\sqrt{5}+5}{10}} \end{bmatrix} \begin{bmatrix} \sqrt{\sqrt{5}+3} & 0 & 0 & 0 \\ 0 & \sqrt{-\sqrt{5}+3} & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \sqrt{\frac{1}{5+\sqrt{5}}} & \sqrt{\frac{1}{5-\sqrt{5}}} & -\sqrt{\frac{1}{5-\sqrt{5}}} & \sqrt{\frac{1}{5+\sqrt{5}}} \\ \sqrt{\frac{1}{5-\sqrt{5}}} & -\sqrt{\frac{1}{5+\sqrt{5}}} & \sqrt{\frac{1}{5+\sqrt{5}}} & \sqrt{\frac{1}{5-\sqrt{5}}} \\ 0.1 & 0.7 & 0.7 & -0.1 \\ 0.7 & -0.1 & -0.1 & -0.7 \end{bmatrix}$$

$$= \begin{bmatrix} 0.85065 & 0.52573 \\ -0.52573 & 0.85065 \end{bmatrix} \begin{bmatrix} 2.2882 & 0 & 0 & 0 \\ 0 & 0.87403 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.37175 & 0.6015 & -0.6015 & 0.37175 \\ 0.6015 & -0.37175 & 0.37175 & 0.6015 \\ 0.1 & 0.7 & 0.7 & -0.1 \\ 0.7 & -0.1 & -0.1 & -0.7 \end{bmatrix}$$

$$\mathbf{T}_{(\mathbf{3})} = \mathbf{U}_{(\mathbf{3})}\boldsymbol{\Sigma}_{(\mathbf{3})}(\mathbf{V}_{(\mathbf{3})})^T = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} \sqrt{3} & 0 & 0 & 0 \\ 0 & \sqrt{3} & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1/\sqrt{3} & 1/\sqrt{3} & 0 & -1/\sqrt{3} \\ 1/\sqrt{3} & -1/\sqrt{3} & -1/\sqrt{3} & 0 \\ \frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & \sqrt{\frac{2}{3}} & 0 \\ \frac{1}{\sqrt{6}} & \frac{1}{\sqrt{6}} & 0 & \sqrt{\frac{2}{3}} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1.7321 & 0 & 0 & 0 \\ 0 & 1.7321 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0.57735 & 0.57735 & 0 & -0.57735 \\ 0.57735 & -0.57735 & -0.57735 & 0 \\ 0.37272 & -0.37272 & 0.81364 & 0 \\ 0.44093 & 0.37272 & 0 & 0.81364 \end{bmatrix}$$

Thus we can write the three orthogonal matrices $\mathbf{U}^1, \mathbf{U}^2, \mathbf{U}^3$ as

$$\mathbf{U}^1 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$\mathbf{U}^2 = \begin{bmatrix} \sqrt{\frac{\sqrt{5}+5}{10}} & \sqrt{\frac{2}{\sqrt{5}+5}} \\ -\sqrt{\frac{2}{\sqrt{5}+5}} & \sqrt{\frac{\sqrt{5}+5}{10}} \end{bmatrix} = \begin{bmatrix} 0.85065 & 0.52573 \\ -0.52573 & 0.85065 \end{bmatrix}$$

$$\mathbf{U}^3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

Now we want to compute the core tensor $\mathcal{G}$ that will be given by the product,

$$\mathcal{G} = \mathcal{T} \times_1 (\mathbf{U}^1)^T \times_2 (\mathbf{U}^2)^T \times_3 (\mathbf{U}^3)^T$$

substituting the expressions found above for $\mathbf{U}^1$, $\mathbf{U}^2$ and $\mathbf{U}^3$ we obtain,

$$\mathcal{G} = \begin{array}{cccc} & & \left(\sqrt{\frac{\sqrt{5}+5}{10}} - \sqrt{\frac{2}{\sqrt{5}+5}}\right) & \left(-\sqrt{\frac{\sqrt{5}+5}{10}} - \sqrt{\frac{2}{\sqrt{5}+5}}\right) \\ & & -\sqrt{\frac{\sqrt{5}+5}{10}} & \sqrt{\frac{2}{\sqrt{5}+5}} \\ \sqrt{\frac{\sqrt{5}+5}{10}} & -\sqrt{\frac{2}{\sqrt{5}+5}} & & \\ \left(\sqrt{\frac{\sqrt{5}+5}{10}} - \sqrt{\frac{2}{\sqrt{5}+5}}\right) & \left(-\sqrt{\frac{\sqrt{5}+5}{10}} - \sqrt{\frac{2}{\sqrt{5}+5}}\right) & & \end{array}$$

$$= \begin{array}{cccc} & & 0.32492 & -1.3764 \\ & & -0.85065 & 0.52573 \\ 0.85065 & -0.52573 & & \\ 0.32492 & -1.3764 & & \end{array}$$

Thus, we can express the HOSVD of the given tensor $\mathcal{T}$ as,

$$\mathcal{T} = \left(\begin{array}{cccc} & & \left(\sqrt{\frac{\sqrt{5}+5}{10}} - \sqrt{\frac{2}{\sqrt{5}+5}}\right) & \left(-\sqrt{\frac{\sqrt{5}+5}{10}} - \sqrt{\frac{2}{\sqrt{5}+5}}\right) \\ & & -\sqrt{\frac{\sqrt{5}+5}{10}} & \sqrt{\frac{2}{\sqrt{5}+5}} \\ \sqrt{\frac{\sqrt{5}+5}{10}} & -\sqrt{\frac{2}{\sqrt{5}+5}} & & \\ \left(\sqrt{\frac{\sqrt{5}+5}{10}} - \sqrt{\frac{2}{\sqrt{5}+5}}\right) & \left(-\sqrt{\frac{\sqrt{5}+5}{10}} - \sqrt{\frac{2}{\sqrt{5}+5}}\right) & & \end{array} \right)$$

$$\times_1 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times_2 \begin{bmatrix} \sqrt{\frac{\sqrt{5}+5}{10}} & \sqrt{\frac{2}{\sqrt{5}+5}} \\ -\sqrt{\frac{2}{\sqrt{5}+5}} & \sqrt{\frac{\sqrt{5}+5}{10}} \end{bmatrix} \times_3 \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$= \left(\begin{array}{cccc} & & 0.32492 & -1.3764 \\ & & -0.85065 & 0.52573 \\ 0.85065 & -0.52573 & & \\ 0.32492 & -1.3764 & & \end{array} \right) \times_1 \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \times_2 \begin{bmatrix} 0.85065 & 0.52573 \\ -0.52573 & 0.85065 \end{bmatrix} \times_3 \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

□

1.6 Rank Issues

We have seen that tensors are closely related to matrices. We see that we can find a multidimensional counterpart for many features of matrices, say dimension, multiplication or decomposition. One of the most interesting aspects of tensors is **rank**, and we will study rank related problems throughout the rest of this report.

1.6.1 Defining Rank

Although the concept of rank when referred to tensors is related to that of matrix rank, there are important differences between them. There is not even a unique way of generalizing the concept from matrices to their higher-order counterpart.

Tensor Rank

We have seen before that a tensor can be rewritten as the sum of several other tensors that arise as the result of computing the outer product of three vectors. We have seen how the PARAFAC decomposition is based on this idea.

We will use the result above to define the rank of a tensor as the minimum number of vector triples that yield the tensor as their sum (see figures 1.11, 1.11, 1.12 above). We see that **rank** is a fundamental concept when talking about the PARAFAC decomposition.

Tensors and k-rank

The k-rank of a matrix $\mathbf{A}$, denoted $k_{\mathbf{A}}$, is defined as the maximum number k such that any k columns in $\mathbf{A}$ are linearly independent [27]. This concept was first introduced by Kruskal, whom it owes the k in its name, and will be fundamental when studying uniqueness.

Example 1.6

$$\text{Let } \mathcal{T} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} -1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

$$= \begin{array}{cccc} & & 0 & 2 \\ & & 2 & 0 \\ 2 & 0 & & \\ 0 & 2 & & \end{array}$$

Hence we can write $\mathcal{T} = [\![\mathbf{A},\ \mathbf{B},\ \mathbf{C}]\!]$ where the component matrices are

$$\mathbf{A} = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix},\ \mathbf{B} = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix},\ \mathbf{C} = \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}$$

Calculating the rank of each component matrix we find,

$$k_{\mathbf{A}} = 2,\ k_{\mathbf{B}} = 2,\ k_{\mathbf{C}} = 2,$$

□

Tensor n-rank

Generalizing the concept of matrix row rank and column rank, Lathauwer [9, **?**] defines the n-rank of a tensor as the dimension of the vector space spanned by the n-mode vectors (fibers) of $\mathcal{T}$, that is, the n-rank of a tensor is given by the column rank of its mode-n matricization.

Thus we can write

$$rank_n\left(\mathcal{T}\right) = \operatorname{rank}\left(\mathcal{T}_{(n)}\right)$$

Example 1.7

Let us consider the tensor

$$\mathcal{T} = \begin{array}{cccc} & & 2 & 0 \\ & & 2 & 0 \\ 0 & 2 & & \\ 0 & 2 & & \end{array}$$

Then, computing the rank of the matricizations of $\mathcal{T}$ along the different modes we can find out the different n-ranks of $\mathcal{T}$,

$$\mathcal{T}_{(1)} = \begin{bmatrix} 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \end{bmatrix} \quad \Longrightarrow$$

1-*rank*
We can see that this matrix has rank 1. Hence we deduce that 1-rank = 1

$$\mathcal{T}_{(2)} = \begin{bmatrix} 0 & 0 & 2 & 2 \\ 2 & 2 & 0 & 0 \end{bmatrix} \quad \Longrightarrow$$

2-*rank*
We can see that this matrix has rank 2. Hence we deduce that 2-rank = 2.

$$\mathcal{T}_{(3)} = \begin{bmatrix} 0 & 0 & 2 & 2 \\ 2 & 2 & 0 & 0 \end{bmatrix} \quad \Longrightarrow$$

3-*rank*
We can see that this matrix has rank 2. Hence we deduce that 3-rank = 2.

□

1.6.2 Problems about Rank

We can easily find the rank of a matrix. However, computing the rank of a tensor is not an easy issue.

These difficulties makes decomposition a much more complicated operation than it is for matrices. In fact, there is no straightforward algorithm to determine the rank of a tensor [2].

This leads to the question of determining how often do tensors of a certain rank occur when considering tensors of a given size. How often do tensors have rank 2 when considering $2 \times 2 \times 2$ tensors? And how often will rank 3 occur?

Typical Rank

In this sense we can define the typical rank of a tensor as the rank that occurs with positive probability for random tensors of a given size. For example, using numerical methods, Kruskal [21] found that rank 2 tensors occur with probability 0.79 when considering $2 \times 2 \times 2$ tensors whereas rank 3 tensors occur only with probability 0.21 when using normal distribution to set the entries of the tensor. We can deduce that both rank 2 and rank 3 are typical rank for $2 \times 2 \times 2$ tensors.

Generic Rank

If the typical rank is unique then we can consider it to be generic, since tensors will have that rank with probability 1.

Rank and Tensor Decompositions

Computing the rank of a tensor will be a fundamental problem when working out tensor decompositions. In general, we cannot compute the PARAFAC decomposition unless we know the rank of the tensor we want to decompose, since we must compute the components simultaneously and these will be vectors, 2×2 matrices or 2×3 matrices depending on the tensor being of rank 1, 2 or 3 respectively. And even if we achieve to compute the corresponding decomposition of a tensor, we still do not know if it is the only one.

In the next Chapter we will explain the issues relating tensor rank and tensor decomposition more closely, whereas in Chapters 3 and 4 we will study uniqueness and degeneracy respectively.

Chapter 2

Tensors

In this chapter we are going to analyze $2 \times 2 \times 2$ tensors concentrating on problems about rank, decompositions and lower rank approximations to a given tensor.

2.1 Computing PARAFAC Components

PARAFAC components are usually estimated by minimization of the quadratic cost function

$$f(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \left\| \mathcal{T} - \sum_{r=1}^{R} \mathbf{a_r} \circ \mathbf{b_r} \circ \mathbf{c_r} \right\|^2 \tag{2.1}$$

When minimizing the function above 2.1, we encounter two different problems:

- If this function becomes zero, then we have computed a decomposition of the tensor $\mathcal{T}$.
- If we can compute the minimum of the function above and it is distinct from zero, then we have computed the best rank R approximation to the given tensor $\mathcal{T}$.

Equation 2.1 is most often minimized by means of the Alternating Least Squares algorithm in which the components are updated mode per mode [10].

Note that the components of the PARAFAC decomposition of $2 \times 2 \times 2$ tensors are either vectors, 2×2 matrices or 2×3 matrices depending on the rank R of the tensor being 1, 2 or 3.

In each case, the component matrices will be defined as,

$$\mathbf{A} = (\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_R})$$

$$\mathbf{B} = (\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_R})$$

$$\mathbf{C} = (\mathbf{c_1}, \mathbf{c_2}, \cdots, \mathbf{c_R})$$

with vectors $\mathbf{a}_i$, $\mathbf{b}_i$, and $\mathbf{c}_i$, $i = 1, \cdots, R$ as columns.

We can rewrite the quadratic cost function given in equation 2.1 as,

$$f(\mathbf{A},\mathbf{B},\mathbf{C}) = \left\|\mathcal{T} - [\![\mathbf{A},\ \mathbf{B},\ \mathbf{C}]\!]\right\|^2 \tag{2.2}$$

Using the Alternating Least Squares algorithm to solve this equation, we find that the ALS fixes **B** and **C** to find **A**. Then takes **A** and **C** to update **B**. Then takes **A** and the updated **B** to update **C**. The updating process is iterated until we find some convergence criterion.

Using equations 1.6, 1.7 and 1.8, we can write equation 2.1 in matricized form, one per mode, as follows,

$$\min_{\mathbf{A}} \left\|\mathcal{T}_{(1)} - \mathbf{A}\left(\mathbf{C}\odot\mathbf{B}\right)^T\right\| \tag{2.3}$$

$$\min_{\mathbf{B}} \left\|\mathcal{T}_{(2)} - \mathbf{B}\left(\mathbf{C}\odot\mathbf{A}\right)^T\right\| \tag{2.4}$$

$$\min_{\mathbf{C}} \left\|\mathcal{T}_{(3)} - \mathbf{C}\left(\mathbf{B}\odot\mathbf{A}\right)^T\right\| \tag{2.5}$$

Solving the equations above, we find that we can update each component matrix **A**, **B** and **C** as follows,

$$\mathbf{A} \longleftarrow \mathcal{T}_{(1)}\left[\left(\mathbf{C}\odot\mathbf{B}\right)^T\right]^\dagger \tag{2.6}$$

$$\mathbf{B} \longleftarrow \mathcal{T}_{(2)}\left[\left(\mathbf{C}\odot\mathbf{A}\right)^T\right]^\dagger \tag{2.7}$$

$$\mathbf{C} \longleftarrow \mathcal{T}_{(3)}\left[\left(\mathbf{B}\odot\mathbf{A}\right)^T\right]^\dagger \tag{2.8}$$

which we can rewrite as in Kolda [2],

$$\mathbf{A} \longleftarrow \mathcal{T}_{(1)}\left(\mathbf{C}\odot\mathbf{B}\right)\left(\mathbf{C}^T\mathbf{C} * \mathbf{B}^T\mathbf{B}\right)^\dagger \tag{2.9}$$

$$\mathbf{B} \longleftarrow \mathcal{T}_{(2)}\left(\mathbf{C}\odot\mathbf{A}\right)\left(\mathbf{C}^T\mathbf{C} * \mathbf{A}^T\mathbf{A}\right)^\dagger \tag{2.10}$$

$$\mathbf{C} \longleftarrow \mathcal{T}_{(3)}\left(\mathbf{B}\odot\mathbf{A}\right)\left(\mathbf{B}^T\mathbf{B} * \mathbf{A}^T\mathbf{A}\right)^\dagger \tag{2.11}$$

2.2 Rank 1 Tensors

We shall begin the analysis of the $2 \times 2 \times 2$ tensors by considering the structure of the rank 1 tensors, which we know can be written as the outer product of 3 vectors.

Example 2.1

We begin studying the rank 1 tensor proposed in Kruskal [21]
We have

$$\mathcal{T} = \begin{array}{cccc} & & 10 & 20 \\ & & 30 & 60 \\ 1 & 2 & & \\ 3 & 6 & & \end{array}$$

We can spot that $\mathcal{T}$ is a rank 1 tensor since all its mode-1 fibers are multiples of the $\mathcal{T}_{i11}$ fiber, given by

$$\mathcal{T}_{i11} = \begin{bmatrix} 1 \\ 3 \end{bmatrix}$$

Using MATLAB's Tensor Toolbox [3], we find the decomposition,

$$\mathcal{T} = 71.0634 \begin{pmatrix} 0.3162 \\ 0.9487 \end{pmatrix} \circ \begin{pmatrix} 0.4472 \\ 0.8944 \end{pmatrix} \circ \begin{pmatrix} 0.0995 \\ 0.9950 \end{pmatrix}$$

where $\mathbf{a} = \begin{pmatrix} 0.3162 \\ 0.9487 \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} 0.4472 \\ 0.8944 \end{pmatrix}$ and $\mathbf{c} = \begin{pmatrix} 0.0995 \\ 0.9950 \end{pmatrix}$ are normalized.

Note that the vectors **a, b** and **c** are multiples of the vectors $\begin{pmatrix} 1 \\ 3 \end{pmatrix}$, $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 10 \end{pmatrix}$ respectively.

□

We are going to use an Alternating Least Squares algorithm to compute the PARAFAC decompositions of different tensors and we will try to reach some results for a general case which can reveal something about the inner structure of rank 1 tensors.

Since we want to compute the PARAFAC decomposition of rank 1 tensors, the component matrices **A**, **B** and **C** will be given by vectors $\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \end{pmatrix}$ and $\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \end{pmatrix}$ so that the updating equations 2.6, 2.7, 2.8 used by the Alternating Least Squares algorithm become,

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \longleftarrow \mathcal{T}_{(1)} \left[\left(\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \odot \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \right)^T \right]^\dagger \tag{2.12}$$

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \longleftarrow \mathcal{T}_{(2)} \left[\left(\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \odot \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \right)^T \right]^\dagger \tag{2.13}$$

$$\begin{pmatrix} c_1 \\ c_2 \end{pmatrix} \longleftarrow \mathcal{T}_{(3)} \left[\left(\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \odot \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \right)^T \right]^\dagger \tag{2.14}$$

which we can rewrite using equations 2.9 , 2.10 , 2.11 , as in Kolda [2],

$$\mathbf{a} \longleftarrow \mathcal{T}_{(1)} \left(\mathbf{c} \odot \mathbf{b}\right) \left(\mathbf{c}^T \mathbf{c} * \mathbf{b}^T \mathbf{b}\right)^\dagger \tag{2.15}$$

$$\mathbf{b} \longleftarrow \mathcal{T}_{(2)} \left(\mathbf{c} \odot \mathbf{a}\right) \left(\mathbf{c}^T \mathbf{c} * \mathbf{a}^T \mathbf{a}\right)^\dagger \tag{2.16}$$

$$\mathbf{c} \longleftarrow \mathcal{T}_{(3)} \left(\mathbf{b} \odot \mathbf{a}\right) \left(\mathbf{b}^T \mathbf{b} * \mathbf{a}^T \mathbf{a}\right)^\dagger \tag{2.17}$$

Note that the expression given by $\left(\mathbf{v}^T \mathbf{v} * \mathbf{w}^T \mathbf{w}\right)^\dagger$ for two vectors $\mathbf{v}, \mathbf{w} \in \mathbb{R}^2$ is a scalar if $\mathbf{v}, \mathbf{w} \neq 0$.

2.2.1 Working out the Decomposition of a Rank 1 Tensor Using ALS

We now take another rank 1 tensor, similar to the one proposed by Kruskal and work the decomposition by hand using the Alternating Least Squares algorithm to see the basic steps of the algorithm.

Take,

$$\mathcal{T} = \begin{array}{cccc} & & 2 & 4 \\ & & 6 & 12 \\ 1 & 2 & & \\ 3 & 6 & & \end{array}$$

We set the starting values $\mathbf{b}_0 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ and $\mathbf{c}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and use equation 2.15 to compute $\mathbf{a}_1$.

$$\mathbf{a}_1 = \mathcal{T}_{(1)} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} \odot \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right) \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}^T \begin{pmatrix} 1 \\ 0 \end{pmatrix} * \begin{pmatrix} 1 \\ -1 \end{pmatrix}^T \begin{pmatrix} 1 \\ -1 \end{pmatrix} \right)^\dagger$$

$$= \begin{bmatrix} 1 & 2 & 2 & 4 \\ 3 & 6 & 6 & 12 \end{bmatrix} \begin{pmatrix} 1 \\ -1 \\ 0 \\ 0 \end{pmatrix} ((1) * (2))^\dagger = \begin{pmatrix} -1/2 \\ -3/2 \end{pmatrix}$$

Now we set $\mathbf{c}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\mathbf{a}_1 = \begin{pmatrix} -1/2 \\ -3/2 \end{pmatrix}$ and use equation 2.16 to compute $\mathbf{b}_1$.

$$\mathbf{b}_1 = \mathcal{T}_{(2)} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} \odot \begin{pmatrix} -1/2 \\ -3/2 \end{pmatrix} \right) \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}^T \begin{pmatrix} 1 \\ 0 \end{pmatrix} * \begin{pmatrix} -1/2 \\ -3/2 \end{pmatrix}^T \begin{pmatrix} -1/2 \\ -3/2 \end{pmatrix} \right)^\dagger$$

$$= \begin{bmatrix} 1 & 3 & 2 & 6 \\ 2 & 6 & 4 & 12 \end{bmatrix} \begin{pmatrix} -1/2 \\ -3/2 \\ 0 \\ 0 \end{pmatrix} ((1) * (5/2))^{\dagger} = \begin{pmatrix} -2 \\ -4 \end{pmatrix}$$

Now we set $\mathbf{b}_1 = \begin{pmatrix} -2 \\ -4 \end{pmatrix}$ and $\mathbf{a}_1 = \begin{pmatrix} -1/2 \\ -3/2 \end{pmatrix}$ and use equation 2.17 to compute $\mathbf{c}_1$.

$$\mathbf{c}_1 = \mathcal{T}_{(3)} \left(\begin{pmatrix} -2 \\ -4 \end{pmatrix} \odot \begin{pmatrix} -1/2 \\ -3/2 \end{pmatrix} \right) \left(\begin{pmatrix} -2 \\ -4 \end{pmatrix}^T \begin{pmatrix} -2 \\ -4 \end{pmatrix} * \begin{pmatrix} -1/2 \\ -3/2 \end{pmatrix}^T \begin{pmatrix} -1/2 \\ -3/2 \end{pmatrix} \right)^{\dagger}$$

$$= \begin{bmatrix} 1 & 3 & 2 & 6 \\ 2 & 6 & 4 & 12 \end{bmatrix} \begin{pmatrix} 1 \\ 3 \\ 4 \\ 6 \end{pmatrix} ((20) * (5/2))^{\dagger} = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

Going through the above algorithm again to find $\mathbf{a}_2, \mathbf{b}_2$ and $\mathbf{c}_2$, we obtain

$$\mathbf{a}_2 = \mathbf{a}_1 = \begin{pmatrix} -1/2 \\ -3/2 \end{pmatrix}, \; \mathbf{b}_2 = \mathbf{b}_1 = \begin{pmatrix} -2 \\ -4 \end{pmatrix} \text{and } \mathbf{c}_2 = \mathbf{c}_1 = \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

and we stop to iterate since the values for $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ converge.
Thus we can write,

$$\begin{array}{cccc} & & 2 & 4 \\ & & 6 & 12 \\ 1 & 2 & & \\ 3 & 6 & & \end{array} = \begin{pmatrix} -1/2 \\ -3/2 \end{pmatrix} \circ \begin{pmatrix} -2 \\ -4 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

2.2.2 General Rank 1 Tensor

We can see that the relation between the different components of each of the three vectors of the PARAFAC decomposition matches the scalings between the 3 ways of the rank 1 tensor.

We are going to consider a general tensor and see what relations must hold between its entries to make it be a rank 1 tensor.

Let $\mathcal{T}$ be a tensor of rank 1, then we can write it as the outer-product of three vectors.

We can assume,

$$\begin{pmatrix} 1 \\ a_2 \end{pmatrix} \circ \begin{pmatrix} 1 \\ b_2 \end{pmatrix} \circ \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{array}{cccc} & & t_{112} & t_{122} \\ & & t_{212} & t_{222} \\ t_{111} & t_{121} & & \\ t_{211} & t_{221} & & \end{array}$$

with $a_2, b_2, c_1, c_2 \neq 0$ and non-zero entries in the tensor.

Thus, we can rewrite the equation above as set of equations in four variables.

$$\begin{aligned} c_1 &= t_{111} \quad (1)\\ c_2 &= t_{112} \quad (2)\\ b_2c_1 &= t_{121} \quad (3)\\ b_2c_2 &= t_{122} \quad (4)\\ a_2c_1 &= t_{211} \quad (5)\\ a_2c_2 &= t_{212} \quad (6)\\ a_2b_2c_1 &= t_{221} \quad (7)\\ a_2b_2c_2 &= t_{222} \quad (8) \end{aligned}$$

Hence we obtain,

$$a_2 = \frac{t_{221}}{t_{121}} = \frac{t_{211}}{t_{111}} = \frac{t_{212}}{t_{112}} = \frac{t_{222}}{t_{122}},\ b_2 = \frac{t_{121}}{t_{111}} = \frac{t_{122}}{t_{112}} = \frac{t_{222}}{t_{212}} = \frac{t_{221}}{t_{211}},\ c_1 = t_{111},\ c_2 = t_{112}$$

We can see that a_2 denotes the ratio between the entries in each mode-1 fiber of the tensor.

In a similar way, we see that b_2 denotes the ratio between the entries in each mode-2 fiber of the tensor.

Finally, we see that c_2 denotes the ratio between the entries in each mode-3 fiber of the tensor.

Now we are going to consider the case when one or more entries of the vectors are zero. We can find that in these cases, the entire corresponding slice of the tensor becomes zero.

Let us consider the case $a_2 = 0$.

If $a_2 = 0$ then we can write

$$\begin{pmatrix}1\\0\end{pmatrix} \circ \begin{pmatrix}1\\b_2\end{pmatrix} \circ \begin{pmatrix}c_1\\c_2\end{pmatrix} = \begin{array}{cccc} & & t_{112} & t_{122}\\ & & 0 & 0\\ t_{111} & t_{121} & & \\ 0 & 0 & & \end{array}$$

and we can see that the lower horizontal slice $\mathbf{T}_{2::}$ has all entries equal to zero.

Hence we can write the equation above as,

$$\begin{aligned} c_1 &= t_{111} \quad (1)\\ c_2 &= t_{112} \quad (2)\\ b_2c_1 &= t_{121} \quad (3)\\ b_2c_2 &= t_{122} \quad (4) \end{aligned}$$

with solutions,

$$b_2 = \frac{t_{121}}{t_{111}} = \frac{t_{122}}{t_{112}}, c_1 = t_{111}, c_2 = t_{112}$$

Similarly, we can find the general form of the decomposition of a rank 1 tensor when other entries of the component vectors are zero.

Thus we can easily check if a given tensor is rank 1 by checking the ratios between the entries in each mode.

Also, we see that we can find the PARAFAC decomposition of a rank 1 tensor by writing the ratios between the entries in each mode as the entries of each corresponding vector of the decomposition. If the entries of one or more slices of the tensor are zero, then the corresponding entry in the component vector is zero.

2.3 Best Lower Rank Approximation to a Tensor

The question of approximating a matrix by another of lower rank is an important issue in matrix analysis. We want to find a matrix of lower rank that is closest to a given matrix $\mathbf{A}$ of rank r.

Let $\mathbf{A}$ be matrix of rank r and let $\mathbf{B}$ be the matrix of rank k with $k < r$ that is closest to $\mathbf{A}$.

Then

$$\|\mathbf{A} - \mathbf{B}\|_F = \sqrt{\sigma_{k+1}^2 + \ldots + \sigma_r^2}$$

where σ_k is the k-singular value of the matrix $\mathbf{A}$.

We can see that the distance between the matrix and its lower rank approximation is given by a function of the relevant singular values.

We can generalize this concept to the tensor analysis. Nevertheless, we will see that although the underlying ideas are the same, there are important features when computing the best rank approximation to a tensor that makes this issue quite different from its matrix counterpart.

When considering tensors, we want to find some tensor $\mathcal{B}$ which has a lower rank than the tensor $\mathcal{T}$ such that the expression,

$$\|\mathcal{T} - \mathcal{B}\|_F \tag{2.18}$$

is minimized.

While the distance between a matrix and its best lower rank approximation is given by a function of some of the singular values, there is no such straightforward result for tensors and we have to compute it using an iterative method.

This is usually done using an Alternating Least Squares algorithm as explained in section 2.1.

The best rank 1 approximation is an important tool when analyzing tensors. This approximation provides a rank 1 tensor as a result; and these tensors can be easily decomposed as the outer product of vectors as we have seen in the previous section. All tensors can be more or less closely approximated by the result of the outer product of three vectors.

The best rank 2 approximation can provide some information about the actual rank of the given tensor. However, this approximation displays special features that make it very different from its rank 1 counterpart.

Once we have computed the best lower-rank approximation $\mathcal{B}$ to a tensor $\mathcal{T}$, we will be interested in knowing how good an approximation it is. If the expression given in equation 2.18 is very small, then we can deduce that the approximation is very close to the tensor $\mathcal{T}$.

2.3.1 Best Rank 1 Approximation

We can find the best rank 1 approximation to a given tensor by minimizing the quadratic cost function given by equation 2.1, which we can write for $R = 1$ as,

$$f(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \|\mathcal{T} - \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}\|^2 \tag{2.19}$$

By minimizing this equation, we will find the rank 1 tensor that is closest to the given tensor $\mathcal{T}$.

We will use the Alternating Least Squares algorithm, as we did for computing the PARAFAC decomposition of a rank 1 tensor in Section 2.2.

Working out the Best Rank 1 Approximation to a Tensor

We will consider the rank 2 tensor proposed in Kruskal [21]

Let,

$$\mathcal{T} = \begin{array}{cccc} & & 0 & 1 \\ & & 1 & 0 \\ 1 & 0 & & \\ 0 & 1 & & \end{array}$$

We set the starting conditions $\mathbf{b}_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $\mathbf{c}_0 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ and use equation 2.15 to compute $\mathbf{a}_1$.

$$\mathbf{a}_1 = \mathcal{T}_{(1)} \left(\begin{pmatrix} -1 \\ 1 \end{pmatrix} \odot \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right) \left(\begin{pmatrix} -1 \\ 1 \end{pmatrix}^T \begin{pmatrix} -1 \\ 1 \end{pmatrix} * \begin{pmatrix} 1 \\ 0 \end{pmatrix}^T \begin{pmatrix} 1 \\ 0 \end{pmatrix} \right)^\dagger$$

$$= \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{pmatrix} -1 \\ 0 \\ 1 \\ 0 \end{pmatrix} ((2) * (1))^\dagger = \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix}$$

Now we set $\mathbf{c}_0 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$ and $\mathbf{a}_1 = \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix}$ and use equation 2.16 to compute $\mathbf{b}_1$.

$$\mathbf{b}_1 = \mathcal{T}_{(2)} \left(\begin{pmatrix} -1 \\ 1 \end{pmatrix} \odot \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix} \right) \left(\begin{pmatrix} -1 \\ 1 \end{pmatrix}^T \begin{pmatrix} -1 \\ 1 \end{pmatrix} * \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix}^T \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix} \right)^\dagger$$

$$= \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{pmatrix} 1/2 \\ -1/2 \\ -1/2 \\ 1/2 \end{pmatrix} ((2) * (1/2))^\dagger = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

Now we set $\mathbf{b}_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ and $\mathbf{a}_1 = \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix}$ and use equation 2.17 to compute $\mathbf{c}_1$.

$$\mathbf{c}_1 = \mathcal{T}_{(3)} \left(\begin{pmatrix} 1 \\ -1 \end{pmatrix} \odot \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix} \right) \left(\begin{pmatrix} 1 \\ -1 \end{pmatrix}^T \begin{pmatrix} 1 \\ -1 \end{pmatrix} * \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix}^T \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix} \right)^\dagger$$

$$= \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{bmatrix} \begin{pmatrix} -1/2 \\ 1/2 \\ 1/2 \\ -1/2 \end{pmatrix} ((2) * (1/2))^\dagger = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

Going through the above algorithm again to find $\mathbf{a}_2, \mathbf{b}_2$ and $\mathbf{c}_2$, we obtain,

$$\mathbf{a}_2 = \mathbf{a}_1 = \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix}, \ \mathbf{b}_2 = \mathbf{b}_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \text{ and } \mathbf{c}_2 = \mathbf{c}_1 = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

and we stop to iterate since the values for $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$ converge.

Thus we can write, after normalizing the vectors $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$

$$\mathcal{B}_1 = 1.4142 \times \begin{pmatrix} -0.7071 \\ 0.7071 \end{pmatrix} \circ \begin{pmatrix} 0.7071 \\ -0.7071 \end{pmatrix} \circ \begin{pmatrix} -0.7071 \\ 0.7071 \end{pmatrix} = \begin{array}{cccc} & & -1/2 & 1/2 \\ & & 1/2 & -1/2 \\ 1/2 & -1/2 & & \\ -1/2 & 1/2 & & \end{array}$$

MATLAB 2.1

The following MATLAB script performs the computation of the best rank 1 approximation to a given tensor.

MATLAB 2.1 Best Rank 1 Approximation

```
M = [1 2 5 6; 3 4 7 8]; %this matrix denotes the mode-1 matricization
                         of the given tensor
T = tensor(M, [2,2,2]) %this line builds up the tensor
B = parafac_als(T,1) % this line computes the approximation
```

However, when setting the rank 2 tensor $\mathcal{T}$ in MATLAB and running the tensor toolbox as shown in the script above, we obtain the following solution,

```
T is a tensor of size 2 x 2 x 2

T(:,:,1) =
     1     0
     0     1
T(:,:,2) =
     0     1
     1     0

CP_ALS:

 Iter  1: fit = 2.924501e-001 fitdelta = 2.9e-001
 Iter  2: fit = 2.928932e-001 fitdelta = 4.4e-004
 Iter  3: fit = 2.928932e-001 fitdelta = 5.7e-014
 Final fit = 2.928932e-001

B is a ktensor of size 2 x 2 x 2
B.lambda = [ 1.4142 ]

B.U{1} =
    0.7071
    0.7071

B.U{2} =
    0.7071
    0.7071

B.U{3} =
    0.7071
    0.7071
>>
```

We define the term **final fit** of an approximation used in the MATLAB Tensor Toolbox as a percentage showing how close the approximation is to the tensor. If we have a final fit of 1, then the approximation fits exactly and represents the tensor itself making equation 2.18 equal to zero. If we have a final fit much smaller than one, then we can deduce that the approximation is not very close to the tensor.

The final fit of an approximation is given by the expression,

$$1 - \frac{\|\mathcal{T} - \mathcal{B}\|_F}{\|\mathcal{T}\|_F}$$

Thus we can write $\mathcal{B}_2$ as

$$\mathcal{B}_2 = 1.4142\times\begin{pmatrix}0.7071\\0.7071\end{pmatrix}\circ\begin{pmatrix}0.7071\\0.7071\end{pmatrix}\circ\begin{pmatrix}0.7071\\0.7071\end{pmatrix} = \begin{array}{cccc} & & 1/2 & 1/2\\ & & 1/2 & 1/2\\ 1/2 & 1/2 & & \\ 1/2 & 1/2 & & \end{array}$$

where the displayed vectors are normalized.

We can see that $\mathcal{B}_1$ and $\mathcal{B}_2$ represent two different rank 1 approximations to the tensor $\mathcal{T}$.

Conclusion

Substituting the expressions found for $\mathcal{T}$, $\mathcal{B}_1$ and $\mathcal{B}_2$ in equation 2.19 and using equation 1.9 to compute the Frobenius norm of the difference between both tensors we can see that,

$$\|\mathcal{T} - \mathcal{B}_1\| = \left\| \begin{array}{cccc} & & 1/2 & 1/2\\ & & 1/2 & 1/2\\ 1/2 & 1/2 & & \\ 1/2 & 1/2 & & \end{array} \right\|_F = \sqrt{2}$$

Similarly, we see that

$$\|\mathcal{T} - \mathcal{B}_2\| = \left\| \begin{array}{cccc} & & 1/2 & 1/2\\ & & 1/2 & 1/2\\ 1/2 & 1/2 & & \\ 1/2 & 1/2 & & \end{array} \right\|_F = \sqrt{2}$$

Since the value for the distance between the tensor $\mathcal{T}$ and its approximations $\mathcal{B}_1$ and $\mathcal{B}_2$ is the same in both cases, we can deduce that $\mathcal{B}_1$ and $\mathcal{B}_2$ represent equally good approximations to $\mathcal{T}$.

We can see that the final fit of both approximations is,

$$1 - \frac{\sqrt{2}}{2} = 0.29289 \text{ to 5 decimal places}$$

as computed by MATLAB.

Thus we deduce that there is more than one best rank 1 approximation to a tensor of rank 2. That is, “best”does not mean “unique”.

Representing Rank 1 Approximations to a Tensor

We are going to represent the minimizing function $f(\mathbf{a}, \mathbf{b}, \mathbf{c})$ as given in equation 2.19. We can find similar representations in Lathauwer, Moor and Vandewalle [9].

We are going to consider different tensors that we define as

$$\mathcal{T} = \begin{array}{cccc} & & \mathtt{E} & \mathtt{F} \\ & & \mathtt{G} & \mathtt{H} \\ \mathtt{A} & \mathtt{B} & & \\ \mathtt{C} & \mathtt{D} & & \end{array}$$

We want to represent the distance between the tensor $\mathcal{T}$ and its rank 1 approximations in a three dimensional space so that we can visualize the behaviour of the minimizing cost function $f(\mathbf{a}, \mathbf{b}, \mathbf{c})$.

Thus, we must represent $f(\mathbf{a}, \mathbf{b}, \mathbf{c})$ as a function of only two variables. This means that we must assume arbitrary values for some entries of the component vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$ so that the three dimensional representation is possible.

Let us assume that the rank 1 tensor given by the vector outer product $\mathbf{a} \circ \mathbf{b} \circ \mathbf{c}$ is formed by the vectors $\mathbf{a} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix}$, $\mathbf{c} = \begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix}$.

We have chosen vectors $\mathbf{b}$ and $\mathbf{c}$ to be normalized and depending only on one variable respectively and we have chosen an arbitrary vector $\mathbf{a}$ in order to be able to represent $f(\mathbf{a}, \mathbf{b}, \mathbf{c})$ as a function of two variables, α and β.

Thus, we can write

$$\mathbf{a} \circ \mathbf{b} \circ \mathbf{c} = \begin{array}{cccc} & & \cos\alpha\sin\beta & \sin\alpha\sin\beta \\ & & -\cos\alpha\sin\beta & -\sin\alpha\sin\beta \\ \cos\alpha\cos\beta & \sin\alpha\cos\beta & & \\ -\cos\alpha\cos\beta & -\sin\alpha\cos\beta & & \end{array}$$

We find that we can represent the value of the minimizing function 2.19 $\|\mathcal{T} - \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}\|_F^2$ against the values of α and β as,

$$f(\mathbf{a}, \mathbf{b}, \mathbf{c}) = (\mathtt{A} - \cos\alpha\cos\beta)^2 + (\mathtt{B} - \sin\alpha\cos\beta)^2 + (\mathtt{C} + \cos\alpha\cos\beta)^2 + (\mathtt{D} + \sin\alpha\cos\beta)^2$$
$$+ (\mathtt{E} - \cos\alpha\sin\beta)^2 + (\mathtt{F} - \sin\alpha\sin\beta)^2 + (\mathtt{G} + \cos\alpha\sin\beta)^2 + (\mathtt{H} + \sin\alpha\sin\beta)^2$$

MATLAB 2.2

We can sketch a graph of the minimizing function given above by running through the following MATLAB code.

MATLAB 2.2 Code for the Minimizing Function Graph

```
A = t_111; B = t_121; C = t_211; D = t_221;
E = t_112; F = t_122; G = t_212; H = t_222;
T_1 = [A,B,E,F;C,D,G,H];
T = tensor(T_1,[2,2,2])
[x,y] = meshgrid([-pi:0.2:pi]);
N =(A - ((cos(x)).*(cos(y)))).^2 + (B - ((cos(x)).*(cos(y)))).^2
+ (C + ((cos(x)).*(cos(y)))).^2 + (D + ((cos(x)).*(cos(y)))).^2
+ (E - ((cos(x)).*(cos(y)))).^2 + (F - ((cos(x)).*(cos(y)))).^2
+ (G + ((cos(x)).*(cos(y)))).^2 + (H + ((cos(x)).*(cos(y)))).^2;
surfc(x,y,N)
```

This code is constructed setting vector $\mathbf{a} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$ and it can be modified to set anther vector $\mathbf{a}$ as an initial assumption for the problem.

The variables α and β are sketched in a range from $-\pi$ to π

Example 2.2

We are going to consider the tensor given by $\mathcal{T} = \begin{array}{cccc} & & 0 & 1 \\ & & 1 & 0 \\ 1 & 0 & & \\ 0 & 1 & & \end{array}$ and the vectors $\mathbf{a} = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$, $\mathbf{b} = \begin{pmatrix} \cos\alpha \\ \sin\alpha \end{pmatrix}$, $\mathbf{c} = \begin{pmatrix} \cos\beta \\ \sin\beta \end{pmatrix}$.

We can represent the value of the minimizing function given in equation 2.19, that is, $f(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \|\mathcal{T} - \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}\|^2$ against the values of α and β as shown in the following figure.

We can see that the best rank 1 approximations highlighted in deep blue are given by the minima of the function.

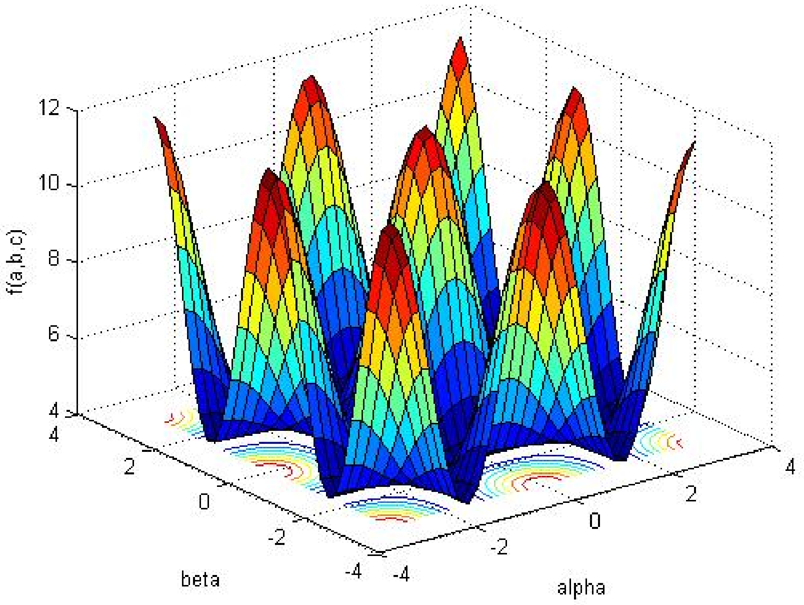

Figure 2.1: Graph of the Rank 1 Approximations to the Rank 2 Tensor proposed in Kruskal [21]

We can see that there are several minima which would provide Best Rank 1 Approximations to the tensor $\mathcal{T}$.

However, when changing the choice for vector **a** to the vector $\mathbf{a} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$, we obtain the following representation of the distance between the tensor and the rank 1 tensors.

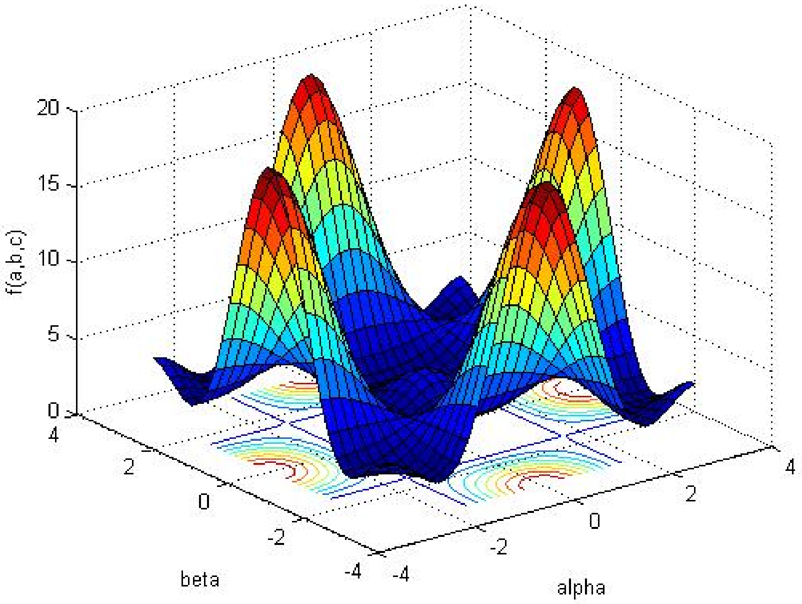

Figure 2.2: Graph of the Rank 1 Approximations to the Rank 2 Tensor proposed in Kruskal [21]

Comparing both representations, we see that the choice of vector **a** improves the computed approximations.

□

Example 2.3

Now, let us consider the tensor given by $\mathcal{T} = \begin{array}{cccc} & & 5 & 6 \\ & & 7 & 8 \\ 1 & 2 & & \\ 3 & 4 & & \end{array}$

As in the previous example, we can represent the value of the minimizing function $f(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \|\mathcal{T} - \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}\|^2$ against the values of α and β as shown in the following graph.

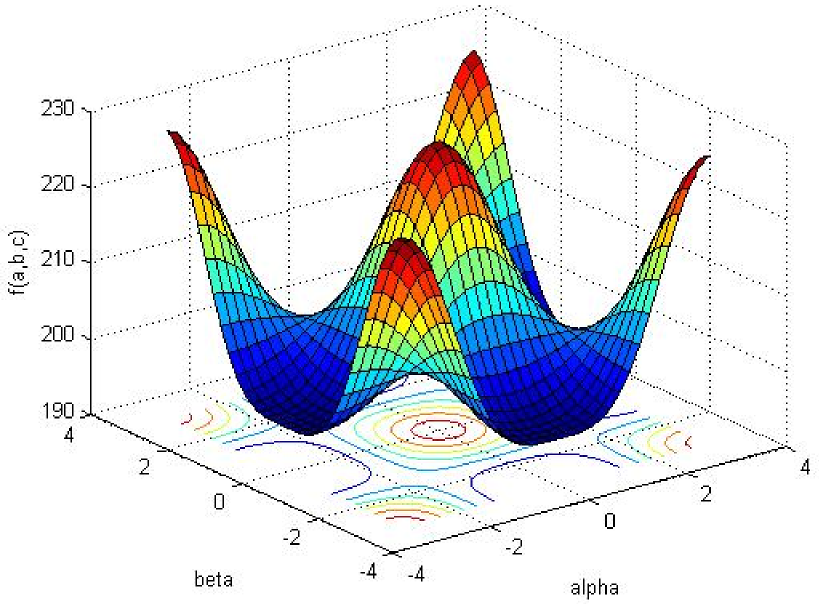

Figure 2.3: Graph of the Rank 1 Approximations to the Tensor given in Example 1.1

We can see that the good rank 1 approximations highlighted in deep blue are attained for many different vectors $\mathbf{a}, \mathbf{b}, \mathbf{c}$.

Now, when computing the best rank 1 approximation using MATLAB as explained in MATLAB 2.1 we find that the proposed best rank 1 approximation is given by,

$$\mathcal{B} = 14.2254 \times \begin{pmatrix} 0.5663 \\ 0.8242 \end{pmatrix} \circ \begin{pmatrix} 0.6406 \\ 0.7679 \end{pmatrix} \circ \begin{pmatrix} 0.3759 \\ 0.9266 \end{pmatrix}$$

Modifying the MATLAB code given in MATLAB 2.2 so that the initial guess for vector **a** matches the proposed best rank 1 approximation, we obtain the graph,

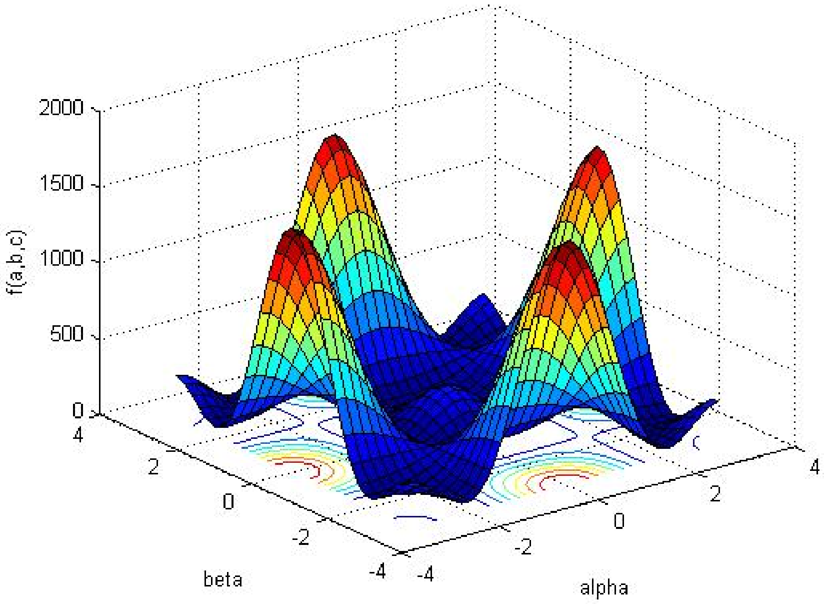

Figure 2.4: Graph of the Rank 1 Approximations to the Tensor given in Example 1.1 with new guess for **a**

As in the previous example, we can see when comparing both representations that the improvement in the choice of vector **a** corresponds to an improvement in the computed approximations so that we find more vectors representing a good approximation to the tensor.

□

2.3.2 Best Rank 2 Approximation

We saw before that we can write the PARAFAC decomposition of a rank R tensor as $\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ where the matrices are given by $\mathbf{A} = (\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_R})$, $\mathbf{B} = (\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_R})$, $\mathbf{C} = (\mathbf{c_1}, \mathbf{c_2}, \cdots, \mathbf{c_R})$ with vectors $\mathbf{a}_i$, $\mathbf{b}_i$, and $\mathbf{c}_i$, $i = 1, \cdots, R$ as columns.

Hence we can compute the rank R approximation to a given tensor by minimizing the function given by equation 2.2 which we rewrite here

$$f(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \left\| \mathcal{T} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \right\|_F^2$$

By computing the minimum of this equation, we will find the rank R tensor that is nearest to the given tensor $\mathcal{T}$.

However, as we will explain later in Chapter 4, this function sometimes fails to have a minimum since the tensor $\mathcal{T}$ can be arbitrarily well approximated by a lower rank tensor and we can deduce that the best lower rank approximation does not exist. This is in fact the case by the rank 2 approximation of a rank 3 tensor. This special behaviour will be defined as **degeneracy** and will be closely related to certain problems arising at very long iterative sequences that do not come to an end.

However, we can define a MATLAB code to compute rank 2 approximations to a tensor.

——— *MATLAB 2.3* ———

The following MATLAB script performs the computation of rank 2 approximations to a given tensor.

MATLAB 2.3 Best Rank 2 Approximation

```
M = [1 2 5 6; 3 4 7 8]; %this matrix denotes the mode-1 matricization
of the given tensor
T = tensor(M, [2,2,2]) %this line builds up the tensor
B = parafac_als(T,2) % this line computes the approximation
```

We will see that we obtain different results each time we run the code.

```
Final fit = 9.764462e-001
B is a ktensor of size 2 x 2 x 2
B.lambda = [ 426.5394      419.2111 ]
B.U{1} =
    0.5279    0.5253
    0.8493    0.8509
B.U{2} =
    0.6245   -0.6234
    0.7810   -0.7819
B.U{3} =
   -0.5847   -0.6078
    0.8112    0.7941
```

```
>>
```

Another result

```
Final fit = 9.980868e-001
B is a ktensor of size 2 x 2 x 2
B.lambda = [ 23.1101      10.4251 ]
B.U{1} =
    0.6497   -0.7761
    0.7602   -0.6306
B.U{2} =
    0.6792    0.7434
    0.7339    0.6688
B.U{3} =
    0.6009    0.8532
    0.7993    0.5216
```

We can rewrite these results as,

$$\begin{aligned}
\mathcal{B}_1 = 426.5394 \times \begin{pmatrix} 0.5279 \\ 0.8493 \end{pmatrix} \circ \begin{pmatrix} 0.6245 \\ 0.7810 \end{pmatrix} \circ \begin{pmatrix} -0.5847 \\ 0.8112 \end{pmatrix} \\
+419.2111 \times \begin{pmatrix} 0.5253 \\ 0.8509 \end{pmatrix} \circ \begin{pmatrix} -0.6234 \\ -0.7819 \end{pmatrix} \circ \begin{pmatrix} -0.6078 \\ 0.7941 \end{pmatrix} \\
= \left[\!\!\left[\begin{pmatrix} 225.17 & 220.21 \\ 362.26 & 356.71 \end{pmatrix}, \begin{pmatrix} 266.37 & -261.34 \\ 333.13 & -327.78 \end{pmatrix}, \begin{pmatrix} -249.4 & -254.80 \\ 346.01 & 332.90 \end{pmatrix} \right]\!\!\right]
\end{aligned} \tag{2.20}$$

We can see that the vectors in each component matrix in $\mathcal{B}_1$ are very similar up to sign change, with differences between corresponding entries of less than 4 %.

$$\begin{aligned}
\mathcal{B}_2 = 23.1101 \times \begin{pmatrix} 0.6497 \\ 0.7602 \end{pmatrix} \circ \begin{pmatrix} 0.6792 \\ 0.7339 \end{pmatrix} \circ \begin{pmatrix} 0.6009 \\ 0.7993 \end{pmatrix} \\
+10.4251 \times \begin{pmatrix} -0.7761 \\ -0.6306 \end{pmatrix} \circ \begin{pmatrix} 0.7434 \\ 0.6688 \end{pmatrix} \circ \begin{pmatrix} 0.8532 \\ 0.5216 \end{pmatrix} \\
= \left[\!\!\left[\begin{pmatrix} 15.015 & -80909 \\ 17.568 & -6.5741 \end{pmatrix}, \begin{pmatrix} 15.696 & 7.75 \\ 16.961 & 6.9723 \end{pmatrix}, \begin{pmatrix} 13.887 & 8.8947 \\ 18.472 & 5.4377 \end{pmatrix} \right]\!\!\right]
\end{aligned} \tag{2.21}$$

We can see that both approximations are very close to the original tensor $\mathcal{T}$, but there is still a significant difference in the final fits computed by MATLAB for both approximations.

In this sense, it is interesting to consider the concept of **border rank** which is defined as the minimum number of rank 1 tensors that are sufficient to approximate the given tensor with arbitrarily small nonzero error.

Thus, we will find that some rank 3 tensors have border rank 2 since they can be arbitrarily well approximated by a rank 2 tensor. This feature will be further studied in chapter 4.

2.4 Higher Rank Tensors

It is interesting to see that when adding the two best rank 1 approximations found in the previous chapter for the rank 2 tensor proposed in Kruskal [21], we obtain the tensor itself and we can deduce that the two rank 1 tensors $\mathcal{B}_1$ and $\mathcal{B}_2$ constitute a PARAFAC decomposition of the tensor.

However, calculating the PARAFAC decomposition of a rank 2 or rank 3 tensor usually involves much more difficult calculations.

Since PARAFAC components must be computed simultaneously, this means working with 2×2 matrices instead of vectors in the case of rank 2 tensors and with 2×3 matrices in the case of rank 3 tensors if an iterative method like the ALS is used to compute the decomposition of the tensor. It also means that we should determine the rank of the tensor *before* beginning to compute its decomposition.

Note that the updating formulas for the ALS algorithm given in section 2.1 give rise to long computations when considering 2×2 or 2×3 matrices instead of vectors as in section 2.2. It is important to remark that these long computations involve calculating the pseudoinverse of a matrix which can lead to problems of discontinuity, so that the ALS will not work as well as when computing the rank 1 decomposition of a tensor as we saw in section 2.2.

Indeed, the usual method to compute the PARAFAC components of a tensor given by the ALS algorithm could be improved by developing an analytical method to compute the exact decomposition of a tensor without using any iterative method which will in most cases lead to numerical errors.

2.4.1 Criteria to Determine the Rank of a Tensor

In the literature we can find different criteria to determine the rank of a given tensor. Nevertheless, there is no such a straightforward algorithm to determine the rank of higher rank tensors as we studied for rank 1 tensors, where we can indeed spot that a tensor is rank 1 by considering the ratio between the entries.

Numerical Criterion

We use Kolda's Tensor Toolbox [3] to compute the best rank 2 approximation to a given tensor. If we can compute a very close fit for the approximation, we will deduce that the given tensor has rank 3 and rank 1 or 2 otherwise.

Example 2.4

1. Let us consider the Rank 2 tensor proposed by Kruskal in [21],

$$\mathcal{T} = \begin{array}{cccc} & & 0 & 1 \\ & & 1 & 0 \\ 1 & 0 & & \\ 0 & 1 & & \end{array}$$

   Setting this tensor in the Tensor Toolbox of MATLAB [3] and running the MATLAB code given above in MATLAB 2.3, computes a best rank 2 approximation with final fit = 9.999999e-001 after 6 iterations.

   This means that the given tensor fits almost perfectly into the decomposition as a sum of two rank 1 tensors and thus we can deduce that the tensor is rank 2.

2. Now let us consider the tensor proposed in Kolda [2]

$$\mathcal{T} = \begin{array}{cccc} & & 0 & 1 \\ & & -1 & 0 \\ 1 & 0 & & \\ 0 & 1 & & \end{array}$$

   Running the MATLAB code given in MATLAB 2.3 computes the best rank 2 approximation with final fit $4.965562e-001$ after 35 iterations. Hence we can deduce that the tensor does not fit very well into this "sum of two rank 1 tensors decomposition". On the other hand we can see that the tensor toolbox computes a best rank 3 approximation with fit $9.999938e-001$ after 6 iterations. Thus we can deduce that the given tensor has rank 3.

3. Let us consider again the tensor we studied in Examples 1.1 and 2.3 given by,

$$\mathcal{T} = \begin{array}{cccc} & & 5 & 6 \\ & & 7 & 8 \\ 1 & 2 & & \\ 3 & 4 & & \end{array}$$

   The tensor toolbox computes a rank 2 approximation with fit= 9.566761e-001 whereas it computes a best rank 3 approximation with fit =9.998599e-001.

   We see that it is impossible to decide whether it is rank 2 or 3 using this iterative method.

This example demonstrates that numerical studies do not always reveal the nature of the tensor.

□

Criterion based on Hyperdeterminants

Tensors where first approached by Cayley [5], which he defined as hypermatrices. The inner structure of 2 × 2 × 2 tensors is governed by a quartic polynomial Δ which Cayley [5] discovered in 1845 and was later further developed by Gelfand, Kapranov and Zelvinsky [15] already in the 20th century. This polynomial is given by the hyperdeterminant of the 2 × 2 × 2 tensor.

Let a 2 × 2 × 2 tensor be defined as

$$\mathcal{T} = \begin{array}{cccc} & & t_{112} & t_{122} \\ & & t_{212} & t_{222} \\ t_{111} & t_{121} & & \\ t_{211} & t_{221} & & \end{array}$$

then we define its hyperdeterminant Δ as

$$\begin{aligned} \Delta(\mathcal{T}) = & \left(t_{111}^2 t_{222}^2 + t_{112}^2 t_{221}^2 + t_{121}^2 t_{212}^2 + t_{122}^2 t_{211}^2\right) \\ & -2\,(t_{111}t_{112}t_{221}t_{222} + t_{111}t_{121}t_{212}t_{222} + t_{111}t_{122}t_{211}t_{222} \\ & +t_{112}t_{121}t_{212}t_{221} + t_{112}t_{122}t_{221}t_{211} + t_{121}t_{122}t_{212}t_{211}) \\ & +4\,(t_{111}t_{122}t_{212}t_{221} + t_{112}t_{121}t_{211}t_{222}) \end{aligned} \tag{2.22}$$

The hyperdeterminant $\Delta(\mathcal{T})$ for the 2 × 2 × 2 tensor $\mathcal{T}$ described above is known as *Kruskal polynomial* by the psychometrics community and is so referred to in the literature [7].

In fact, Kruskal [21] discusses the use a of *certain polynomial* that is defined only for 2 × 2 × 2 arrays as a method to determine the rank of the array. Positive values of the polynomial $\Delta(\mathcal{T})$ signify that the rank of the array is 2 while negative values account for a rank 3 tensor. If the polynomial $\Delta(\mathcal{T})$ is zero, the rank may be either 0, 1 or 3 and cannot be accurately determined using this method.

Ten Berge [28] sharpens the definition of the diagnostic properties proposed by Kruskal [21] and considers only arrays where at least one slice in non-singular and the other slice is non-zero and not proportional to the other.

In the class of arrays proposed by Ten Berge [28], having $\Delta(\mathcal{T}) > 0$ is the necessary and sufficient conditions for rank 2. If $\Delta(\mathcal{T}) < 0$, then the array has rank 3. The arrays with $\Delta(\mathcal{T}) = 0$ are special because although their rank is 3, they may be approximated arbitrarily well by arrays whose rank is 2. These arrays will be considered more closely in Chapter 4.

Example 2.5

1. Let us consider the rank 1 tensor proposed in Kruskal [21]

$$\mathcal{T} = \begin{array}{cccc} & & 10 & 20 \\ & & 30 & 60 \\ 1 & 2 & & \\ 3 & 6 & & \end{array}$$

Computing its hyperdeterminant we find,

$$\Delta(\mathcal{T}) = 0$$

2. Let us now consider the tensor given by

$$\mathcal{T} = \begin{array}{cccc} & & 10 & 30 \\ & & 8 & 24 \\ 7 & 21 & & \\ 6 & 18 & & \end{array}$$

Computing its hyperdeterminant we find,

$$\Delta(\mathcal{T}) = 0$$

In this case, Kruskal's criterium [21] does not provide further information about the rank of the tensor. However, we find that there are singular slices so that Ten Berge's condition [28] is not satisfied. Since we can spot the the tensor has not rank 1, we deduce that it will have rank 2.

3. Let us now consider the tensor given by

$$\mathcal{T} = \begin{array}{cccc} & & 5 & 6 \\ & & 7 & 8 \\ 1 & 2 & & \\ 3 & 4 & & \end{array}$$

Computing its hyperdeterminant we find,

$$\Delta(\mathcal{T}) = 0$$

In this case, as in the tensor considered above, we can see that Kruskal's criterium does not provide further information about the rank of the tensor.

Since the conditions set by Ten Berge [28] are met and there are no non-singular slices, we can decide that the tensor is rank 3. This tensor will however show some special behaviour when computing lower rank approximations.

$\square$

The results given in this example show that tensors such that $\Delta() = 0$ can present different ranks. This example shows also that Kruskal's criterium in not enough to evaluate the rank of the tensor in these cases and we need to study more closely its inner structure.

Example 2.6

Let us consider the rank 2 tensor proposed in Kruskal [21]

$$\mathcal{T} = \begin{array}{cccc} & & 0 & 1 \\ & & 1 & 0 \\ 1 & 0 & & \\ 0 & 1 & & \end{array}$$

Computing its hyperdeterminant we find,

$$\Delta(\mathcal{T}) = 4$$

We can see that computing the value of the hyperdeterminant gives a positive value as expected for a rank 2 tensor.

□

Example 2.7

Let us now consider the rank 3 tensor proposed in Kolda [2]

$$\mathcal{T} = \begin{array}{cccc} & & 0 & 1 \\ & & -1 & 0 \\ 1 & 0 & & \\ 0 & 1 & & \end{array}$$

Computing its hyperdeterminant we find,

$$\Delta(\mathcal{T}) = -4$$

We can see that computing the value of the hyperdeterminant provides a negative value as expected for a rank 3 tensor.

□

These examples demonstrate that if the hyperdeterminant of a tensor turns out to be $\Delta(\mathcal{T}) = 0$, then we have to study more closely the inner structure of the tensor to discover its rank. We can see that setting the condition explained by Ten Berge [28] on the slices being non-singular, the inner structure becomes clearer and we can decide the actual rank of the tensor.

Criterium based on Eigenvectors

Ten Berge [28] claims that a sufficient condition for a given tensor $\mathcal{T}$ of size $p \times p \times 2$ to have rank p is that the matrix given by the product of the mode-3 slices (frontal slices) $\mathbf{T}_{::\mathbf{2}}(\mathbf{T}_{::\mathbf{1}})^{\mathbf{-1}}$ has p distinct real eigenvalues.

Then it follows that a $2 \times 2 \times 2$ tensor will have rank 2 is the slice product $\mathbf{T}_{::\mathbf{2}}(\mathbf{T}_{::\mathbf{1}})^{\mathbf{-1}}$ has 2 distinct real eigenvalues.

Stegeman [24] gives a more accurate rank condition for real-valued $p \times p \times 2$ tensors.

Let $\mathcal{T}$ be a real-valued $p \times p \times 2$ array with $p \times p$ slices $\mathbf{T}_{::\mathbf{2}}$ and $\mathbf{T}_{::\mathbf{1}}$. Suppose $(\mathbf{T}_{::\mathbf{1}})^{-1}$ exists. The following statement holds:

- If $\mathbf{T}_{::\mathbf{2}}(\mathbf{T}_{::\mathbf{1}})^{-\mathbf{1}}$ has p real eigenvalues and is diagonalizable, then $\text{rank}(\mathcal{T}) = p$.
- If $\mathbf{T}_{::\mathbf{2}}(\mathbf{T}_{::\mathbf{1}})^{-\mathbf{1}}$ has at least one pair of complex eigenvalues, then $\text{rank}(\mathcal{T}) \geq p+1$
- If $\mathbf{T}_{::\mathbf{2}}(\mathbf{T}_{::\mathbf{1}})^{-\mathbf{1}}$ has p real eigenvalues but is not diagonalizable, then $\text{rank}(\mathcal{T}) \geq p+1$

Thus we can deduce that a $2 \times 2 \times 2$ tensor will have rank 2 if the corresponding product $\mathbf{T}_{::\mathbf{2}}(\mathbf{T}_{::\mathbf{1}})^{-\mathbf{1}}$ has two real eigenvalues and is diagonalizable and will have rank 3 if the slice product is not diagonalizable or has complex eigenvalues.

Example 2.8

1. Let us consider the rank 2 tensor proposed in Kruskal [21],

   $$\text{Let } \mathcal{T} = \begin{array}{cccc} & & 0 & 1 \\ & & 1 & 0 \\ 1 & 0 & & \\ 0 & 1 & & \end{array}$$

   we have $\mathbf{T}_{::\mathbf{1}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\mathbf{T}_{::\mathbf{2}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

   Thus the relevant slice product is given by,

   $$\mathbf{T}_{::\mathbf{2}}(\mathbf{T}_{::\mathbf{1}})^{-1} = \left(\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\right)\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)^{-1} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

   with eigenvalues given by $\lambda_1 = 1$ and $\lambda_2 = -1$.

   Since the eigenvalues are real and distinct we deduce that the tensor is rank 2 as expected.

2. Let us consider the rank 3 tensor proposed in Kolda [2],

   $$\text{Let } \mathcal{T} = \begin{array}{cccc} & & 0 & 1 \\ & & -1 & 0 \\ 1 & 0 & & \\ 0 & 1 & & \end{array}$$

   we have $\mathbf{T}_{::\mathbf{1}} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$ and $\mathbf{T}_{::\mathbf{2}} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

   Thus the relevant slice product is given by,

   $$\mathbf{T}_{::\mathbf{2}}(\mathbf{T}_{::\mathbf{1}})^{-1} = \left(\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}\right)\left(\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right)^{-1} = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

   with eigenvalues given by $\lambda_1 = i$ and $\lambda_2 = -i$.

   Since the eigenvalues are complex we deduce that the tensor is rank 3 as expected.

3. Let us consider the tensor given by,

$$\mathcal{T} = \begin{array}{cccc} & & 5 & 6 \\ & & 7 & 8 \\ 1 & 2 & & \\ 3 & 4 & & \end{array}$$

we have $\mathbf{T}_{::\mathbf{1}} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$ and $\mathbf{T}_{::\mathbf{2}} = \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$

Thus the relevant slice product is given by,

$$\mathbf{T}_{::\mathbf{2}}(\mathbf{T}_{::\mathbf{1}})^{-1} = \left(\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}\right)\left(\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\right)^{-1} = \begin{bmatrix} -1 & 2 \\ -2 & 3 \end{bmatrix}$$

with an only eigenvalue given by $\lambda = 1$

Since the eigenvalue has multiplicity two and the matrix is not diagonalizable we can deduce that the tensor is rank 3.

4. Let us again consider the tensor given by

$$\mathcal{T} = \begin{array}{cccc} & & 10 & 30 \\ & & 8 & 24 \\ 7 & 21 & & \\ 6 & 18 & & \end{array}$$

We can see that both frontal slices are singular and this is also the case for the horizontal slices.

Thus we must consider the lateral slices.

The relevant slice product is

$$\mathbf{T}_{::\mathbf{2}}(\mathbf{T}_{::\mathbf{1}})^{-1} = \left(\begin{bmatrix} 21 & 30 \\ 18 & 24 \end{bmatrix}\right)\left(\begin{bmatrix} 7 & 10 \\ 6 & 8 \end{bmatrix}\right)^{-1} = \begin{bmatrix} 3 & 0 \\ 0 & 3 \end{bmatrix}$$

with an only eigenvalue given by $\lambda = 3$

Since the eigenvalue has multiplicity two and the matrix is diagonalizable we can deduce that the tensor is rank 2.

□

Criterium based on the Relationships between the Entries of the Component Vectors

The criteria considered above do help to determine the actual rank of the tensor. They do not give however any help to compute the decomposition of the tensor. In the next section we will compute a certain polynomial based on the entries of the component vectors of the PARAFAC decomposition of the tensor which will throw information about the rank of the tensor. We also explain an analytical method to compute the PARAFAC decomposition of a tensor.

2.4.2 General Form of a Rank 2 Tensor

Just as we have computed a general form for rank 1 tensors, we will now try to find out the relationships between the entries of the tensor and the entries of the vectors that form its PARAFAC decomposition so that we can compute the PARAFAC decomposition of a rank 2 tensor over $\mathbb{R}^2$ and of a rank 3 tensor over $\mathbb{C}$. We also want to find a more straightforward, analytic way of deciding if a given tensor is rank 2 or 3.

Let $\mathcal{T}$ be a tensor of rank 2, then we can write it as a sum of two outer-products of vector triples.

Then we can write,

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \circ \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \circ \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} d_1 \\ d_2 \end{pmatrix} \circ \begin{pmatrix} e_1 \\ e_2 \end{pmatrix} \circ \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{array}{cccc} & & t_{112} & t_{122} \\ & & t_{212} & t_{222} \\ t_{111} & t_{121} & & \\ t_{211} & t_{221} & & \end{array} \tag{2.23}$$

Without loss of generality we assume that the entries a_1, b_1, d_1 and e_1 are nonzero, and we scale the component vectors so that we can write the decomposition of $\mathcal{T}$ as,

$$\begin{pmatrix} 1 \\ a_2 \end{pmatrix} \circ \begin{pmatrix} 1 \\ b_2 \end{pmatrix} \circ \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} 1 \\ d_2 \end{pmatrix} \circ \begin{pmatrix} 1 \\ e_2 \end{pmatrix} \circ \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{array}{cccc} & & t_{112} & t_{122} \\ & & t_{212} & t_{222} \\ t_{111} & t_{121} & & \\ t_{211} & t_{221} & & \end{array} \tag{2.24}$$

Note that the cases when any of the entries a_1, b_1, d_1 and e_1 are zero can be considered as special cases that will not change the general setting of the problem considered.

We can rewrite the equation above as set of equations in eight variables.

$$\begin{aligned} c_1 + f_1 &= t_{111} \quad (1) \\ c_2 + f_2 &= t_{112} \quad (2) \\ b_2c_1 + e_2f_1 &= t_{121} \quad (3) \\ b_2c_2 + e_2f_2 &= t_{122} \quad (4) \\ a_2c_1 + d_2f_1 &= t_{211} \quad (5) \\ a_2c_2 + d_2f_2 &= t_{212} \quad (6) \\ a_2b_2c_1 + d_2e_2f_1 &= t_{221} \quad (7) \\ a_2b_2c_2 + d_2e_2f_2 &= t_{222} \quad (8) \end{aligned}$$

Solving the equations when some entries of the component vectors are equal

If either $a_2 = d_2$, $b_2 = e_2$ or $\frac{c_1}{c_2} = \frac{f_1}{f_2}$ then we can consider the equation above as a matrix equation since we find that some equations are multiples of each other.

Let us consider the case when we have $b_2 = e_2$. We can see that the other cases considered above can be solved in a similar way as will be now explained for this case.

Let $b_2 = e_2$, then we can rewrite equations (1), (2), (3), (4), (5), (6), (7) and (8) as,

$$\begin{aligned} c_1 + f_1 &= t_{111} \quad (1)\\ c_2 + f_2 &= t_{112} \quad (2)\\ b_2(c_1 + f_1) &= t_{121} \quad (3)\\ b_2(c_2 + f_2) &= t_{122} \quad (4)\\ a_2c_1 + d_2f_1 &= t_{211} \quad (5)\\ a_2c_2 + d_2f_2 &= t_{212} \quad (6)\\ b_2(a_2c_1 + d_2f_1) &= t_{221} \quad (7)\\ b_2(a_2c_2 + d_2f_2) &= t_{222} \quad (8) \end{aligned}$$

We can see that if $b_2 = e_2$, then equation (3), (4), (7) and (8) are multiples of equations (1), (2), (5) and (6) and the values of b_2 and e_2 are given by the ratio of the entries along the 2-mode of the tensor.

The fact that $b_2 = e_2$ turns the problem above into a matrix problem that we can write as,

$$\begin{pmatrix} 1 \\ a_2 \end{pmatrix} \circ \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} 1 \\ d_2 \end{pmatrix} \circ \begin{pmatrix} f_1 \\ f_2 \end{pmatrix} = \begin{bmatrix} t_{111} & t_{112} \\ t_{211} & t_{212} \end{bmatrix} \tag{2.25}$$

Thus we deduce that there exist infinitely many possible values for the entries of the vectors.

Hence we find that the decomposition of a given tensor in not unique when $b_2 = e_2$ and the above equations hold.

Similarly, we can deduce that the decomposition of the tensor will not be unique when we have $a_2 = d_2$ or $\frac{c_1}{c_2} = \frac{f_1}{f_2}$.

Solving the equations when the entries of the component vectors are different

We will now solve equation 2.24 for the case when $b_2 \neq e_2$

From equations (1) and (2) we can obtain the following expressions,

$$f_1 = t_{111} - c_1 \text{ and } f_2 = t_{112} - c_2$$

which we can substitute in the remaining six equations to obtain a new system of equations in six variables.

$$\begin{aligned} b_2c_1 + e_2(t_{111} - c_1) &= t_{121} \quad (3)\\ b_2c_2 + e_2(t_{112} - c_2) &= t_{122} \quad (4)\\ a_2c_1 + d_2(t_{111} - c_1) &= t_{211} \quad (5)\\ a_2c_2 + d_2(t_{112} - c_2) &= t_{212} \quad (6)\\ a_2b_2c_1 + d_2e_2(t_{111} - c_1) &= t_{221} \quad (7)\\ a_2b_2c_2 + d_2e_2(t_{112} - c_2) &= t_{222} \quad (8) \end{aligned}$$

From equation (3) we can obtain the following expression,

$$c_1 = \frac{t_{121} - e_2t_{111}}{b_2 - e_2} \tag{2.26}$$

Similarly, from equation (4) we can obtain the following expression,

$$c_2 = \frac{t_{122} - e_2 t_{112}}{b_2 - e_2} \tag{2.27}$$

Thus we can see that the equations above will only be valid if $b_2 \neq e_2$.

Substituting the expressions for c_1 and c_2 in equations (5), (6), (7) and (8) we obtain the following equations,

$$\begin{aligned} a_2 \tfrac{t_{121} - e_2 t_{111}}{b_2 - e_2} + d_2 \left(t_{111} - \tfrac{t_{121} - e_2 t_{111}}{b_2 - e_2} \right) &= t_{211} \quad (5) \\ a_2 \tfrac{t_{122} - e_2 t_{112}}{b_2 - e_2} + d_2 \left((t_{112} - \tfrac{t_{122} - e_2 t_{112}}{b_2 - e_2} \right) &= t_{212} \quad (6) \\ a_2 b_2 \tfrac{t_{121} - e_2 t_{111}}{b_2 - e_2} + d_2 e_2 \left(t_{111} - \tfrac{t_{121} - e_2 t_{111}}{b_2 - e_2} \right) &= t_{221} \quad (7) \\ a_2 b_2 \tfrac{t_{122} - e_2 t_{112}}{b_2 - e_2} + d_2 e_2 \left((t_{112} - \tfrac{t_{122} - e_2 t_{112}}{b_2 - e_2} \right) &= t_{222} \quad (8) \end{aligned}$$

multiplying equation (5) by e_2 and substracting from equation (7) we obtain the following expression for a_2,

$$a_2 = \frac{t_{221} - e_2 t_{211}}{t_{121} - e_2 t_{111}}$$

which holds for $t_{121} - e_2 t_{111} \neq 0$.

Multiplying equation (6) by e_2 and substracting from equation (8) above and substituting the expression for a_2 in the result we obtain

$$e_2^2(t_{211}t_{112} - t_{111}t_{212}) + e_2(t_{212}t_{121} - t_{211}t_{122} + t_{111}t_{222} - t_{221}t_{112}) - t_{121}t_{222} + t_{221}t_{122} = 0 \tag{2.28}$$

with solutions,

$$e_2 = \frac{-(t_{212}t_{121} - t_{211}t_{122} + t_{111}t_{222} - t_{221}t_{112}) \pm \sqrt{\Delta}}{2\,(t_{211}t_{112} - t_{111}t_{212})} \quad \text{for} \quad t_{211}t_{112} - t_{111}t_{212} \neq 0$$

where the discriminant of the second-order equation is given by,

$$\Delta = (t_{212}t_{121} - t_{211}t_{122} + t_{111}t_{222} - t_{221}t_{112})^2 - 4(t_{211}t_{112} - t_{111}t_{212})(t_{121}t_{222} - t_{221}t_{122})$$

which is in fact the same expression as the hyperdeterminant for the $2 \times 2 \times 2$ tensor $\mathcal{T}$ as defined in equation 2.22.

Solving equation 2.24 in a similar way as shown above, we find the equation

$$b_2^2(t_{211}t_{112} - t_{111}t_{212}) + b_2(t_{212}t_{121} - t_{211}t_{122} + t_{111}t_{222} - t_{221}t_{112}) - t_{121}t_{222} + t_{221}t_{122} = 0$$

which has the same roots as the equation 2.28 above.

Thus we deduce that one root of the second-order equation 2.28 corresponds to the entry b_2 of the corresponding component vector while the other root of the equation corresponds to the entry e_2. We can see that is does not make any difference which root we consider first since it would only be a trivial change of order between both rank 1 components.

Hence we can deduce that both roots, although different, provide the same decomposition of the tensor.

This second-order equation provides useful information about the rank of the tensor.

- If the equation has distinct real roots, then the tensor will have rank 2.
- If the roots are complex, then we can deduce that these analytic equations above will compute the complex PARAFAC decomposition of the tensor and thus we can deduce that the tensor has rank 3.
- If the equation has equal roots, then we have a contradiction. Note that we have solved equation 2.24 setting the conditions $b_2 \neq e_2$ and $t_{211}t_{112} - t_{111}t_{212} \neq 0$. Nevertheless, the second-order equation 2.28 has equal roots for the case when $\Delta = 0$. Thus we can prove by contradiction that we cannot write a rank 3 tensor for which its hyperdeterminant is zero as the sum of two rank 1 tensors.

We can see that these results explain the importance of the hyperdeterminant in the inner structure of the tensor and clarify the use of the Kruskal polynomial or hyperdeterminant to compute the rank of a tensor.

Now we want to find explicit expressions for the rest of the entries of the vectors of the decomposition.

Equating the expressions obtained for $(a_2 - d_2)$ in equations (5) and (6), we obtain the expression,

$$\frac{t_{211} - d_2 t_{111}}{c_1} = \frac{t_{212} - d_2 t_{112}}{c_2} \tag{2.29}$$

Substituting the expressions for c_1 and c_2 given in 2.26 and 2.27 in this equation above 2.29 we obtain,

$$\frac{t_{211} - d_2 t_{111}}{t_{121} - e_2 t_{111}} = \frac{t_{212} - d_2 t_{112}}{t_{122} - e_2 t_{112}}$$

which we can solve for d_2 as

$$d_2 = \frac{(t_{212}t_{121} - t_{211}t_{122}) - e_2(t_{111}t_{212} - t_{211}t_{112})}{t_{112}t_{121} - t_{111}t_{122}}$$

Thus we obtain,

$$e_2 = \frac{-(t_{212}t_{121}-t_{211}t_{122}+t_{111}t_{222}-t_{221}t_{112})+\sqrt{\Delta}}{2(t_{211}t_{112}-t_{111}t_{212})}$$

$$b_2 = \frac{-(t_{212}t_{121}-t_{211}t_{122}+t_{111}t_{222}-t_{221}t_{112})-\sqrt{\Delta}}{2(t_{211}t_{112}-t_{111}t_{212})}$$

$$a_2 = \frac{t_{221}-e_2t_{211}}{t_{121}-e_2t_{111}}$$

$$d_2 = \frac{(t_{212}t_{121}-t_{211}t_{122})-e_2(t_{111}t_{212}-t_{211}t_{112})}{t_{112}t_{121}-t_{111}t_{122}}$$

$$c_1 = \frac{t_{121}-e_2t_{111}}{b-e}$$

$$c_2 = \frac{t_{122}-e_2t_{112}}{b-e}$$

$$f_1 = t_{111} - c_1$$

$$f_2 = t_{112} - c_2$$

We can use these equations to compute an analytical decomposition of a tensor as a sum of two rank 1 tensors.

MATLAB 2.4

The following MATLAB code will help us compute the decomposition of different tensors. The entries of the tensor are labeled `A,B,C,D,E,F,G,H` as usual.

MATLAB 2.4 Computing the Rank 2 PARAFAC Decomposition

```
A=1; B=0; C=1; D=-1; E=-1; F=1; G=1; H=0;
T_1 = [A,B,E,F;C,D,G,H];
T = tensor(T_1,[2,2,2])
kruskalpol = [(C*E-A*G) (G*B-C*F+A*H-D*E) (-B*H+D*F)]
r = roots(kruskalpol)
e = r(1)
b = r(2)
d = ((G*B-C*F)-(e)*(A*G-C*E))/(E*B-A*F)
a = (D-e*C)/(B-A*e)   %   equation 3 minus e times equation 1
c_1 = (B-e*A)/(b-e)
c_2 = (F-e*E)/(b-e)
f_1 = A-(c_1)
f_2 = E-(c_2)
```

Note that using this MATLAB code, we can compute the decomposition of a tensor without knowing before hand its actual rank.

If we obtain a real decomposition, we can deduce that the tensor has rank 2.

If we obtain a complex decomposition, we will deduce that the actual rank of the tensor considered is 3.

Tensors revisited

We are going to discuss several tensors in the light of the analytic equations we have found above relating the entries of the tensors to their PARAFAC decomposition as the sum of two rank 1 tensors.

Example 2.9

Let us consider the rank 3 tensor proposed in Kolda [2]

$$\mathcal{T} = \begin{array}{cccc} & & 0 & 1 \\ & & -1 & 0 \\ 1 & 0 & & \\ 0 & 1 & & \end{array}$$

In example 2.7 we computed the hyperdeterminant of this tensor as

$$\Delta(\mathcal{T}) = -4$$

and we can deduce that the tensor $\mathcal{T}$ has rank 3.

Solving equation 2.28 for the values given in the entries of this tensor we find

$$e_2^2 + 1 = 0$$

with solutions $e_2 = \pm i$.

The solutions are complex, as expected for a rank 3 tensor.

Thus, we can deduce that the equations above will compute the complex PARAFAC decomposition of this rank 3 tensor as a sum of two complex rank 1 tensors.

Running through the MATLAB code 2.4 to find the decomposition of $\mathcal{T}$, we find

- $e_2 = i,\ d_2 = -i,\ a_2 = i,\ b_2 = -i,\ c_1 = \frac{1}{2},\ c_2 = \frac{i}{2},\ f_1 = \frac{1}{2},\ f_2 = -\frac{i}{2}$ if e_2 takes the values of the first root.

- $e_2 = -i,\ d_2 = i,\ a_2 = -i,\ b_2 = i,\ c_1 = \frac{1}{2},\ c_2 = -\frac{i}{2},\ f_1 = \frac{1}{2},\ f_2 = \frac{i}{2}$ if e_2 takes the values of the second root.

Thus we can write the complex decomposition for $\mathcal{T}$ as

$$\mathcal{T} = \begin{pmatrix} 1 \\ i \end{pmatrix} \circ \begin{pmatrix} 1 \\ -i \end{pmatrix} \circ \begin{pmatrix} 1/2 \\ i/2 \end{pmatrix} + \begin{pmatrix} 1 \\ -i \end{pmatrix} \circ \begin{pmatrix} 1 \\ i \end{pmatrix} \circ \begin{pmatrix} 1/2 \\ -i/2 \end{pmatrix}$$

We can see that both choices provide the same decomposition, so that this decomposition is essentially unique.

□

Example 2.10

Now we are going to consider the rank 2 tensor proposed in Kruskal [21].

$$\mathcal{T} = \begin{array}{cccc} & & 0 & 1 \\ & & 1 & 0 \\ 1 & 0 & & \\ 0 & 1 & & \end{array}$$

In example 2.6 we computed the hyperdeterminant of the tensor to be

$$\Delta(\mathcal{T}) = 4$$

and we can deduce that the tensor $\mathcal{T}$ has rank 2. Indeed, solving equation 2.28 for the values given in the entries of $\mathcal{T}$ we find

$$e_2^2 - 1 = 0$$

with real solutions $e_2 = \pm 1$, as expected for a rank 2 tensor. Thus, we can deduce that the method above will compute the PARAFAC decomposition of this tensor as a sum of two rank 1 tensors.

Running through the equations to find the decomposition of $\mathcal{T}$, we find,

- $e_2 = 1,\ d_2 = 1,\ a_2 = -1,\ b_2 = -1,\ c_1 = \frac{1}{2},\ c_2 = -\frac{1}{2},\ f_1 = \frac{1}{2},\ f_2 = \frac{1}{2}$
  If e_2 takes the value of the first root.

- $e_2 = -1,\ d_2 = -1,\ a_2 = 1,\ b_2 = 1,\ c_1 = \frac{1}{2},\ c_2 = \frac{1}{2},\ f_1 = \frac{1}{2},\ f_2 = -\frac{1}{2}$
  If e_2 takes the value of the second root.

Thus we can write the PARAFAC decomposition for $\mathcal{T}$ as,

$$\mathcal{T} = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} 1/2 \\ -1/2 \end{pmatrix}$$

Note that the two sets of values for the entries of the rank 1 tensors do not produce different decompositions and we deduce that the decomposition is essentially unique.

$\square$

In the following example we give the decomposition of a tensor for which equation 2.28 does not give any relevant result so that we must solve equation 2.25 to compute the decomposition of the tensor . We find important differences between the decomposition of this tensor and the decompositions of the tensors in the two previous examples. We shall see that we have to assume arbitrary values to compute one result so that the decomposition provided is not unique.

Example 2.11

Let us consider the tensor given by

$$\mathcal{T} = \begin{array}{cccc} & & 10 & 30 \\ & & 8 & 24 \\ 7 & 21 & & \\ 6 & 18 & & \end{array}$$

In example 2.5 we computed the hyperdeterminant of this tensor as

$$\Delta(\mathcal{T}) = 0$$

Hence we deduce that equation 2.28 will have two equal roots. In fact, substituting the values for the entries of the tensor in equation 2.28 we obtain

$$4e_2^2 - 24e_2 + 36 = 0$$

with solution $e_2 = 3$.

Thus we can deduce that $b_2 = e_2 = 3$ so that we cannot use the MATLAB code 2.4 to compute the decomposition. Indeed we cannot use back substitution to compute the decomposition of the tensor and we must solve equation 2.25 instead.

Note that the ratio between the entries of the tensor along the 2-mode is constant and is equal to 3.

Substituting the values for the entries of the tensor in equation 2.25 we can write

$$\begin{aligned} c_1 + f_1 &= 7 \quad (1) \\ c_2 + f_2 &= 10 \quad (2) \\ a_2 c_1 + d_2 f_1 &= 6 \quad (3) \\ a_2 c_2 + d_2 f_2 &= 8 \quad (4) \end{aligned}$$

We can consider different combinations of values for the variables that will fix the equations. We are going to consider two different arbitrary choices for the above variables.

- Take $c_1 = 1,\ f_1 = 6$

  Equation (1) above holds for values for c_1 and f_1. Substituting these values in equation (3) above, we find

  $$a_2 + 6d_2 = 6$$

  which also holds for infinite linear combinations of the two variables.

Let us take $a_2 = -6$ and $d_2 = 2$.

Then, we can write equations (2) and (4) above as

$$\begin{aligned} c_2 + f_2 &= 10 \\ -6c_2 + 2f_2 &= 8 \end{aligned}$$

with solutions given by $c_2 = \frac{3}{2}$ and $f_2 = \frac{17}{2}$

Thus, we can write the decomposition of $\mathcal{T}$ as,

$$\mathcal{T} = \begin{pmatrix} 1 \\ -6 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 3 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 3/2 \end{pmatrix} + \begin{pmatrix} 1 \\ 2 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 3 \end{pmatrix} \circ \begin{pmatrix} 6 \\ 17/2 \end{pmatrix}$$

- Take $c_1 = 2,\ f_1 = 5$

  Equation (1) above holds for values for c_1 and f_1. Substituting these values in equation (3) above, we find

  $$2a_2 + 5d_2 = 6$$

  Let us take $a_2 = -2$ and $d_2 = 2$.

  Then, we can write equations (2) and (4) above as

  $$\begin{aligned} c_2 + f_2 &= 10 \\ -2c_2 + 2f_2 &= 8 \end{aligned}$$

  with solutions given by $c_2 = 3$ and $f_2 = 7$.

  Thus, we can write the decomposition of $\mathcal{T}$ as

  $$\mathcal{T} = \begin{pmatrix} 1 \\ -2 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 3 \end{pmatrix} \circ \begin{pmatrix} 2 \\ 3 \end{pmatrix} + \begin{pmatrix} 1 \\ 2 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 3 \end{pmatrix} \circ \begin{pmatrix} 5 \\ 7 \end{pmatrix}$$

□

We can deduce that if $b_2 = e_2$ and the ratio between the entries along the mode-2 is constant and equal to b_2, that is, the lateral slices are multiples of each other, then the tensor does not have an unique decomposition.

Similarly, we can deduce that if $a_2 = d_2$ and the ratio between the entries along the mode-1 is constant and equal to a_2, that is, the lateral slices are multiples of each other, then the tensor does not have an unique decomposition either. In this case, the second-order equation 2.28 becomes zero and does not give any result.

In this case, we can rewrite equation 2.24 as

$$\begin{aligned} c_1 + f_1 &= t_{111} \quad (1) \\ c_2 + f_2 &= t_{112} \quad (2) \\ b_2c_1 + e_2f_1 &= t_{121} \quad (3) \\ b_2c_2 + e_2f_2 &= t_{122} \quad (4) \end{aligned}$$

Finally, we find that if $\frac{c_1}{c_2} = \frac{f_1}{f_2}$ then the frontal slices $\mathbf{T}_{::1}$ and $\mathbf{T}_{::2}$ are related by the same ratio. In this case, equation 2.28 also turns zero without giving any result. We can deduce that the tensor does not have an unique decomposition.

We can see that, in these cases, we can even spot that the given tensor is rank 2 just by considering the ratios between the entries, just as by tensors of rank 1.

Example 2.12

Let us consider the rank 3 tensor studied in previous examples 1.1, 2.3 given by

$$\mathcal{T} = \begin{array}{cccc} & & 5 & 6 \\ & & 7 & 8 \\ 1 & 2 & & \\ 3 & 4 & & \end{array}$$

In example 2.5 we computed the hyperdeterminant of this tensor to be

$$\Delta(\mathcal{T}) = 0$$

As we saw in the example, the hyperdeterminant does not provide enough information to compute the rank of the tensor, but considering the conditions set by Ten Berge [28] on the slices of the tensor we deduced in example 2.5 that the tensor has rank 3.

Now we want to compute its complex rank 2 decomposition.
Substituting the values for the entries of the tensor in equation 2.28 we obtain,

$$8e_2^2 - 16e_2 + 8 = 0$$

with solution $e_2 = 1$. We find a double root as expected when the hyperdeterminant is zero. Thus we can deduce that $b_2 = e_2 = 1$ and we must compute the decomposition using equation 2.25 as argued in example 2.11

Nevertheless, we can see that no ratio holds between the entries of the tensor along the different modes, so that we cannot compute its decomposition in the same way as in the previous example 2.11 since the corresponding equations do not hold.
Thus we deduce that this tensor does not "fit" in the 2 component model approached here and we cannot compute the corresponding complex PARAFAC decomposition.

Indeed, all rank 3 tensors for which we find that its hyperdeterminant is zero will have this special feature which will be more closely considered in chapter 4

It is important to remark that although the generic rank of 2 × 2 × 2 tensors when considered over $\mathbb{C}$ is two, the maximum rank of these tensors considered over $\mathbb{C}$ is three as is shown in Friedland [13, 14]. So that we can consider all rank 3 tensors with $\Delta(\mathcal{T}) = 0$ to have complex rank 3.

□

2.4.3 General Form of a Rank 3 Tensor

In the previous section we have computed the decomposition of a tensor as sum of two rank 1 tensors. Hence we have found the decomposition over $\mathbb{R}$ of rank 2 tensors and the complex rank 2 decomposition over $\mathbb{C}$ of real rank 3 tensors.

In this section we will compute the PARAFAC decomposition over $\mathbb{R}$ of a rank 3 tensor as the sum of three rank 1 tensors.

Let $\mathcal{T}$ be a tensor of rank 3, then we can write it as a sum of three outer products of vector triples.

We will add a third vector triple to the model given by equation 2.24 that we studied in the previous section.

We have

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} \alpha \\ \beta \end{pmatrix} + \begin{pmatrix} 1 \\ a_2 \end{pmatrix} \circ \begin{pmatrix} 1 \\ b_2 \end{pmatrix} \circ \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + \begin{pmatrix} 1 \\ d_2 \end{pmatrix} \circ \begin{pmatrix} 1 \\ e_2 \end{pmatrix} \circ \begin{pmatrix} f_1 \\ f_2 \end{pmatrix}$$

$$= \begin{array}{cccc} & & t_{112} & t_{122} \\ & & t_{212} & t_{222} \\ t_{111} & t_{121} & & \\ t_{211} & t_{221} & & \end{array} \tag{2.30}$$

Note that this model will no longer provide an unique decomposition of the tensor, since it represents a system of equations in ten variables while there are only eight entries in the tensor. Already at this stage, we can see that rank 3 tensors never present an unique decomposition over $\mathbb{R}$.

Writing the system of equations represented by equation 2.30 we obtain

$$\begin{aligned} c_1 + f_1 &= t_{111} \quad (1) \\ c_2 + f_2 &= t_{112} \quad (2) \\ b_2c_1 + e_2f_1 &= t_{121} \quad (3) \\ b_2c_2 + e_2f_2 &= t_{122} \quad (4) \\ a_2c_1 + d_2f_1 &= t_{211} \quad (5) \\ a_2c_2 + d_2f_2 &= t_{212} \quad (6) \\ \alpha + a_2b_2c_1 + d_2e_2f_1 &= t_{221} \quad (7) \\ \beta + a_2b_2c_2 + d_2e_2f_2 &= t_{222} \quad (8) \end{aligned}$$

We can see that the variables α and β only appear in the last two equations so that we can solve this system of equations in a similar way as we solved equation 2.24 in the previous section.

From equations (1) and (2) we obtain

$$f_1 = t_{111} - c_1 \text{ and } f_2 = t_{112} - c_2$$

substituting these values for f_1 and f_2 in equations (3), (4) and solving for c_1 and c_2 we obtain

$$c_1 = \frac{t_{121} - e_2 t_{111}}{b_2 - e_2} \text{ and } c_2 = \frac{t_{122} - e_2 t_{112}}{b_2 - e_2}$$

Substituting these values in equations (5) and (6) above we find

$$\begin{cases} a_2(\frac{t_{121}-e_2 t_{111}}{b_2-e_2}) + d_2(t_{111} - \frac{t_{121}-e_2 t_{111}}{b_2-e_2}) & = & t_{211} \\ a_2(\frac{t_{122}-e_2 t_{112}}{b_2-e_2}) + d_2(t_{112} - \frac{t_{122}-e t_{112}}{b_2-e_2}) & = & t_{212} \end{cases}$$

which we can rearrange as

$$\begin{cases} a_2(t_{121} - e_2 t_{111}) + d_2(b_2 t_{111} - t_{121}) & = & (b_2 - e_2) t_{211} \\ a_2(t_{122} - e t_{112}) + d_2(b_2 t_{112} - t_{122}) & = & (b_2 - e_2) t_{212} \end{cases}$$

We discussed above that we were trying to solve a system in ten unknowns so that we can assume two variables to take arbitrary values.

Let us set $b_2 = 1$ and $e_2 = -1$, then

$$\begin{cases} a_2(t_{121} + t_{111}) + d_2(t_{111} - t_{121}) & = & 2t_{211} \\ a_2(t_{122} + t_{112}) + d_2(t_{112} - t_{122}) & = & 2t_{212} \end{cases}$$

Thus we find

$$d_2 = \frac{t_{211}(t_{112} + t_{122}) - t_{212}(t_{111} + t_{121})}{t_{111} t_{122} - t_{121} t_{112}}$$

and

$$a_2 = \frac{t_{111}^2 t_{212} - t_{121}^2 t_{212} + t_{111} t_{211} t_{122} - t_{121} t_{211} t_{112} - t_{111} t_{211} t_{112} + t_{121} t_{211} t_{122}}{(t_{111} + t_{121})(t_{111} t_{122} - t_{121} t_{112})}$$

as well as

$$c_1 = \frac{t_{111} + t_{121}}{2},\ c_2 = \frac{t_{122} + t_{112}}{2} \text{ and } f_1 = \frac{t_{111} - t_{121}}{2},\ f_2 = \frac{t_{112} - t_{122}}{2}$$

Thus we can easily find the variables α and β by the values for the rest of variables in equations (7) and (8) and solving for α and β.

Note that we have assumed that the two variables b_2 and e_2 take some given value. Thus we can deduce that the equations above do not have an unique solution, which means that the tensor does not have an unique real rank 3 decomposition.

MATLAB 2.5

The following MATLAB code will help us compute a rank 3 decomposition of different tensors. The entries of the tensor are labelled `A,B,C,D,E,F,G,H` as usual. We assume that $b_2 = 1$ and $e_2 = -1$.

MATLAB 2.5 Computing the Rank 3 PARAFAC Decomposition

```
A=1; B=0; C=1; D=-1; E=-1; F=1; G=1; H=0;
T_1 = [A,B,E,F;C,D,G,H];
T = tensor(T_1,[2,2,2])
e=-1
b=1
c_1 = (A+B)/2
c_2 = (E+F)/(2)
d=(C*(E+F)-G*(A+B))/(A*F-B*E)
a=(G*A^2-G*B^2+A*C*F-B*C*E-A*C*E+B*C*F)/((A+B)*(A*F-B*E))
f_1 = A-(c_1)
f_2 = E-(c_2)
alpha= D-a.*b*c_1-d.*e.*f_1
beta= H-a.*b*c_2-d.*e.*f_2
```

Note that these results are computed only for case when $b_2 = 1$ and $e_2 = -1$. It is important to remark that different choices for these entries of the component vectors will produce different equations and results.

Tensors revisited again

We are going to study some tensors we already decomposed in the previous section. There we considered the complex rank 2 decomposition of these tensors, now we are going to compute their decomposition over $\mathbb{R}$ as sums of three rank 1 tensors using the equations above 2.30.

Example 2.13

1. Let us consider again the tensor proposed in Kolda [2]

$$\mathcal{T} = \begin{array}{cccc} & & 0 & 1 \\ & & -1 & 0 \\ 1 & 0 & & \\ 0 & 1 & & \end{array}$$

   In example 2.7 we computed the hyperdeterminant of this tensor to be $\Delta(\mathcal{T}) = 0$ and in example 2.9 we found its complex rank 2 decomposition.

Running through the MATLAB code 2.5 we find the following real decomposition

$$\mathcal{T} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 2 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} 1/2 \\ -1/2 \end{pmatrix}$$

2. Let us now consider the rank 3 tensor proposed in Lathauwer [11]

$$\mathcal{T} = \begin{array}{cccc} & & 1 & 0 \\ & & 0 & 0 \\ 0 & 1 & & \\ 1 & 0 & & \end{array}$$

Using equation 2.22 we compute its hyperdeterminant to be $\Delta(\mathcal{T}) = 0$. Since the slices are non singular, we deduce that the tensor has rank 3.

Running through the MATLAB code 2.5 we find

$$\mathcal{T} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 0 \\ -1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix}$$

3. Let us now consider the rank 3 tensor given by,

$$\mathcal{T} = \begin{array}{cccc} & & 5 & 6 \\ & & 7 & 8 \\ 1 & 2 & & \\ 3 & 4 & & \end{array}$$

In example 2.5 we computed its hyperdeterminant to be $\Delta(\mathcal{T}) = 0$ and in example 2.12 we deduced that this tensor could not be written as the sum of two real or complex rank 1 tensors.

Now we compute the rank 3 decomposition running through the MATLAB code 2.5 to find

$$\mathcal{T} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 4 \\ 4 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 3/2 \\ 11/2 \end{pmatrix} + \begin{pmatrix} 1 \\ -3 \end{pmatrix} \circ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} -1/2 \\ -1/2 \end{pmatrix}$$

□

Since we could have assumed other values for the variables b_2 and e_2, we can deduce that these decompositions represent one of the many possible ways of writing a 3 rank tensor as a sum of three rank 1 tensors.

Chapter 3

Uniqueness

Uniqueness is an important feature of the PARAFAC tensor decomposition. As we shall see, this feature is not always present and this leads to important differences between tensors.

In the previous chapters we have found the PARAFAC decomposition of several tensors.

We have found that some tensors have an unique decomposition whereas other tensors do not.

The first results about the uniqueness of tensor PARAFAC decompositions are found in the work of Harshman [17], but the most general sufficient condition for essential uniqueness is due to Kruskal [20].

Theorem 3.1 Let $\mathcal{T} = [\![\mathbf{A}, \mathbf{B}, \mathbf{C}]\!]$ where the matrices are given by $\mathbf{A} = (\mathbf{a_1}, \mathbf{a_2}, \cdots, \mathbf{a_R})$, $\mathbf{B} = (\mathbf{b_1}, \mathbf{b_2}, \cdots, \mathbf{b_R})$, $\mathbf{C} = (\mathbf{c_1}, \mathbf{c_2}, \cdots, \mathbf{c_R})$ with vectors $\mathbf{a}_i$, $\mathbf{b}_i$, and $\mathbf{c}_i$, $i = 1, \cdots, R$ as columns. then the condition,

$$k_A + k_B + k_C \geq 2R + 2$$

where k_A, k_B and k_C is the k-rank of the matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ respectively and R is the rank of $\mathcal{T}$ is sufficient for uniqueness.

A proof can be found in Stegeman and Sidiropoulos [26]. Ten Berge and Sidiropoulos [29] have shown that the *sufficient* condition set by Kruskal is also *necessary* for uniqueness of the decompositions of rank 2 or rank 3 tensors.

Note also that this condition cannot be met for rank 1 tensors. However the uniqueness of the PARAFAC decomposition in this case has already been proved by Harshman [17] and it follows that the above condition is not necessary for rank 1 tensors.

Uniqueness for Rank 1 Tensors

We found before that the PARAFAC decomposition of rank 1 tensors is given by three vectors representing the ratio between the entries of the tensor along each different mode. Since there is only one non-trivial form to represent this ratio, we can deduce that the PARAFAC decomposition of rank 1 tensors is unique.

Uniqueness for Rank 2 Tensors

When considering rank 2 decompositions, we have found several examples of tensors that can be expressed as a sum of rank 1 tensors in only one essential way, producing a unique decomposition. However, we have also found examples of rank 2 tensors that can be rewritten as a sum of two rank 1 tensors in different ways, so that their decomposition is not unique. We give several examples demonstrating this feature.

Example 3.1

Let us consider the decomposition we found in Example 2.10 for the tensor proposed in Kruskal [21].

We have,

$$\begin{array}{cccc} & & 0 & 1 \\ & & 1 & 0 \\ 1 & 0 & & \\ 0 & 1 & & \end{array} = \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} 1/2 \\ -1/2 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix}$$

with component matrices given by,

$$\mathbf{A} = \begin{pmatrix} 1 & 1 \\ -1 & 1 \end{pmatrix}, \ \mathbf{B} = \begin{pmatrix} 1 & 1 \\ -1 & 0 \end{pmatrix} \text{ and } \mathbf{C} = \begin{pmatrix} 1/2 & 1/2 \\ -1/2 & 1/2 \end{pmatrix}$$

We can see that $k_A = 2,\ k_B = 2,\ k_C = 2$ and the condition given in Theorem 3.1 is satisfied.

$\square$

Example 3.2

Let us consider the tensor given in Example 2.11.

We have found that,

$$\mathcal{T} = \begin{array}{cccc} & & 10 & 30 \\ & & 8 & 24 \\ 7 & 21 & & \\ 6 & 18 & & \end{array} = \begin{pmatrix} 1 \\ -2 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 3 \end{pmatrix} \circ \begin{pmatrix} 2 \\ 3 \end{pmatrix} + \begin{pmatrix} 1 \\ 2 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 3 \end{pmatrix} \circ \begin{pmatrix} 5 \\ 7 \end{pmatrix}$$

Thus we can write,

$$\mathcal{T} = \left[\!\!\left[\begin{pmatrix} 1 & 1 \\ -2 & 2 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 3 & 3 \end{pmatrix}, \begin{pmatrix} 2 & 5 \\ 3 & 7 \end{pmatrix} \right]\!\!\right]$$

We can see that

$$k_A = 2,\ k_B = 1,\ k_C = 2$$

We find that the sufficient condition proposed by Kruskal in Theorem 3.1 is not satisfied and we deduce that this tensor does not have an unique rank 2 decomposition.

In fact, we can compute other decompositions, for example,

$$\mathcal{T} = \left[\!\!\left[\begin{pmatrix} 1 & 1 \\ -6 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 1 \\ 3 & 3 \end{pmatrix}, \begin{pmatrix} 1 & 6 \\ 3/2 & 17/2 \end{pmatrix} \right]\!\!\right]$$

□

We can summarize these features as follows

- If the component matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ have full rank, then we have $k_A = 2,\ k_B = 2,\ k_C = 2$ and the condition given in Theorem 7.1 is satisfied. Thus we can deduce that rank 2 decompositions are unique if the components matrices have full rank.
- If either of the component matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ does not have full rank, then we have $k_A + k_B + k_C < 6$ and the condition given in Theorem 7.1 is not satisfied. Note that the component matrix will not have full rank if the corresponding component vectors are multiples of each other. This is the case when there exists some ratio between the slices of the tensor as explained in example 2.11.

Uniqueness for Rank 3 Tensors

When considering the PARAFAC decomposition of a $2 \times 2 \times 2$ tensor as the sum of three rank 1 tensors, we find that the component matrices are of size 2×3 so that we can deduce that they have a maximum rank of 2.

Hence we deduce that the maximum number which can be attained by the sum of the k-rank corresponding to each component matrix is 6 whereas Theorem 7.1 sets a bound of $2 \times 3 + 2 = 8$ for uniqueness.

Thus we deduce that rank 3 tensor decompositions are never unique over $\mathbb{R}$.

Example 3.3

Let us now consider the tensor proposed in Lathauwer [11],

$$\mathcal{T} = \begin{array}{cccc} & & 1 & 0 \\ & & 0 & 0 \\ 0 & 1 & & \\ 1 & 0 & & \end{array}$$

In [11] we find the following decomposition,

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 0 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \circ \begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 0 \end{pmatrix} \circ \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

with $\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$, $\mathbf{B} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}$ and $\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ as component matrices.

However, solving Equation 2.30 for this tensor we find,

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 0 \\ -1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} -1/2 \\ 1/2 \end{pmatrix}$$

with $\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$, $\mathbf{B} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$ and $\mathbf{C} = \begin{pmatrix} 0 & 1/2 & -1/2 \\ -1 & 1/2 & 1/2 \end{pmatrix}$ as component matrices.

With these examples we can see that rank 3 decompositions are not unique.

□

Example 3.4

Let us now consider the tensor proposed in Kolda [2]

$$\mathcal{T} = \begin{array}{cccc} & & 0 & 1 \\ & & -1 & 0 \\ 1 & 0 & & \\ 0 & 1 & & \end{array}$$

We found in example 2.13 the following decomposition for this tensor,

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 2 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1/2 \\ 1/2 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} 1/2 \\ -1/2 \end{pmatrix}$$

with $\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & -1 & 1 \end{pmatrix}$, $\mathbf{B} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 1 & -1 \end{pmatrix}$ and $\mathbf{C} = \begin{pmatrix} 2 & 1/2 & 1/2 \\ 0 & 1/2 & -1/2 \end{pmatrix}$ as component matrices.

As in the previous example, we find another decomposition for this tensor proposed in Kolda [2],

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 0 \end{pmatrix} \circ \begin{pmatrix} 1 \\ -1 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 0 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

with $\mathbf{A} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & -1 \end{pmatrix}$, $\mathbf{B} = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$ and $\mathbf{C} = \begin{pmatrix} 1 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix}$ as component matrices.

□

Chapter 4

Degeneracy

This chapter is dedicated to the study of the rank 3 tensors for which the hyperdeterminant is zero.

In the previous chapters we have encountered tensors that display special features, i.e., the hyperdeterminant of the tensor is zero, see 2.5, the numerical criterion based on approximations fails to produce any relevant information, see example 2.4 or the fact that it is impossible to write these tensors as the sum of two real or complex rank 1 tensors, see example 2.12. These features describe the behaviour of the tensors studied in this chapter. They will play an important role when considering the rank 2 approximations to the tensor.

It is important to remark that Kruskal [21] explained that for the tensors for which his polynomial attained the zero value, the rank could not be decided.

This issue is solved when Ten Berge [28] introduces the condition on the entries of the tensor that the slices should be non singular. In this way, he achieves to prove that tensors such that their hyperdeterminant is zero which have non singular slices will have rank 3.

4.1 Degeneracy Parabola

Let us consider again the hyperdeterminant of a tensor as given by equation 2.22. Since we are considering $2 \times 2 \times 2$ tensors we are dealing with 8 entries which can be considered as 8 variables. But since we would like to gain some insight in what it means that the hyperdeterminant of a tensor is zero, we are going to attempt to represent the hyperdeterminant as a function of only two variables.

Thus, let us consider equation 2.22 taking the entries t_{221} and t_{222} as variables, as explained in Paatero [23]. We can rewrite the equation for the hyperdeterminant of the tensor as follows, relabelling the entries t_{221} and t_{222} as $\mathtt{D}$ and $\mathtt{H}$ to highlight them.

$$\begin{aligned}\Delta(\mathcal{T}) \equiv D(\mathtt{D},\ \mathtt{H}) = {} & t_{111}^2\mathtt{H}^2 + t_{112}^2\mathtt{D}^2 + t_{121}^2 t_{212}^2 + t_{122}^2 t_{211}^2 \\ & -2\left(t_{111}t_{112}\mathtt{DH} + t_{111}t_{121}t_{212}\mathtt{H} + t_{111}t_{122}t_{211}\mathtt{H}\right. \\ & \left. +t_{112}t_{121}t_{212}\mathtt{D} + t_{112}t_{122}t_{211}\mathtt{D} + t_{121}t_{122}t_{212}t_{211}\right) \\ & +4\left(t_{111}t_{122}t_{212}\mathtt{D} + t_{112}t_{121}t_{211}\mathtt{H}\right) \end{aligned} \tag{4.1}$$

Rearranging this equation we obtain,

$$D(\mathtt{D},\ \mathtt{H}) = t_{112}^2\mathtt{D}^2 - 2t_{111}t_{112}\mathtt{DH} + t_{111}^2\mathtt{H}^2 + (4t_{111}t_{122}t_{212} - 2t_{112}t_{121}t_{212} - 2t_{112}t_{122}t_{211})\,\mathtt{D}$$

$$+\,(4t_{112}t_{121}t_{211} - 2t_{111}t_{121}t_{212} - 2t_{111}t_{122}t_{211})\,\mathtt{H} + t_{121}^2 t_{212}^2 + t_{122}^2 t_{211}^2 - 2t_{121}t_{122}t_{212}t_{211}$$

We can see that,

$$(-2t_{111}t_{112})^2 - 4t_{112}^2 t_{111}^2 = 0$$

Hence, we can deduce that equation $D\,(\mathtt{D},\ \mathtt{H}) = 0$ represents a parabola for all values of the entries t_{111}, t_{121}, t_{211}, t_{112}, t_{122}, t_{212} except for $t_{111} = t_{112} = 0$. We shall denote the parabola $D(\mathtt{D},\mathtt{H})$ as the degeneracy parabola for $\mathcal{T}$ [33].

- The points above this parabola represent tensors for which $\Delta(\mathcal{T}) > 0$ and thus are tensors of rank 2.
- Similarly, tensors with $\Delta(\mathcal{T}) < 0$ and therefore of rank 3 are represented below the parabola.
- The points on the parabola represent tensors with $\Delta(\mathcal{T}) = 0$.

Note that the points on the parabola represent tensors with the only condition that $\Delta(\mathcal{T}) = 0$. That is, represented on this parabola we find tensors of rank 1, 2 and 3 as seen in example 2.5.

Nevertheless, De Silva and Lim [7] have shown that the *generic* rank of the tensors such that $\Delta(\mathcal{T}) = 0$ is 3. Thus, we can consider that, generically, the tensors represented on the parabola will have rank 3. These tensors are on the boundary between the sets representing the rank 2 and rank 3 tensors. The tensors on the parabola $D(\mathtt{D},\mathtt{H})$ are defined to be **degenerate**.

Example 4.1

Let us consider the tensor studied in examples 1.1, 2.3, 2.5, 2.12 given by,

$$\mathcal{T} = \begin{array}{cccc} & & 5 & 6 \\ & & 7 & 8 \\ 1 & 2 & & \\ 3 & 4 & & \end{array}$$

In example 2.5 we computed its hyperdeterminant to be,

$$\Delta(\mathcal{T}) = 0$$

The corresponding parabola is given by,

$$D(\mathtt{D},\ \mathtt{H}) = 25\mathtt{D}^2 - 10\mathtt{DH} + \mathtt{H}^2 - 152\mathtt{D} + 56\mathtt{H} + 16$$

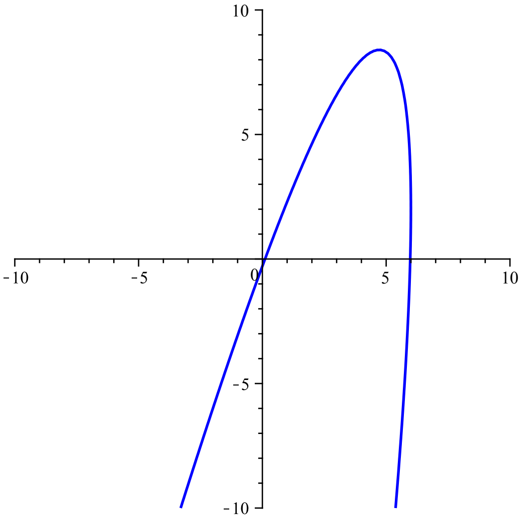

Figure 4.1: Visualization of the parabola $25\mathtt{D}^2 - 10\mathtt{DH} + \mathtt{H}^2 - 152\mathtt{D} + 56\mathtt{H} + 16 = 0$

We can visualize rank 2 tensors to be "above" the parabola while rank 3 tensors are represented "below" the parabola.

□

4.2 Defining Degeneracy

We can see that the tensors represented on the degeneracy parabola are on the boundary between the set of tensors of rank 3 with $\Delta(\mathcal{T}) < 0$ and the set of rank 2 tensors with $\Delta(\mathcal{T}) > 0$. It follows that they can be approximated arbitrarily close from the set of rank 2 tensors. In this sense, we can define these tensors to have rank 3 but *border rank* 2 as pointed out in section 2.3.2. Thus, we find that we cannot compute a best rank 2 approximation to a rank 3 tensor. That is, the minimizing cost equation given by

$$f(\mathbf{A}, \mathbf{B}, \mathbf{C}) = \left\| \mathcal{T} - \sum_{r=1}^{2} \mathbf{a_r} \circ \mathbf{b_r} \circ \mathbf{c_r} \right\|^2 \tag{4.2}$$

has no minimum.

This means that any iterative algorithm trying to find the best rank 2 approximation to a rank 3 tensor will have problems along the iteration. As the iterations approximate the tensor, the change in each step will become very small defining a behaviour denoted as *swamp* as defined by Mitchell and Burdick [22].

In these cases the PARAFAC decomposition displays a pattern of degeneracy as explained in Stegeman [25], that is defined by the following features:

- For the component matrices $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$ the columns corresponding to the degenerate factors will become nearly equal to each other or to each other's negative.
- The magnitudes of the elements of the columns in the unrestricted component matrix will become arbitrarily large.

Example 4.2

We are going to consider the computations obtained with the MATLAB Tensor Toolbox for the rank 2 approximation of the tensor.

Let

$$\mathcal{T} = \begin{array}{cccc} & & 5 & 6 \\ & & 7 & 8 \\ 1 & 2 & & \\ 3 & 4 & & \end{array}$$

In example MATLAB 2.3 we computed two rank 2 approximations to $\mathcal{T}$, each giving a different value for the minimizing function 4.2.

Let us consider the approximation given by,

$$\mathcal{B}_1 = 426.5394 \times \begin{pmatrix} 0.5279 \\ 0.8493 \end{pmatrix} \circ \begin{pmatrix} 0.6245 \\ 0.7810 \end{pmatrix} \circ \begin{pmatrix} -0.5847 \\ 0.8112 \end{pmatrix}$$

$$+419.2111 \times \begin{pmatrix} 0.5253 \\ 0.8509 \end{pmatrix} \circ \begin{pmatrix} -0.6234 \\ -0.7819 \end{pmatrix} \circ \begin{pmatrix} -0.6078 \\ 0.7941 \end{pmatrix}$$

Rewriting the decomposition for $\mathcal{B}_1$ in terms of its component matrices we find,

$$\mathcal{B}_1 = \left[\!\!\left[\begin{pmatrix} 225.17 & 220.21 \\ 362.26 & 356.71 \end{pmatrix}, \begin{pmatrix} 266.37 & -261.34 \\ 333.13 & -327.78 \end{pmatrix}, \begin{pmatrix} -249.4 & -254.80 \\ 346.01 & 332.90 \end{pmatrix} \right]\!\!\right]$$

As already highlighted in example MATLAB 2.3, the vectors in each component matrix are very similar, with differences between the corresponding entries of less than 4 %, up to sign change.

This example demonstrates the features of degenerate PARAFAC decomposition as stated above.

□

We will now consider a tensor studied in Stegeman [24] which he explains has been the only tensor for which it has been proved that the objective function has no minimum.

Example 4.3

Let us consider the tensor studied in Stegeman [24] given by

$$\mathcal{T} = \begin{array}{cccc} & & 0 & 1 \\ & & 1 & 0 \\ 1 & 0 & & \\ 0 & -1 & & \end{array}$$

We can compute its hyperdeterminant to be

$$\Delta(\mathcal{T}) = -4$$

Thus, we deduce that the tensor is of rank 3.

Running through the code given in MATLAB 2.3 we find different approximations to the tensor $\mathcal{T}$.

Let us consider the following one,

```
Final fit = 4.998490e-001
B2 is a ktensor of size 2 x 2 x 2
B2.lambda = [ 28.8224      28.8224 ]
B2.U{1} =
   -0.1250    0.1594
    0.9922   -0.9872
B2.U{2} =
    0.5345    0.5049
    0.8451    0.8632
B2.U{3} =
    0.3770    0.4089
    0.9262    0.9126
```

which we can rewrite as,

$$\mathcal{T} = 28.8224 \times \begin{pmatrix} -0.1250 \\ 0.9922 \end{pmatrix} \circ \begin{pmatrix} 0.5345 \\ 0.8451 \end{pmatrix} \circ \begin{pmatrix} 0.3770 \\ 0.9262 \end{pmatrix}$$

$$+28.8224 \times \begin{pmatrix} 0.1594 \\ -0.9872 \end{pmatrix} \circ \begin{pmatrix} 0.5049 \\ 0.8632 \end{pmatrix} \circ \begin{pmatrix} 0.4089 \\ 0.9126 \end{pmatrix}$$

$$= \left[\!\!\left[\begin{pmatrix} -3.6028 & 4.5943 \\ 28.598 & -28.453 \end{pmatrix}, \begin{pmatrix} 15.406 & 14.552 \\ 24.358 & 24.879 \end{pmatrix}, \begin{pmatrix} 10.866 & 11.785 \\ 26.695 & 26.303 \end{pmatrix} \right]\!\!\right]$$

We find the same features of degeneracy as in the example above.

The corresponding degeneracy parabola is given by,

$$D(\mathtt{D},\ \mathtt{H}) = 4\mathtt{D} + \mathtt{H}^2$$

Note that we can represent the tensor above $\mathcal{T}$ to be at the focus of the parabola.

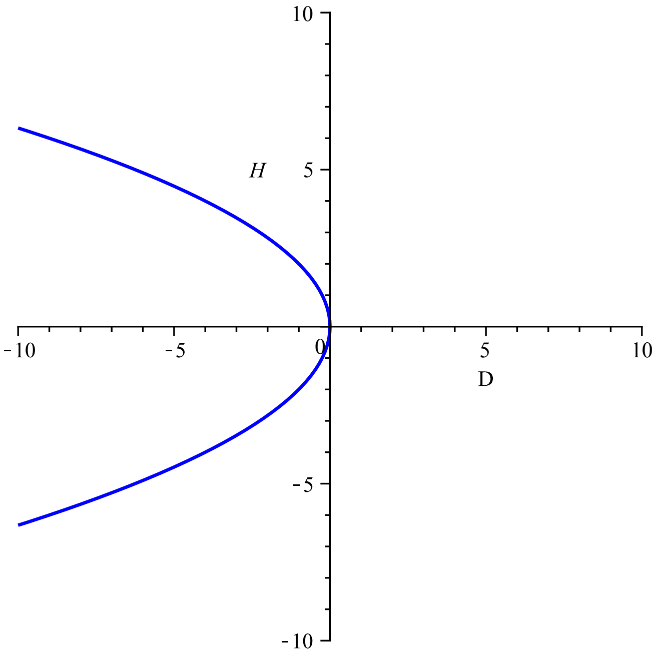

Figure 4.2: Visualization of the parabola $4\mathtt{D} + \mathtt{H}^2 = 0$

□

Example 4.4

Let us now consider the tensor proposed in Lathauwer [11],

$$\mathcal{T} = \begin{array}{cccc} & & 1 & 0 \\ & & 0 & 0 \\ 0 & 1 & & \\ 1 & 0 & & \end{array}$$

Running through the code given in MATLAB 2.3, we find,

```
Final fit = 9.723627e-001
B is a ktensor of size 2 x 2 x 2
P.lambda = [ 2.4501       2.4501 ]
B.U{1} =
    0.9768    0.9768
    0.2140   -0.2140
B.U{2} =
    0.9769   -0.9769
    0.2137    0.2137
B.U{3} =
    0.9770    0.9770
    0.2134   -0.2134
```

with equal columns in each component matrix up to sign change. In fact, we will obtain such degenerate solutions for almost every rank 2 approximation computed with MATLAB.

The corresponding degeneracy parabola is given by,

$$D(\texttt{D, H}) = \texttt{D}^2 + 4\texttt{H}$$

We can represent the tensor $\mathcal{T}$ to be at the maximum of the parabola.

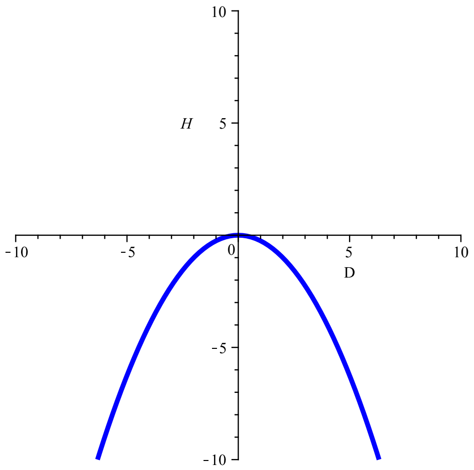

Figure 4.3: Visualization of the parabola $\texttt{D}^2 + 4\texttt{H} = 0$

□

Hyperdeterminants Revisited

Cayley defined the inner structure of 2 × 2 × 2 hypermatrices or tensors to be governed by the polynomial $\Delta(\mathcal{T})$ as defined in equation 2.22.

It is logical that we should find an explanation for degeneracy in the information provided by the study of the hyperdeterminant $\Delta(\mathcal{T})$.

We have seen that degenerate tensors are identified with the fact that they cannot be decomposed as the sum of two real of complex rank 1 tensors, that is, equation 2.23 cannot be solved.

This is given by the fact that the expression for e_2 as root of equation 2.28 must attain two different values, that is, $\Delta(\mathcal{T}) \neq 0$.

Let us now recall the expressions for the entries of the component vectors as given by the solutions of equation 2.24

$$e_2 = \frac{-(t_{212}t_{121} - t_{211}t_{122} + t_{111}t_{222} - t_{221}t_{112}) + \sqrt{\Delta}}{2\,(t_{211}t_{112} - t_{111}t_{212})}$$

$$b_2 = \frac{-(t_{212}t_{121} - t_{211}t_{122} + t_{111}t_{222} - t_{221}t_{112}) - \sqrt{\Delta}}{2\,(t_{211}t_{112} - t_{111}t_{212})}$$

$$a_2 = \frac{\sqrt{\Delta}t_{211} + t_{211}^2 t_{122} + 2t_{111}t_{212}t_{221} - t_{111}t_{211}t_{222} - t_{211}t_{112}t_{221} - t_{121}t_{211}t_{212}}{\sqrt{\Delta}t_{111} - t_{111}^2 t_{222} - 2t_{112}t_{121}t_{211} + t_{111}t_{112}t_{221} + t_{211}t_{111}t_{122} + t_{121}t_{111}t_{212}}$$

$$d_2 = \frac{t_{111}t_{222} - t_{112}t_{221} - t_{121}t_{212} + t_{211}t_{122} - \sqrt{\Delta}}{2t_{111}t_{122} - 2t_{112}t_{121}}$$

$$c_1 = \frac{t_{111}}{2} + \frac{1}{2\sqrt{\Delta}}(-t_{111}^2 t_{222} - 2t_{121}t_{211}t_{112} + t_{111}t_{112}t_{221} + t_{121}t_{212}t_{111} + t_{211}t_{122}t_{111})$$

$$c_2 = \frac{t_{112}}{2} + \frac{1}{2\sqrt{\Delta}}(t_{112}^2 t_{221} + 2t_{111}t_{122}t_{212} - t_{111}t_{112}t_{222} - t_{121}t_{212}t_{112} - t_{211}t_{122}t_{112})$$

$$f_1 = \frac{t_{111}}{2} - \frac{1}{2\sqrt{\Delta}}(-t_{111}^2 t_{222} - 2t_{121}t_{211}t_{112} + t_{111}t_{112}t_{221} + t_{121}t_{212}t_{111} + t_{211}t_{122}t_{111})$$

$$f_2 = \frac{t_{112}}{2} - \frac{1}{2\sqrt{\Delta}}(t_{112}^2 t_{221} + 2t_{111}t_{122}t_{212} - t_{111}t_{112}t_{222} - t_{121}t_{212}t_{112} - t_{211}t_{122}t_{112})$$

We want to study the behaviour of these expressions when $\Delta(\mathcal{T}) \to 0$.

If $\Delta(\mathcal{T}) \to 0$ then we find $(b_2 - e_2) \to 0$. Thus, it follows that the values for the entries c_1 and c_2 become infinitely large. Similarly, the entries f_1 and f_2 but with opposite sign to c_1 and c_2 respectively.

This behaviour of the entries of the component vectors explains the pattern of degeneracy arisen when trying to approximate rank 3 tensors.

Example 4.5

Let us consider the degeneracy parabola for $\mathcal{T} = \begin{array}{cccc} & & 5 & 6\\ & & 7 & 8\\ 1 & 2 & & \\ 3 & 4 & & \end{array}$ as given in Figure 4.3.

We find that a rank 2 tensor which is very close to $\mathcal{T}$ is given by

$$\mathcal{B}_1 = \begin{array}{cccc} & & 5 & 6\\ & & 7 & 8.001\\ 1 & 2 & & \\ 3 & 4 & & \end{array}$$

with $\|\mathcal{T} - \mathcal{B}_1\| = 10^{-6}$ and $\Delta(\mathcal{B}_1) = 0.032$.

Running through the MATLAB code 2.4 to find the decomposition of $\mathcal{B}_1$, we find,

$$\mathcal{B}_1 = \begin{pmatrix} 1 \\ 0.9775 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 0.9888 \end{pmatrix} \circ \begin{pmatrix} -44.2235 \\ -42.2346 \end{pmatrix} + \begin{pmatrix} 1 \\ 1.0222 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1.0111 \end{pmatrix} \circ \begin{pmatrix} 45.2235 \\ 47.2346 \end{pmatrix}$$

A closer rank 2 approximation to the tensor $\mathcal{T}$ is given

$$\mathcal{B}_2 = \begin{array}{cccc} & & 5 & 6\\ & & 7 & 8.000001\\ 1 & 2 & & \\ 3 & 4 & & \end{array}$$

with $\|\mathcal{T} - \mathcal{B}_2\| = 10^{-12}$ and $\Delta(\mathcal{B}_2) = 3.2 \times 10^{-5}$.

Running through the MATLAB code 2.4 to find the decomposition of $\mathcal{B}_2$, we find,

$$\mathcal{B}_2 = \begin{pmatrix} 1 \\ 0.9993 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 0.9996 \end{pmatrix} \circ \begin{pmatrix} -1413.7 \\ -1411.7 \end{pmatrix} + \begin{pmatrix} 1 \\ 1.0007 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1.0004 \end{pmatrix} \circ \begin{pmatrix} 1414.7 \\ 1416.7 \end{pmatrix}$$

An even closer rank 2 approximation to $\mathcal{T}$ is

$$\mathcal{B}_3 = \begin{array}{cccc} & & 5 & 6\\ & & 7 & 8 + 10^{-9}\\ 1 & 2 & & \\ 3 & 4 & & \end{array}$$

with $\|\mathcal{T} - \mathcal{B}_3\| = 10^{-18}$ and $\Delta(\mathcal{B}_3) = 3.2 \times 10^{-8}$.

Running through the MATLAB code 2.4 to find the decomposition of $\mathcal{B}_3$, we find,

$$\mathcal{B}_3 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} -44721 \\ -44719 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 1 \\ 1 \end{pmatrix} \circ \begin{pmatrix} 44722 \\ 44724 \end{pmatrix}$$

We can see that MATLAB no longer supports with so many decimal places so that it approximates the exact values to 5 decimal places.

This last approximation shows how dramatically can degeneracy alter the results obtained by numerical methods.

□

Chapter 5

Classification of $2 \times 2 \times 2$ Tensors

Richard Harshman liked to explain Multi-Way Factor Analysis (MFA) as one tells a story: with words, sentences, appealing for intuition, and with few formulas. Sometimes, the story turned to a tale, which required the belief of participants, because of the lack of proof of some strange - but correct - results.

These are the first words of a paper [6] written by P. Comon, X. Luciani and A. L. F de Almeida in a special issue of the Journal of Chemometrics in memory of Richard Harshman, founder of the PARAFAC model.

Following the words by Harshman, we can think of a thesis as a story or as a theatre play. First, some characters are presented, then we get to know them better, so that we can recognize their special features and behaviour. Now the story comes to an end and all the different characters presented are welcome back to the stage to bid the listener farewell.

Classifying Tensors

In this chapter we give a classification of $2 \times 2 \times 2$ tensors regarding the different features studied throughout this report.

Taking only the feature of **tensor rank** into account, we can classify tensors in 4 different sets, namely, rank 0, rank 1, rank 2 and rank 3.

In chapter 3, we saw that rank 2 tensors can have either an unique or a non-unique decompositions depending on whether there is some ratio between the slices.

When considering rank 3 tensors, we find that some have a negative value for their hyperdeterminant while other have $\Delta(\mathcal{T}) = 0$.

We can summarize these ideas in the following table showing the different tensors studied as examples in this report and displaying their special features.

Furthermore, De Silva and Lim [7] propose a classification of tensors in $\mathbb{R}^{2\times2\times2}$ in eight canonical forms. Each of the tensors given in the following classification table represents one of the canonical forms defined by De Silva and Lim [7]. We can consider each of the tensors given in the table as a representative of each of the eight equivalence classes in which $2 \times 2 \times 2$ tensors are classified.

$\mathcal{T}$	$\Delta(\mathcal{T})$	**Tensor Rank**	$k-$rank	Border Rank	Entries
$\begin{matrix} & & 0 & 0\\ & & 0 & 0\\ 0 & 0 & & \\ 0 & 0 & & \end{matrix}$	0	**0**	$k_{\mathbf{A}}=0, k_{\mathbf{B}}=0, k_{\mathbf{C}}=0$ Unique decomposition	0	All equal to zero
$\begin{matrix} & & 2 & 4\\ & & 6 & 12\\ 1 & 2 & & \\ 3 & 6 & & \end{matrix}$	0	**1**	$k_{\mathbf{A}}=1, k_{\mathbf{B}}=1, k_{\mathbf{C}}=1$ Unique decomposition	1	Ratios between all slices
$\begin{matrix} & & 10 & 30\\ & & 8 & 24\\ 7 & 21 & & \\ 6 & 18 & & \end{matrix}$	0	**2**	$k_{\mathbf{A}}=2, k_{\mathbf{B}}=1, k_{\mathbf{C}}=2$ Non unique	2	Ratio between lateral slices. $b_2=e_2$
$\begin{matrix} & & 3 & 4\\ & & 6 & 8\\ 1 & 2 & & \\ 2 & 4 & & \end{matrix}$	0	**2**	$k_{\mathbf{A}}=1, k_{\mathbf{B}}=2, k_{\mathbf{C}}=2$ Non unique	2	Ratio between horizontal slices. $a_2=d_2$
$\begin{matrix} & & 2 & 4\\ & & 6 & 8\\ 1 & 2 & & \\ 3 & 4 & & \end{matrix}$	0	**2**	$k_{\mathbf{A}}=2, k_{\mathbf{B}}=2, k_{\mathbf{C}}=1$ Non unique	2	Ratio between frontal slices. $\frac{c_1}{c_2}=\frac{f_1}{f_2}$
$\begin{matrix} & & 0 & 1\\ & & 1 & 0\\ 1 & 0 & & \\ 0 & 1 & & \end{matrix}$	+	**2**	$k_{\mathbf{A}}=2, k_{\mathbf{B}}=2, k_{\mathbf{C}}=2$ Unique decomposition	2	
$\begin{matrix} & & 5 & 6\\ & & 7 & 8\\ 1 & 2 & & \\ 3 & 4 & & \end{matrix}$	0	**3** Complex Rank: 3	$k_{\mathbf{A}}=2, k_{\mathbf{B}}=2, k_{\mathbf{C}}=2$ Non unique	2	
$\begin{matrix} & & 0 & 1\\ & & -1 & 0\\ 1 & 0 & & \\ 0 & 1 & & \end{matrix}$	$-$	**3** Complex Rank: 2	$k_{\mathbf{A}}=2, k_{\mathbf{B}}=2, k_{\mathbf{C}}=2$ Non unique	3	

Note that the three equivalence classes given by the rank 2 tensors without unique decomposition could be considered as only one equivalence class. Since these three classes arise from the fact that the slices in each of the three modes are multiples of each other, we have considered it more convenient to consider the three classes independently.

Now that we have classified $2 \times 2 \times 2$ tensors in the eight classes presented, we would like to know how often does each of the different classes occur. Kruskal undertook numerical studies which revealed that rank 2 tensors occur with probability 79% while rank 3 tensors occur with probability 21%. He considered the cases when $\Delta(\mathcal{T}) > 0$ to have rank 2 and the cases when $\Delta(\mathcal{T}) < 0$ to have rank 3, so that both rank 2 and rank 3 are typical ranks for $2 \times 2 \times 2$ tensors.

We can see that the set of the tensor with $\Delta(\mathcal{T}) = 0$ has zero volume. Nevertheless, if we consider this subset, it can be proven that these tensors have generic rank 3.

Conclusion

In this thesis we have studied the features of $2 \times 2 \times 2$ tensors and we have found methods to compute the Higher Order Singular Value Decomposition (HOSVD) and PARAFAC decomposition of these tensors. We have also studied the conditions leading to degeneracy and have related them to the inner structure of the $2 \times 2 \times 2$ tensors. Finally we have been able to classify the tensors studied in 8 different classes, namely the 8 canonical forms proposed in De Silva and Lim [7].

Further Work

Much work can be done in the computation of decompositions of larger size tensors. It is interesting to remark that in many cases the known results about rank relay on numerical studies so that much work can be done in the analytical explanation of these results.

Bibliography

[1] B. W. Bader and T. G. Kolda, *Algorithm 862: MATLAB tensor classes for fast algorithm prototyping*, ACM Trans. Math. Software, 32 (2006) pp. 635-653.

[2] B. W. Bader and T. G. Kolda *Tensor Decompositions and Applications*, SIAM Review, Vol 51. No 3. (2009) pp. 455-500.

[3] B. W. Bader and T. G. Kolda, *MATLAB Tensor Toolbox*, Version 2.2. Available at http://csmr.ca.sandia.gov/˜tgkolda/TensorToolbox/, 2007

[4] J. D. Carroll and J. J. Chang, *Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition*, Psychometrika, 45 (1980), pp. 3-24.

[5] A. Cayley *On the theory of linear transformations*, Cambridge Math. J. 4 (1845), 1-16; Collected Papers, Vol. 1, p. 80-94, Cambridge Univ. Press, 1889.

[6] P. Comon, X. Luciani and A. L. F. de Almeida, *Tensor Decompositions, Alternating Least Squares and other Tales*, Journal of Chemometrics, 23 (2009), pp. 393-405.

[7] V. De Silva and L. H. Lim *Tensor Rank and the Ill-Posedness of the Best Low-Rank Approximation Problem* SIAM J. Matrix Anal. Appl., 30 (2008), pp. 1084-1127.

[8] L. De Lathauwer, B. De Moor, and J. Vandewalle, *A multilinear singular value decomposition*, SIAM J. matrix Anal. Appl., 21 (2000), pp. 1253-1278.

[9] L. De Lathauwer, B. De Moor, and J. Vandewalle, *On the best rank-1 and rank-$(R_1, R_2, \ldots, R_N)$ approximation of higher-order tensors*, SIAM J. matrix Anal. Appl., 21 (2000), pp. 1324-1342.

[10] L. De Lathauwer and D. Nion, *Decompositions of a higher-order tensor in block terms - Part III: Alternating least squares algorithms*, SIAM J. matrix Anal. Appl., 26 (2004), pp. 295-327.

[11] L. De Lathauwer, *Signal Processing Based on Multilinear Algebra*, Ph.D. thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 1997

[12] G. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, Psychometrika, 1 (1936), pp. 211-218.

[13] S. Friedland, *On the generic rank of 3-tensors*, arXiv:0805.3777v3 [math.AG] 29 Jul 2009

[14] S. Friedland *3-Tensors: ranks and approximations*, Workshop on Algorithms for Modern Massive Data Sets, Stanford-Yahoo, June 21-24, 2006. (www.math.uic.edu/∼ friedland)

[15] I. M. Gelfand, M. M. Kapranov and A. V. Zelevinsky *Discriminants, Resultants, and Multidimensional Determinants*, Birkhäuser Boston, Boston MA, 1994

[16] R. A. Harshman, *Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis*, UCLA Working Papers in Phonetics, 16 (1970) pp. 1-84. Available at http.//publish.uwo.ca/˜harshman/wpppfac0.pdf

[17] R. A. Harshman, *Determination and proof of minimum uniqueness conditions for PARAFAC1*, UCLA Working Papers in Phonetics, 16 (1972) pp. 111-117.

[18] F. L. Hitchcock, *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys., 6 (1927), pp 164-189.

[19] F. L. Hitchcock, *Multiple invariants and generalized rank of a p-way matrix or tensor*, J. Math. Phys., 7 (1927), pp 39-79.

[20] J. B. Kruskal, *Three-way arrays: Rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics*, Linear Algebra Appl., 18 (1977), pp. 95-138.

[21] J. B. Kruskal, *Rank, decomposition, and uniqueness for 3-way and N-way arrays* in Multiway Data Analysis, R. Coppi and S. Bolasco, eds., Noth-Holland, amsterdam 1989, pp. 7-18.

[22] B. C. Mitchell and D. S. Burdick, *Slowly converging PARAFAC sequences: swamps and two-factor degeneracies.*, Journal of Chemometrics, 8 (1994), pp. 155-168.

[23] P. Paatero *Construction and analysis of degenerate PARAFAC models*, Journal of Chemometrics, 14 (2000), pp. 285-299.

[24] A. Stegeman, *Degeneracy in CANDECOMP/PARAFAC explained for $p \times p \times 2$ arrays of rank $p + 1$ or higher*, Psychometrika, 71 (2006), pp. 483-501.

[25] A. Stegeman, *Degeneracy in CANDECOMP/PARAFAC and INDSCAL explained for several three sliced arrays with a two-valued typical rank*, Psychometrika, 72 (2007), pp. 601-619.

[26] A. Stegeman and N. D. Sidiropoulos, *On Kruskal's uniqueness condition for the CANDECOMP/PARAFAC decomposition*, Linear Algebra Appl., 420 (2007), pp. 540-552.

[27] A. Stegeman, J. M. F. Ten Berge and L. De Lathauwer, *Sufficient conditions for uniqueness in CANDECOMP/PARAFAC and INDISCAL with random component matrices*, Psychometrika, 71 (2006), pp. 219-229.

[28] J. M. F. Ten Berge, *Kruskal's polynomial for* $2 \times 2 \times 2$ *arrays and a generalization to* $2 \times n \times n$ *arrays*, Psychometrika, 56 (1991), pp. 631-636.

[29] J. M. F. Ten Berge and N. D. Sidiropoulos, *On uniqueness in CANDECOMP/PARAFAC*, Psychometrika, 67 (2002), pp. 399-409.

[30] L. R. Tucker, *Implication of factor analysis of three-way matrices for measurement of change*, in Problems in Measuring Change, C. W. Harris, ed., University of Wisconsin Press, 1963, pp. 122-137.

[31] L. R. Tucker, *The extension of factor analysis to three-dimensional matrices*, in Contributions to Mathematical Psychology, H. Gulliksen and N. Frederiksen, eds., Holt, Rinehardt, & Winston, New York, 1964, pp. 110-127.

[32] L. R. Tucker, *Some mathematical notes on three-mode factor analysis*, Psychometrika, 31 (1966), pp. 279-311.

[33] B. J. H. Zijlstra and H. A. L. Kiers *Degenerate solutions obtained from several variants of factor analysis*, Chemometrics, 16 (2002), pp. 596-605.