

# A CAS AIDED SURVEY OF CP DECOMPOSITION AND RANK-1 APPROXIMATION OF A 3<sup>rd</sup>-ORDER TENSOR

Vineet Bhatt and S. Kumar

Communicated by Ayman Badawi

MSC 2010 Classifications: 15A69; 15A72; 65Y04, 15A03.

Keywords and phrases: multilinear algebra, unfolding, alternative least square, PARAFAC, CANDECOMP, CP.

The authors express their hearty thanks and gratefulness to all those scientists whose master-pieces have been consulted during the preparation of the present research article. Special thanks are due to Professors R. A. Harshman, T. G. Kolda and F.L. Hitchcock, whose fabulous research work has been entirely consulted to expose the present research article.

**Abstract.** The present manuscript is simply a survey on using computer algebraic system in computing the CANDECOMP /PARAFAC model of a 3rd-order tensor  $\mathcal{A}$ . More precisely, we have computed CP (CANDECOMP/ PARAFAC) decomposition of a 3rd-order tensor  $\mathcal{A}$  manually using ALS method as well as by employing MATLAB. Final fit of approximation has been calculated to have a good agreement between best rank one approximations of the tensor  $\mathcal{A}$  with itself. It is observed that the computed final fit (0.99991) is quite closed to 1 and hence the best one rank approximation has been declared in quite good agreement with  $\mathcal{A}$ .

## 1 Introduction

It is well known that a matrix can be decomposed in various ways depending on the type of problem that one wishes to solve. If one wants to solve a system of equations then one probably chooses 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 one wants to compute operations on a matrix or discover the inner geometrical structure of the transformation given by a matrix then one will probably prefer to work with a diagonalized version of the original matrix and will use the SVD decomposition. Thus, it can be observed that tensor decomposition will be an important issue when analyzing tensors. There are various decompositions techniques to analysis problems regarding tensor decomposition [27]. Basically there are three fundamental approaches to decompose a tensor, first one is Tucker model (multi-linear SVD or HOSVD), second one is CANDECOMP/PARAFAC and the third approach is non-negative tensor factorization. In the present paper, it is emphasized that the HOSVD is not necessarily an exact rank revealing approach [27] because of the involvement of unitary (orthogonal) tensor decomposition. Instead of the HOSVD approach, we shall seek for the decomposition of a tensor as a linear combination of a minimum number of possibly non-orthogonal rank-1 terms. This type decomposition is often called Canonical Decomposition (CANDECOMP) or Parallel Factors model (PARAFAC). This tensor decomposition was first introduced by Hitchcock [1], [2] and Eckart and Young [3]. However it was not fully developed till 1970. The enthusiastic work of Harshman [4] and Carroll and Chang [5] gave the pace to the evolution of PARAFAC/CANDECOMP models for multiway arrays. Both papers appeared in Psychometrika and revealing the same decomposition model. However, various researcher have evoked the PARAFAC/CANDECOMP in various names. The name Polyadic form of a tensor used by Hitchcock [1], PARAFAC (parallel factors) used by Harshman [4], CANDECOMP or CAND (canonical decomposition) used by Carroll and Chang [6], Topographic components model name used by Mocks [7] and recently the most commonly used word CP (CANDECOMP/PARAFAC) has been suggested by Kiers [8]. The CANDECOMP/PARAFAC is based on the fact that tensors can be rewritten as the sum of several other tensors. CP decomposition has many fields of applications. For instance, Carroll and Chang [6] introduced CANDECOMP in context of analyzing the multiple similarity or dissimilarity matrices from a variety of subjects. Harshman [4] introduced PARAFAC because it eliminates the ambiguity associated with twodimensional PCA and thus has a better uniqueness property. He was motivated by Cattell's principle of parallel proportional profiles [9]. He applied it to vowel-sound data where different individuals (mode-1) spoke different vowels (mode-2) and the formant (i.e., the pitch) was measured (mode-3). Further, Appellof and Davidson [10] pioneered the use of CP in chemometrics in 1981. Also, Andersson and Bro [11] surveyed its use in chemometrics. In particular, CP has

proved its usefulness in the modeling of fluorescence excitation-emission data. Sidiropoulos, Bro, and Giannakis [12] considered the application of CP to sensor array processing. Other applications in telecommunications include [13], [14]. Many authors have used CP decompositions in neuroscience. As mentioned previously, Mocks [7] independently discovered CP in the context of event-related potentials in brain imaging. This work also includes results on uniqueness and remarks about how to choose the number of factors of brain imaging data. Andersen and Rayens [15] applied CP-decomposition to fMRI data arranged as voxels by time by run and also as voxels by time by trial by run. Moreover, Martinez-Montes et al. [16], [17] have applied CP to a time-varying EEG spectrum arranged as a three-dimensional array with modes corresponding to time, frequency, and channel. Morup et al. [24] looked at a similar problem and then, Morup, Hansen, and Arnfred [25] have released a MATLAB toolbox called ERPWAVELAB for multichannel analysis of time-frequency transformed event-related activity of EEG and MEG data. Acar et al. [18] and De Vos et al. [19],[20] used CP for analyzing epileptic seizures. Stegeman [21] explained the differences between a three way extension of ICA and CP for multi-subject fMRI data in terms of the higher-order statistical properties. In text analysis, Bader, Berry, and Browne [22] used CP for automatic conversation detection in email over time using a term-by-author-by-time array. Shashua and Levin [23] applied CP to image compression and classification. In this paper, now we are going to calculate rank-1 approximation of a third order  $(3 \times 3 \times 3)$ -tensor. We start with few preliminaries of multilinear algebra like vector, matrix and higher order tensor products, which are useful to meet our objectives.

## 2 Vector, Matrix and Tensor products

First we define different types products of vector, matrices and third order tensors with appropriate examples. We have discussed the generalizing process of one dimensional and two-dimensional array upto multidimensional arrays. This section deals with the necessary tools to describe tensors and to work with them. This section also intended to make clear that although tensors are closely related to matrices, there are many important differences between them which make matrix analysis and tensor analysis quite different subjects, each with their own open questions and specific applications. We will also give examples on how to compute tensors with MATLAB. Allover study of this section is useful for CP decomposition. Now we are going to define step by step Methodology for array and tensor multiplication.

### 2.1 Arrays

Whenever we encounter data we must think about 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.

### 2.2 Vectors

A vector is a one-dimensional array or we can say that an array consisting of a single column or row is called a vector. In another words, a matrix with only one column is called vector, or simply a vector. The entries of a vector are called its components. The set of all columns vectors with  $n$  components is denoted by  $\mathbb{R}^n$ , and a matrix with only one row is called a row vector. We can write vector with two components  $\mathbf{a} \in \mathbb{R}^2$  as  $\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix}$  and vector with three components

$$\mathbf{a} \in \mathbb{R}^3 \text{ as } \mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}.$$

#### Vector Addition

We can add two or more vectors by adding their corresponding entries or components. Let vector  $\mathbf{a} \in \mathbb{R}^3$  and vector  $\mathbf{b} \in \mathbb{R}^3$ , then we can write their sum as

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_1 + b_1 \\ a_2 + b_2 \\ a_3 + b_3 \end{pmatrix}. \quad (2.1)$$

## Vector Products

While we can define vector addition only in one way but we find different ways of multiplying the entries of the vectors.

### Vector inner Product

Let we have two vectors are  $\mathbf{a} \in \mathbb{R}^3 = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$  and  $\mathbf{b} \in \mathbb{R}^3 = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$ , then we can write the inner product of two vectors as

$$\langle \mathbf{a}, \mathbf{b} \rangle = \left\langle \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}, \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \right\rangle = \{a_1 b_1 + a_2 b_2 + a_3 b_3\}, \quad (2.2)$$

we can see that product gives a scalar as a results.

### Vector outer product

Let we have two vectors are  $\mathbf{a} \in \mathbb{R}^3 = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$  and  $\mathbf{b} \in \mathbb{R}^3 = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$ , then we can write the outer product of two vectors

$$\mathbf{a} \circ \mathbf{b} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \circ \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} a_1 b_1 & a_1 b_2 & a_1 b_3 \\ a_2 b_1 & a_2 b_2 & a_2 b_3 \\ a_3 b_1 & a_3 b_2 & a_3 b_3 \end{pmatrix}, \quad (2.3)$$

we can see that product gives a scalar as a results.

## Norm and Normalization of vector

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.

*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. Let a vector  $\mathbf{a} \in \mathbb{R}^n$ , Euclidean norm of  $\mathbf{a}$  is defined as

$$\|\mathbf{a}\| = \sum_{(i=1)}^n a_i^2 = \sqrt{\langle \mathbf{a}, \mathbf{a} \rangle}. \quad (2.4)$$

## 2.3 Matrix

We define a matrix as a two-dimensional array of  $I$  rows and  $J$  columns. A system of  $(I \times J)$  numbers arranged a rectangular formation along  $I$  rows and  $J$  columns and bounded by the brackets  $[]$  and  $()$  is called  $I$  by  $J$  matrix.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1J} \\ a_{21} & a_{22} & \dots & a_{2J} \\ \dots & \dots & \dots & \dots \\ a_{I1} & a_{I2} & \dots & a_{IJ} \end{pmatrix}_{(I \times J)}. \quad (2.5)$$

### Adding Matrices

Proceeding in the same way we saw for vectors, we can add matrices by adding the corresponding entries of each matrix.

**Matrix Products**

As we saw vector multiplication in above subsection, we can define different ways of multiplying matrices. In this paper we use following four matrix products.

**Usual Matrix Multiplication**

Usual matrix multiplication is a well known matrix multiplication, let we have two matrix  $A = (a_{ij})_{(I \times J)}$  and  $B = (b_{jk})_{(J \times K)}$ , then their multiplication  $A \times B = C = (c_{ik})_{I \times K}$ . In this multiplication number of column in matrix  $A$  is equal to number of row in matrix  $B$ .

**Hadamard Product of two matrices**

This matrix product, first defined by the French mathematician Hadamard, is the element wise matrix product. This matrix multiplication performed for same size two matrix. Let  $A =$

$$(a_{ij})_{(I \times J)} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1J} \\ a_{21} & a_{22} & \dots & a_{2J} \\ \dots & \dots & \dots & \dots \\ a_{I1} & a_{I2} & \dots & a_{IJ} \end{pmatrix}_{(I \times J)} \quad \text{and } B = (b_{ij})_{(I \times J)} = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1J} \\ b_{21} & b_{22} & \dots & b_{2J} \\ \dots & \dots & \dots & \dots \\ b_{I1} & b_{I2} & \dots & b_{IJ} \end{pmatrix}_{(I \times J)}$$

then their Hadamard product is defined as

$$A * B = \begin{pmatrix} a_{11} b_{11} & a_{12} b_{12} & \dots & a_{1J} b_{1J} \\ a_{21} b_{21} & a_{22} b_{22} & \dots & a_{2J} b_{2J} \\ \dots & \dots & \dots & \dots \\ a_{I1} b_{I1} & a_{I2} b_{I2} & \dots & a_{IJ} b_{IJ} \end{pmatrix}_{(I \times J)}, \tag{2.6}$$

we find resultant output matrix is also same size.

**Kronecker Product of two matrices**

The Kronecker product multiplies any two matrices of any size. Let  $A = (a_{ij})_{(I \times J)}$  and  $B = (b_{kl})_{(K \times L)}$  be two matrix, then their Kronecker product is defined as

$$A \otimes B = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1J} \\ a_{21} & a_{22} & \dots & a_{2J} \\ \dots & \dots & \dots & \dots \\ a_{I1} & a_{I2} & \dots & a_{IJ} \end{pmatrix}_{(I \times J)} \otimes \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1L} \\ b_{21} & b_{22} & \dots & b_{2L} \\ \dots & \dots & \dots & \dots \\ b_{K1} & b_{K2} & \dots & b_{KL} \end{pmatrix}_{(K \times L)} =$$

$$\begin{pmatrix} a_{11} \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1L} \\ b_{21} & b_{22} & \dots & b_{2L} \\ \dots & \dots & \dots & \dots \\ b_{K1} & b_{K2} & \dots & b_{KL} \end{pmatrix} & a_{12} \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1L} \\ b_{21} & b_{22} & \dots & b_{2L} \\ \dots & \dots & \dots & \dots \\ b_{K1} & b_{K2} & \dots & b_{KL} \end{pmatrix} & \dots & a_{1J} \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1L} \\ b_{21} & b_{22} & \dots & b_{2L} \\ \dots & \dots & \dots & \dots \\ b_{K1} & b_{K2} & \dots & b_{KL} \end{pmatrix} \\ a_{21} \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1L} \\ b_{21} & b_{22} & \dots & b_{2L} \\ \dots & \dots & \dots & \dots \\ b_{K1} & b_{K2} & \dots & b_{KL} \end{pmatrix} & a_{22} \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1L} \\ b_{21} & b_{22} & \dots & b_{2L} \\ \dots & \dots & \dots & \dots \\ b_{K1} & b_{K2} & \dots & b_{KL} \end{pmatrix} & \dots & a_{2J} \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1L} \\ b_{21} & b_{22} & \dots & b_{2L} \\ \dots & \dots & \dots & \dots \\ b_{K1} & b_{K2} & \dots & b_{KL} \end{pmatrix} \\ \dots & \dots & \dots & \dots \\ a_{I1} \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1L} \\ b_{21} & b_{22} & \dots & b_{2L} \\ \dots & \dots & \dots & \dots \\ b_{K1} & b_{K2} & \dots & b_{KL} \end{pmatrix} & a_{I2} \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1L} \\ b_{21} & b_{22} & \dots & b_{2L} \\ \dots & \dots & \dots & \dots \\ b_{K1} & b_{K2} & \dots & b_{KL} \end{pmatrix} & \dots & a_{IJ} \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1L} \\ b_{21} & b_{22} & \dots & b_{2L} \\ \dots & \dots & \dots & \dots \\ b_{K1} & b_{K2} & \dots & b_{KL} \end{pmatrix} \end{pmatrix}. \tag{2.7}$$

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

**Khatri-Rao Product**

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 we have two matrix  $A = (a_{ij})_{(I \times J)}$  and  $B = (b_{kj})_{(K \times J)}$ , then their Khatri-Rao product is performed as,

$$\begin{aligned}
 A \odot B &= \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1J} \\ a_{21} & a_{22} & \dots & a_{2J} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{I1} & a_{I2} & \dots & a_{IJ} \end{pmatrix}_{(I \times J)} \odot \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1J} \\ b_{21} & b_{22} & \dots & b_{2J} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ b_{K1} & b_{K2} & \dots & b_{KJ} \end{pmatrix}_{(K \times J)} \\
 &= \begin{pmatrix} \left( \begin{matrix} a_{11} \\ a_{21} \\ \dots \\ a_{I1} \end{matrix} \right) \otimes \left( \begin{matrix} b_{11} \\ b_{21} \\ \dots \\ b_{K1} \end{matrix} \right) & \dots & \left( \begin{matrix} a_{1J} \\ a_{2J} \\ \dots \\ a_{IJ} \end{matrix} \right) \otimes \left( \begin{matrix} b_{1J} \\ b_{2J} \\ \dots \\ b_{KJ} \end{matrix} \right) \end{pmatrix} \\
 &= \begin{pmatrix} \begin{matrix} a_{11} \begin{pmatrix} b_{11} \\ b_{21} \\ \dots \\ b_{K1} \end{pmatrix} \\ a_{21} \begin{pmatrix} b_{11} \\ b_{21} \\ \dots \\ b_{K1} \end{pmatrix} \\ \dots \\ a_{31} \begin{pmatrix} b_{11} \\ b_{21} \\ \dots \\ b_{K1} \end{pmatrix} \\ \dots \\ a_{I1} \begin{pmatrix} b_{11} \\ b_{21} \\ \dots \\ b_{K1} \end{pmatrix} \end{matrix} & \begin{matrix} a_{12} \begin{pmatrix} b_{12} \\ b_{22} \\ \dots \\ b_{K2} \end{pmatrix} \\ a_{22} \begin{pmatrix} b_{12} \\ b_{22} \\ \dots \\ b_{K2} \end{pmatrix} \\ \dots \\ a_{32} \begin{pmatrix} b_{12} \\ b_{22} \\ \dots \\ b_{K2} \end{pmatrix} \\ \dots \\ a_{I2} \begin{pmatrix} b_{12} \\ b_{22} \\ \dots \\ b_{K2} \end{pmatrix} \end{matrix} & \dots & \begin{matrix} a_{1J} \begin{pmatrix} b_{1J} \\ b_{2J} \\ \dots \\ b_{KJ} \end{pmatrix} \\ a_{2J} \begin{pmatrix} b_{1J} \\ b_{2J} \\ \dots \\ b_{KJ} \end{pmatrix} \\ \dots \\ a_{3J} \begin{pmatrix} b_{1J} \\ b_{2J} \\ \dots \\ b_{KJ} \end{pmatrix} \\ \dots \\ a_{IJ} \begin{pmatrix} b_{1J} \\ b_{2J} \\ \dots \\ b_{KJ} \end{pmatrix} \end{matrix} \end{pmatrix}. \quad (2.8)
 \end{aligned}$$

Note that the Khatri-Rao product and the Kronecker product are identical when considering vectors, i.e.,  $\mathbf{a} \odot \mathbf{b} = \mathbf{a} \otimes \mathbf{b}$ . Kharti-Rao product produces an output matrix of size  $((IK) \times J)$ .

**Matrix scalar Product**

Let we have two matrix  $A = (a_{ij})_{(I \times J)}$  and  $B = (b_{ij})_{(I \times J)}$  of same size, Then we can define the scalar product

$$\begin{aligned}
 \langle A, B \rangle &= \left\langle \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1J} \\ a_{21} & a_{22} & \dots & a_{2J} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{I1} & a_{I2} & \dots & a_{IJ} \end{pmatrix}, \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1J} \\ b_{21} & b_{22} & \dots & b_{2J} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ b_{I1} & b_{I2} & \dots & b_{IJ} \end{pmatrix} \right\rangle \\
 &= \sum_{(i=1)}^I \sum_{(j=1)}^J (a_{ij}b_{ij}) = Tr(A^t B). \quad (2.9)
 \end{aligned}$$

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**

For a matrix  $A \in \mathbb{R}^{I \times J}$ , we define its Frobenius norm as

$$\| A \| = \left( \sum_{(i=1)}^I \sum_{(j=1)}^J |a_{ij}|^2 \right)^{1/2} = \sqrt{\langle A, A \rangle}. \tag{2.10}$$

We will see that the matrix norm, of the difference between two matrices  $A = (a_{ij})_{(I \times J)}$ ,  $B = (b_{ij})_{(I \times J)}$  by  $\|A - B\|_F$  will define the distance between the two matrices.

**2.4 Tensor Multiplication**

Above we have defined vectors and matrix multiplication but in case of higher order tensor multiplication it is something quite different. First we define different types of tensor multiplication same as vector and matrix multiplication.

**Tensor Inner product**

Let we have two third order tensor  $\mathcal{A}$  and  $\mathcal{D}$  such that  $\mathcal{A} = \begin{matrix} & a_5 & a_6 \\ a_1 & a_2 & a_7 & a_8 \\ a_3 & a_4 & & \end{matrix}$  and  $\mathcal{D} = \begin{matrix} & b_5 & b_6 \\ & b_7 & b_8 \\ b_1 & b_2 \\ b_3 & b_4 \end{matrix}$ .

Inner product of both tensor is defined by

$$\langle \mathcal{A}, \mathcal{D} \rangle = \{a_1b_1 + a_2b_2 + a_3b_3 + a_4b_4 + a_5b_5 + a_6b_6 + a_7b_7 + a_8b_8\}. \tag{2.11}$$

**Tensor-matrix product**

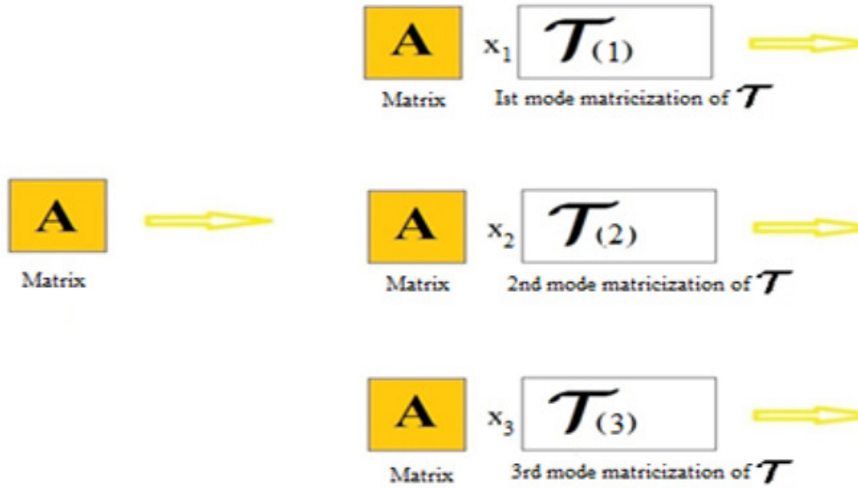
we denote the n-mode product of a tensor  $\mathcal{A}$  with a matrix  $A$  as  $\mathcal{P} = \mathcal{A} \times_n A$  where each mode-n fiber of  $\mathcal{A}$  to compute each mode-n fiber of the resulting tensor  $\mathcal{P}$  (see figure-1).

$$\mathcal{P} = \mathcal{A} \times_n A \iff \mathcal{P}_n = A\mathcal{A}_n. \tag{2.12}$$

Let we have a third order tensor  $\mathcal{A} \in \mathbb{R}^{3 \times 3 \times 3}$ , such that

$$\mathcal{A} = \begin{matrix} & & & 20 & 21 & 22 \\ & & & 23 & 24 & 25 \\ & & & 27 & 28 & 29 \\ 10 & 13 & 17 \\ 11 & 14 & 18 \\ 12 & 16 & 19 \\ 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{matrix} \tag{2.13}$$





**Figure 1.** Visualization of Tensor times Matrix along the Different Modes

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. Let us consider a tensor as in above example, and 1st mode multiplication with vector is defined as

$$P = A \times_n \mathbf{a}, \tag{2.17}$$

similarly 2nd and 3rd-mode multiplication performed. By running MATLAB code (see Appendix-1(b)) for tensor time vector, we obtain,

$$M = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 \\ 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 \end{pmatrix} \text{ and } T \text{ is a tensor of size } (3 \times 3 \times 3),$$

which is obtained from matrix  $M$ .

$$T(:, :, 1) = \begin{matrix} 1 & 2 & 3 & & & & & & \\ 10 & 11 & 12 & , T(:, :, 2) = & 13 & 14 & 15 & , T(:, :, 3) = & 16 & 17 & 18 & , \text{ and a vector} \\ 19 & 20 & 21 & & 22 & 23 & 24 & & 25 & 26 & 27 \end{matrix}$$

$$\mathbf{a} = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}.$$

$P1$  is a tensor of size  $(3 \times 3)$ , which is 1-mode tensor time vector multiplication  $P1(:, :) =$   
 168 213 258  
 183 228 273  
 198 243 288

$P2$  is a tensor of size  $(3 \times 3)$ , which is 2-mode tensor time vector multiplication  $P2(:, :) =$   
 32 77 122  
 167 212 257  
 302 347 392

$P3$  is a tensor of size  $(3 \times 3)$ , which is 3-mode tensor time vector multiplication  $P3(:, :) =$   
 66 81 96  
 201 216 231 .  
 336 351 366

**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 (2.4) and (2.10), we defined the Euclidean norm of the vector and the Frobenius matrix norm respectively.





### t of three vectors

**Figure 2.** Visualization of outer product of three vectors

Similarly, we can define the Frobenius norm of a tensor  $\mathcal{A}$  of size  $(I \times J \times K)$  by the equation

$$\|\mathcal{A}\|_F = \left( \sum_{i=1}^I \sum_{j=1}^J \sum_{k=1}^K |a_{ijk}|^2 \right)^{1/2} = \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle}, \tag{2.18}$$

where  $\langle \mathcal{A}, \mathcal{A} \rangle$  is the inner product of the tensor by itself. Let us consider the tensor  $\mathcal{A}$ , from equation (2.13), then its norm is

$$\begin{aligned} \|\mathcal{A}\|_F &= \sqrt{\langle \mathcal{A}, \mathcal{A} \rangle} = (1^2 + 2^2 + 3^2 + 4^2 + 5^2 + 6^2 + 7^2 + 8^2 + 9^2 + 10^2 + 11^2 + 12^2 \\ &+ 13^2 + 14^2 + 16^2 + 17^2 + 18^2 + 19^2 + 20^2 + 21^2 + 22^2 + 23^2 + 24^2 + 25^2 + 27^2 + 28^2 + 29^2)^{1/2} = \\ & \sqrt{6930} = 83.2466 \end{aligned}$$

### 3 CANDECOM/PARAFAC (CP Decomposition)

For CP decomposition, first we are going to define outer product of three vectors, which produces a tensor.

Outer Product of three vectors: In above section, when studying vector products, we saw that the outer product of two vectors produces a matrix. (See equation (2.3)). Taking this idea, 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,(see figure-2). Mathematical we can write the outer product of three vectors with two components  $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^2$  as

$$\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \circ \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} \circ \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} = \begin{matrix} a_1 b_1 c_1 & a_1 b_2 c_1 \\ a_2 b_1 c_1 & a_2 b_2 c_1 \end{matrix} \tag{3.1}$$

and outer product of three vectors with three components  $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^3$  as

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \circ \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \circ \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{matrix} a_1b_1c_3 & a_1b_2c_3 & a_1b_3c_3 \\ a_2b_1c_3 & a_2b_2c_3 & a_2b_3c_3 \\ a_3b_1c_3 & a_3b_2c_3 & a_3b_3c_3 \\ a_1b_1c_2 & a_1b_2c_2 & a_1b_3c_2 \\ a_2b_1c_2 & a_2b_2c_2 & a_2b_3c_2 \\ a_3b_1c_2 & a_3b_2c_2 & a_3b_3c_2 \\ a_1b_1c_1 & a_1b_2c_1 & a_1b_3c_1 \\ a_2b_1c_1 & a_2b_2c_1 & a_2b_3c_1 \\ a_3b_1c_1 & a_3b_2c_1 & a_3b_3c_1 \end{matrix} . \quad (3.2)$$

We can see resultant produced a tensor of order three by outer product of three vectors. This concept is most important for CP- Decomposition. We can rewrite the outer product of three vectors as a matricization of the resulting tensor along the different modes in the following way,

$$\mathcal{A} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \left( \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \odot \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \right)^t, \quad (3.3)$$

$$\mathcal{A} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \left( \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \odot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \right)^t, \quad (3.4)$$

$$\mathcal{A} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \left( \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \odot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \right)^t. \quad (3.5)$$

The MATLAB code in Appendix-1(c) performs the outer product of three given vectors and computes the matricization along the three modes of the resulting tensor. Running through MATLAB code (see Appendix-1(c)), we obtain the following result, Note that we use the kron command of MATLAB to compute the Khatri-Rao products given in equations (3.3), (3.4) and (3.5). As we can saw in above section the Kronecker and KhatriRao products are identical when considering vectors. Output of above MATLAB code.

$$\mathbf{a} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} 4 \\ 5 \\ 6 \end{pmatrix}, \mathbf{c} = \begin{pmatrix} 7 \\ 8 \\ 9 \end{pmatrix}$$

$$T1 = \begin{pmatrix} 28 & 35 & 42 & 32 & 40 & 48 & 36 & 45 & 54 \\ 56 & 70 & 84 & 64 & 80 & 96 & 72 & 90 & 108 \\ 84 & 105 & 126 & 96 & 120 & 144 & 108 & 135 & 162 \end{pmatrix}$$

$T$  is a tensor of size  $3 \times 3 \times 3$ , which is obtained from  $T1$ .

$$T(:, :, 1) = \begin{matrix} 28 & 35 & 42 \\ 56 & 70 & 84 \\ 84 & 105 & 126 \end{matrix}, T(:, :, 2) = \begin{matrix} 32 & 40 & 48 \\ 64 & 80 & 96 \\ 96 & 120 & 144 \end{matrix}, T(:, :, 3) = \begin{matrix} 36 & 45 & 54 \\ 72 & 90 & 108 \\ 108 & 135 & 162 \end{matrix}$$

$$T2 = \begin{pmatrix} 28 & 56 & 84 & 32 & 64 & 96 & 36 & 72 & 180 \\ 35 & 70 & 105 & 40 & 80 & 120 & 45 & 90 & 135 \\ 42 & 84 & 126 & 48 & 96 & 144 & 54 & 108 & 162 \end{pmatrix}$$

$T$  is a tensor of size  $3 \times 3 \times 3$ , which is obtained from  $T2$ .

$$T(:, :, 1) = \begin{matrix} 28 & 56 & 84 \\ 35 & 70 & 105 \\ 42 & 84 & 126 \end{matrix}, T(:, :, 2) = \begin{matrix} 32 & 64 & 96 \\ 40 & 80 & 120 \\ 48 & 96 & 144 \end{matrix}, T(:, :, 3) = \begin{matrix} 36 & 72 & 180 \\ 45 & 90 & 135 \\ 54 & 108 & 162 \end{matrix}$$

$$T3 = \begin{pmatrix} 28 & 56 & 84 & 35 & 70 & 105 & 42 & 84 & 126 \\ 32 & 64 & 96 & 40 & 80 & 120 & 48 & 96 & 144 \\ 36 & 72 & 108 & 45 & 90 & 135 & 54 & 108 & 162 \end{pmatrix}$$

$T$  is a tensor of size  $3 \times 3 \times 3$ , which is obtained from  $T3$ .

$$T(:, :, 1) = \begin{bmatrix} 28 & 56 & 84 \\ 32 & 64 & 96 \\ 36 & 72 & 108 \end{bmatrix}, T(:, :, 2) = \begin{bmatrix} 35 & 70 & 105 \\ 40 & 80 & 120 \\ 45 & 90 & 135 \end{bmatrix}, T(:, :, 3) = \begin{bmatrix} 42 & 84 & 126 \\ 48 & 96 & 144 \\ 54 & 108 & 162 \end{bmatrix}. \text{ In CP}$$

decomposition, we can write a tensor as the sum of several other tensors. We show above 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. CP-decomposition is a technique of decomposition in which we can rewrite a given tensor as a sum of several rank-1 tensors. We can also determine the rank of a tensor by CP-decomposition, if any tensor expressed a sum of two rank-1 tensor, then its rank is 2 and if tensor expressed as sum of three rank-1 tensor, then tensor rank is 3. We can write CP-decomposition mathematically as

$$\mathcal{A} = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = [[ABC]]. \tag{3.6}$$

Where  $R$  is the number of vector triples that compose tensor  $\mathcal{A}$ , when added up. Where the matrices are given by  $A = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_R)$ ,  $B = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_R)$  and  $C = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_R)$ , with vectors  $\mathbf{a}_i, \mathbf{b}_i$  and  $\mathbf{c}_i$ , as column vector, where  $i = 1, 2, \dots, R$ . Hence we can write the CP-decomposition of a rank  $R$  tensor as

$$\mathcal{A} = \mathbf{a}_1 \circ \mathbf{b}_1 \circ \mathbf{c}_1 + \mathbf{a}_2 \circ \mathbf{b}_2 \circ \mathbf{c}_2 + \dots + \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r = [[ABC]]. \tag{3.7}$$

In this section we are going to analyze third order  $(3 \times 3 \times 3)$ -tensors concentrating on problems about rank, decomposition and lower rank approximation to a given tensor.

### 3.1 Algebra for computing PARAFAC component

PARAFAC components are roughly calculated by minimization of the quadratic cost function (QCF).

$$f(A, B, C) = \|\mathcal{A} - \sum_{r=1}^R \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r\|^2. \tag{3.8}$$

When minimizing the function  $f(A, B, C)$ , we faced two different problems; (i) If  $f(A, B, C)$  become zero, then we have computed a decomposition of the tensor  $\mathcal{A}$ . (ii) We will able to computed best rank- $R$  approximation to a given tensor  $\mathcal{A}$ , if we can compute the minimum of the function above and it is distinct from zero. Equation (3.8) is most often minimized by means of the ALS algorithm in which the components are updated mode per mode. Always the components of PARAFAC decomposition of  $(3 \times 3 \times 3)$ -tensors are either vectors and  $(3 \times 3)$ -matrices depending on the rank- $R$  of the tensor being 1,2 or 3. In each case, the components matrices will be defined as,  $A = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_R)$ ,  $B = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_R)$  and  $C = (\mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_R)$ , where  $\mathbf{a}_i, \mathbf{b}_i$  and  $\mathbf{c}_i$  are vectors and  $i = 1, 2, \dots, R$  as columns. We can write QCF, given in eq. (3.8) by using equation (3.7).

$$f(A, B, C) = \|\mathcal{A} - [[AB, C]]\|^2. \tag{3.9}$$

Using ALS algorithm to solve this equation, we find that ALS fixes  $B$  and  $C$  to find  $A$ , then takes  $A$  and  $C$  to update  $B$  finally takes  $A$  and updated  $B$  to update  $C$ . The updating process is iterated until, we find some convergence criterion. Matricization of equation (3.8) with its different mode of unfolding is

$$\min_A \|\mathcal{A}_1 - A(C \odot B)^T\| \tag{3.10}$$

$$\min_A \|\mathcal{A}_2 - B(C \odot A)^T\| \tag{3.11}$$

$$\min_A \|\mathcal{A}_3 - C(B \odot A)^T\|. \tag{3.12}$$

Where  $\mathcal{A}_1, \mathcal{A}_2$  and  $\mathcal{A}_3$  are the first, second and third mode of unfolding of tensor  $\mathcal{A}$ . Solving these equations, we find that we can update each components matrix  $A, B$  and  $C$  as follows,

$$A \leftarrow \mathcal{A}_1[(C \odot B)^t]^{-1}, \tag{3.13}$$

$$B \leftarrow \mathcal{A}_2[(C \odot A)^t]^{-1}, \tag{3.14}$$

$$C \leftarrow \mathcal{A}_3[(B \odot A)^t]^{-1}. \tag{3.15}$$

Which is write as in Kolda [27].

$$A \leftarrow \mathcal{A}_1(C \odot B)(C^t C * B^t B)^{-1}, \quad (3.16)$$

$$B \leftarrow \mathcal{A}_2(C \odot A)(C^t C * A^t A)^{-1}, \quad (3.17)$$

$$C \leftarrow \mathcal{A}_3(B \odot A)(B^t B * A^t A)^{-1}. \quad (3.18)$$

### 3.2 PARAFAC Computation of a tensor by ALS method

We are going to use ALS algorithm to compute PARAFAC decomposition of a third order ( $3 \times 3 \times 3$ )-tensor, and we effort to reach some results for a general case which can disclose something about the inner structure of rank-1 tensors. For computing the PARAFAC decomposition of

rank-1 tensor, the components matrices  $A$ ,  $B$  and  $C$  will be given by vectors  $\mathbf{a} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}$ ,

$\mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$  and  $\mathbf{c} = \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}$ , so that equation (3.13), (3.14) and (3.15) used by the ALS algorithm become,

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \leftarrow \mathcal{A}_1 \left[ \left( \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \odot \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \right)^t \right]^{-1}, \quad (3.19)$$

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \leftarrow \mathcal{A}_2 \left[ \left( \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \odot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \right)^t \right]^{-1}, \quad (3.20)$$

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \leftarrow \mathcal{A}_3 \left[ \left( \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \odot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \right)^t \right]^{-1}. \quad (3.21)$$

we can rewrite equation (3.19), (3.20) and (3.21) according the equation (3.16), (3.17) and (3.18), as in [27].

$$\begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \leftarrow \mathcal{A}_1 \left( \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \odot \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \right) \left( \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}^t \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} * \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}^t \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \right)^{-1}, \quad (3.22)$$

$$\begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \leftarrow \mathcal{A}_2 \left( \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \odot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \right) \left( \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix}^t \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} * \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}^t \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \right)^{-1}, \quad (3.23)$$

$$\begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} \leftarrow \mathcal{A}_3 \left( \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \odot \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \right) \left( \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}^t \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} * \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix}^t \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} \right)^{-1}, \quad (3.24)$$

all above vectors  $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^3$ .

### 3.3 Relation between the rank one tensor and different tensor components

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

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

$$\begin{pmatrix} 1 \\ a_2 \\ a_3 \end{pmatrix} \circ \begin{pmatrix} 1 \\ b_2 \\ b_3 \end{pmatrix} \circ \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{matrix} & c_3 & b_2c_3 & b_3c_3 \\ & a_2c_3 & a_2c_3 & a_2b_3c_3 \\ & a_3c_3 & a_3b_2c_3 & a_3b_3c_3 \\ c_2 & b_2c_2 & b_3c_2 & \\ a_2c_2 & a_2b_2c_2 & a_2b_3c_2 & \\ a_3c_2 & a_3b_2c_2 & a_3b_3c_2 & \\ c_1 & b_2c_1 & b_3c_1 & \\ a_2c_1 & a_2b_2c_1 & a_2b_3c_1 & \\ a_3c_1 & a_3b_2c_1 & a_3b_3c_1 & \\ \alpha_{113} & \alpha_{123} & \alpha_{133} & \\ \alpha_{213} & \alpha_{223} & \alpha_{233} & \\ \alpha_{313} & \alpha_{323} & \alpha_{333} & \\ \alpha_{112} & \alpha_{122} & \alpha_{132} & \\ \alpha_{212} & \alpha_{222} & \alpha_{232} & \\ \alpha_{312} & \alpha_{322} & \alpha_{332} & \\ \alpha_{111} & \alpha_{121} & \alpha_{131} & \\ \alpha_{211} & \alpha_{221} & \alpha_{231} & \\ \alpha_{311} & \alpha_{321} & \alpha_{331} & \end{matrix},$$

With  $a_2, a_3, b_2, b_3, c_1, c_2, c_3 \neq 0$  and non zero entries in the tensor. According outer product above and by compares both side, we can write the equations.

$$\begin{aligned} \alpha_{111} &= c_1, \alpha_{121} = b_2c_1, \alpha_{131} = b_3c_1, \alpha_{211} = a_2c_1, \alpha_{221} = a_2b_2c_1, \alpha_{231} = a_2b_3c_1, \alpha_{311} = a_3c_1, \\ \alpha_{321} &= a_3b_2c_1, \alpha_{331} = a_3b_3c_1, \\ \alpha_{112} &= c_2, \alpha_{122} = b_2c_2, \alpha_{132} = b_3c_2, \alpha_{212} = a_2c_2, \alpha_{222} = a_2b_2c_2, \alpha_{232} = a_2b_3c_2, \alpha_{312} = a_3c_2, \\ \alpha_{322} &= a_3b_2c_2, \alpha_{332} = a_3b_3c_2, \\ \alpha_{113} &= c_3, \alpha_{123} = b_2c_3, \alpha_{133} = b_3c_3, \alpha_{213} = a_2c_3, \alpha_{223} = a_2b_2c_3, \alpha_{233} = a_2b_3c_3, \alpha_{313} = a_3c_3, \alpha_{323} = a_3b_2c_3, \\ \alpha_{333} &= a_3b_3c_3. \end{aligned}$$

By solving above equation we find all entries of the vector, such that  $a_1 = 1, b_1 = 1, c_1 = \alpha_{111}, c_2 = \alpha_{112}, c_3 = \alpha_{113}$ ,

$$\begin{aligned} a_2 &= \frac{\alpha_{213}}{\alpha_{113}} = \frac{\alpha_{211}}{\alpha_{111}} = \frac{\alpha_{212}}{\alpha_{112}} = \frac{\alpha_{221}}{\alpha_{121}} = \frac{\alpha_{231}}{\alpha_{131}} = \frac{\alpha_{222}}{\alpha_{122}} = \frac{\alpha_{232}}{\alpha_{123}} = \frac{\alpha_{223}}{\alpha_{123}} = \frac{\alpha_{233}}{\alpha_{133}}, \\ a_3 &= \frac{\alpha_{311}}{\alpha_{111}} = \frac{\alpha_{312}}{\alpha_{112}} = \frac{\alpha_{313}}{\alpha_{113}} = \frac{\alpha_{323}}{\alpha_{123}} = \frac{\alpha_{333}}{\alpha_{133}} = \frac{\alpha_{322}}{\alpha_{122}} = \frac{\alpha_{332}}{\alpha_{132}} = \frac{\alpha_{321}}{\alpha_{121}} = \frac{\alpha_{331}}{\alpha_{131}}, \\ b_2 &= \frac{\alpha_{121}}{\alpha_{111}} = \frac{\alpha_{122}}{\alpha_{112}} = \frac{\alpha_{123}}{\alpha_{113}} = \frac{\alpha_{222}}{\alpha_{122}} = \frac{\alpha_{223}}{\alpha_{123}} = \frac{\alpha_{322}}{\alpha_{122}} = \frac{\alpha_{323}}{\alpha_{123}} = \frac{\alpha_{121}}{\alpha_{111}} = \frac{\alpha_{131}}{\alpha_{111}}, \\ a_3 &= \frac{\alpha_{133}}{\alpha_{113}} = \frac{\alpha_{132}}{\alpha_{112}} = \frac{\alpha_{131}}{\alpha_{111}} = \frac{\alpha_{231}}{\alpha_{121}} = \frac{\alpha_{233}}{\alpha_{123}} = \frac{\alpha_{232}}{\alpha_{122}} = \frac{\alpha_{313}}{\alpha_{113}} = \frac{\alpha_{312}}{\alpha_{112}} = \frac{\alpha_{311}}{\alpha_{111}} = \frac{\alpha_{211}}{\alpha_{111}}. \end{aligned}$$

We can see that  $a_2, a_3, b_2, b_3, c_1, c_2$  and  $c_3$  denotes the ratio between the entries in each mode-1, mode-2 and mode-3 fiber of the tensor respectively. Let we consider that one or more entries of the vectors are zero. We can find that in these cases, the entire corresponding slice of the tensor become zero. Let us consider the case  $a_2 = 0$  and  $a_3 = 0$ .

$$\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \circ \begin{pmatrix} 1 \\ b_2 \\ b_3 \end{pmatrix} \circ \begin{pmatrix} c_1 \\ c_2 \\ c_3 \end{pmatrix} = \begin{matrix} & \alpha_{113} & \alpha_{123} & \alpha_{133} \\ & 0 & 0 & 0 \\ & 0 & 0 & 0 \\ \alpha_{112} & \alpha_{122} & \alpha_{132} & \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \\ \alpha_{111} & \alpha_{121} & \alpha_{131} & \\ 0 & 0 & 0 & \\ 0 & 0 & 0 & \end{matrix},$$

and we can see that lower horizontal slice become 0 when  $a_2 = 0$  and  $a_3 = 0$ . Hence we can write the equation above as  $c_1 = \alpha_{111}, b_2c_1 = \alpha_{121}, b_3c_1 = \alpha_{131}, c_2 = \alpha_{112}, b_2c_2 = \alpha_{122}, b_3c_2 = \alpha_{132}, c_3 = \alpha_{113}, b_2c_3 = \alpha_{123}, b_3c_3 = \alpha_{133}$ . By solving above equation, we find  $c_1 = \alpha_{111}, c_2 = \alpha_{112}, c_3 = \alpha_{113}, b_2 = \frac{\alpha_{121}}{\alpha_{111}} = \frac{\alpha_{122}}{\alpha_{112}} = \frac{\alpha_{123}}{\alpha_{113}}, b_3 = \frac{\alpha_{133}}{\alpha_{113}} = \frac{\alpha_{132}}{\alpha_{112}} = \frac{\alpha_{131}}{\alpha_{111}}$ . 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.

### Numerical example of PARAFAC decomposition of a rank-1 tensor using ALS algorithm

We are going to decompose a third order tensor by hand using ALS algorithm, which is describe above. Let a third order  $(3 \times 3 \times 3)$ -tensor

$$\mathcal{A} = \begin{pmatrix} 3 & 6 & 9 \\ 6 & 12 & 18 \\ 9 & 18 & 27 \\ 2 & 4 & 6 \\ 4 & 8 & 12 \\ 6 & 12 & 18 \\ 1 & 2 & 3 \\ 2 & 4 & 6 \\ 3 & 6 & 9 \end{pmatrix}.$$

By using ALS method, let we set the starting values of vectors  $\mathbf{b}' = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$ ,  $\mathbf{c}' = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$

and we used  $\mathcal{A}_1$  is first mode matricization of tensor  $\mathcal{A}$ . Using equation (3.22),

$$\begin{aligned} \mathbf{a} &= \mathcal{A}_1 \left( \left( \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \odot \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \right) \left( \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}^t \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} * \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}^t \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} \right)^{-1} \\ &= \begin{pmatrix} 1 & 2 & 3 & 2 & 4 & 6 & 3 & 6 & 9 \\ 2 & 4 & 6 & 4 & 8 & 12 & 6 & 12 & 18 \\ 3 & 6 & 9 & 6 & 12 & 18 & 9 & 18 & 27 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ -1 \\ 2 \\ 0 \\ -2 \\ 3 \\ 0 \\ -3 \end{pmatrix} \{14 * 2\}^{-1} \\ &= \begin{pmatrix} -28 \\ -56 \\ -84 \end{pmatrix} * \{28\}^{-1} = \begin{pmatrix} -28/28 \\ -56/28 \\ -84/28 \end{pmatrix} = \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix}. \end{aligned}$$

Now putting vectors  $\mathbf{a} = \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix}$  and  $\mathbf{c}' = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$  in equation (3.23) and we find vector  $\mathbf{b}$

and we used  $\mathcal{A}_2$  is second mode matricization of tensor  $\mathcal{A}$ . Then vector

$$\begin{aligned} \mathbf{b} &= \mathcal{A}_2 \left( \left( \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \odot \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix} \right) \left( \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}^t \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} * \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix}^t \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix} \right)^{-1} \\ &= \begin{pmatrix} 1 & 2 & 3 & 2 & 4 & 6 & 3 & 6 & 9 \\ 2 & 4 & 6 & 4 & 8 & 12 & 6 & 12 & 18 \\ 3 & 6 & 9 & 6 & 12 & 18 & 9 & 18 & 27 \end{pmatrix} \begin{pmatrix} -1 \\ -2 \\ -3 \\ -2 \\ -4 \\ -6 \\ -3 \\ -6 \\ -9 \end{pmatrix} \{14 * 14\}^{-1} \end{aligned}$$

$$= \begin{pmatrix} -196 \\ -392 \\ -588 \end{pmatrix} * \{196\}^{-1} = \begin{pmatrix} -196/196 \\ -392/196 \\ -588/196 \end{pmatrix} = \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix}.$$

Now for vector  $\mathbf{c}$ , we put the vector  $\mathbf{a}$  and  $\mathbf{b}$  in equation (3.24) and we will find vector  $\mathbf{c}$ , here we used  $\mathcal{A}_3$  is third mode matricization of tensor  $\mathcal{A}$ . Then we find

$$\begin{aligned} \mathbf{c} &= \mathcal{A}_3 \left( \left( \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix} \odot \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix} \right) \left( \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix}^t \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix} * \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix}^t \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix} \right)^{-1} \\ &= \begin{pmatrix} 1 & 2 & 3 & 2 & 4 & 6 & 3 & 6 & 9 \\ 2 & 4 & 6 & 4 & 8 & 12 & 6 & 12 & 18 \\ 3 & 6 & 9 & 6 & 12 & 18 & 9 & 18 & 27 \end{pmatrix} \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{pmatrix} \{14 * 14\}^{-1} \\ &= \begin{pmatrix} 196 \\ 392 \\ 588 \end{pmatrix} * \{196\}^{-1} = \begin{pmatrix} 196/196 \\ 392/196 \\ 588/196 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}. \end{aligned}$$

The vector  $\mathbf{a}, \mathbf{b}$  and  $\mathbf{c}$ , we obtain  $\mathbf{a} = \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix}$ ,  $\mathbf{b} = \begin{pmatrix} -1 \\ -2 \\ -3 \end{pmatrix}$  and  $\mathbf{c} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$ . Similarly, going through the above algorithm again to find  $\mathbf{a}_1, \mathbf{b}_1$  and  $\mathbf{c}_1$ , by putting the vectors  $\mathbf{a}, \mathbf{b}$  and  $\mathbf{c}$ , we obtain

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

and we stop to iterate since the values for  $\mathbf{a}, \mathbf{b}$  and  $\mathbf{c}$  converge and correct in four decimal place. Thus we can write rank-1 approximate tensor, after normalizing the vectors  $\mathbf{a}_2, \mathbf{b}_2$  and  $\mathbf{c}_2$ ,

$$\begin{aligned} \mathcal{D}_1 &= 52.3832 \begin{pmatrix} 0.2673 \\ 0.5345 \\ 0.8018 \end{pmatrix} \circ \begin{pmatrix} 0.2673 \\ 0.5345 \\ 0.8018 \end{pmatrix} \circ \begin{pmatrix} 0.2673 \\ 0.5345 \\ 0.8018 \end{pmatrix} \\ &= \begin{matrix} & & & 3.0009 & 6.0007 & 9.0017 \\ & & & 6.0007 & 11.9992 & 18.0000 \\ & & & 9.0017 & 18.0000 & 27.0016 \\ & 2.0005 & 4.0002 & 6.0007 & & \\ & 4.0002 & 7.9990 & 11.9992 & & \\ & 6.0007 & 11.9992 & 18.0000 & & \\ 1.0004 & 2.0005 & 3.0009 & & & \\ 2.0005 & 4.0002 & 6.0007 & & & \\ 3.0009 & 6.0007 & 9.0017 & & & \end{matrix}, \end{aligned}$$

### 3.4 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  $A$  of rank  $r$ . Let  $A$  be matrix of rank  $r$  and  $B$  be the matrix of rank  $k$  with  $k < r$  that is closest to  $A$ . Then

$$\|A - B\|_F = \sqrt{\sigma_{k+1}^2 + \dots + \sigma_r^2}, \tag{3.25}$$

where  $\sigma_k$  is the  $k$ -singular value of the matrix  $A$ .

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  $\mathcal{D}$  which has a lower rank than the tensor  $\mathcal{A}$  such that the expression,

$$\|\mathcal{A} - \mathcal{D}\|_F, \quad (3.26)$$

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 3.2. 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 section 3. 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  $\mathcal{D}$  to a tensor  $\mathcal{A}$ , we will be interested in knowing how good an approximation it is. If the expression given in equation (3.26) is very small, then we can deduce that the approximation is very close to the tensor  $\mathcal{A}$ . We can find the best rank-1 approximation to a given tensor by minimizing the QCF given by equation (3.8), which we can write for  $R = 1$  as,

$$f(\mathbf{a}, \mathbf{b}, \mathbf{c}) = \|\mathcal{A} - \mathbf{a} \circ \mathbf{b} \circ \mathbf{c}\|^2. \quad (3.27)$$

By minimizing this equation, we shall find the rank-1 tensor that is closest to the given tensor  $\mathcal{A}$ . We shall use the ALS algorithm, as we did for computing the PARAFAC decomposition of a rank-1 tensor mentioned above. The MATLAB code performs the computation of the best rank-1 approximation to a given tensor (See Appendix-1(d)). However, we also find rank-1 approximation of a tensor manually with an example given in the above section, for comparing and testing results. We are going to find rank-1 approximation of same tensor by MATLAB. However, we set the rank-3 tensor  $\mathcal{A}$  in MATLAB and run the tensor toolbox as shown in the script (see Appendix-1(d)); we obtain the following solution,

$$M = \begin{pmatrix} 1 & 2 & 3 & 2 & 4 & 6 & 3 & 6 & 9 \\ 2 & 4 & 6 & 4 & 8 & 12 & 6 & 12 & 18 \\ 3 & 6 & 9 & 6 & 12 & 18 & 9 & 18 & 27 \end{pmatrix}$$

$T$  is a tensor of size  $3 \times 3 \times 3$ , which is obtained from matrix  $M$ .

$$\begin{array}{ccc} 1 & 2 & 3 & & & & 2 & 4 & 6 & & & & 3 & 6 & 9 \\ T(:, :, 1) = & 2 & 4 & 6 & , & T(:, :, 2) = & 4 & 8 & 12 & , & T(:, :, 2) = & 6 & 12 & 18 \\ & 3 & 6 & 9 & & & 6 & 12 & 18 & & & & 9 & 18 & 27 \end{array}$$

Iter 1: fit = 1.000000e + 000 fitdelta = 1.0e+000

Iter 2: fit = 1.000000e + 000 fitdelta = 1.8e-008

Final fit = 1.000000e+000

B is a ktensor of size  $3 \times 3 \times 3$

B.lambda = [52.3832]

0.2673

B.U{1}= 0.5345

0.8018

0.2673

B.U{2}= 0.5345

0.8018

0.2673

B.U{3}= 0.5345 .

0.8018

We define the term **final fit** 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 (3.26) 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,

$$1 - \frac{\|\mathcal{A} - \mathcal{D}\|_F}{\|\mathcal{A}\|_F}.$$



By above MATLAB rank-1 approximation, we again obtain a tensor, which is same as  $\mathcal{D}_1$ , find by hand is

$$\mathcal{D}_1 = 52.3832 \begin{pmatrix} 0.2673 \\ 0.5345 \\ 0.8018 \end{pmatrix} \circ \begin{pmatrix} 0.2673 \\ 0.5345 \\ 0.8018 \end{pmatrix} \circ \begin{pmatrix} 0.2673 \\ 0.5345 \\ 0.8018 \end{pmatrix}$$

$$= \begin{matrix} & & & 3.0009 & 6.0007 & 9.0017 \\ & & & 6.0007 & 11.9992 & 18.0000 \\ & & & 9.0017 & 18.0000 & 27.0016 \\ & 2.0005 & 4.0002 & 6.0007 & & \\ = & 4.0002 & 7.9990 & 11.9992 & & \\ & 6.0007 & 11.9992 & 18.0000 & & \\ 1.0004 & 2.0005 & 3.0009 & & & \\ 2.0005 & 4.0002 & 6.0007 & & & \\ 3.0009 & 6.0007 & 9.0017 & & & \end{matrix},$$

where the displayed vectors are normalized. We can see that  $\mathcal{D}_1$  and  $\mathcal{D}_2$  represent two different rank-1 approximations to the tensor  $\mathcal{A}$ . Substituting the expressions found for  $\mathcal{D}_1$  and  $\mathcal{D}_2$  in equation (3.27) and using equation (3.26) to compute the Frobenius norm of the difference between both tensors we can see that,

$$\|\mathcal{A} - \mathcal{D}_1\|_F = 0.0046$$

$$\|\mathcal{A} - \mathcal{D}_2\|_F = 0.0046$$

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

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

$$1 - \frac{\|\mathcal{A} - \mathcal{D}_1\|_F \text{ Or } \|\mathcal{A} - \mathcal{D}_2\|_F}{\|\mathcal{A}\|_F} = 1 - \frac{0.0046}{52.3832} = 1 - 0.00009 = 0.99991.$$

### 3.5 Representing Rank-1 Approximations to a Tensor by three dimensional graph

We are going to represent the minimizing function  $f(\mathbf{a}, \mathbf{b}, \mathbf{c})$  as given in equation (3.27). We can find similar representations in Lathauwer, Moor and Vandewalle [28]. We are going to consider different tensors that we define as

$$\mathcal{A} = \begin{matrix} & & & C_1 & C_2 & C_3 \\ & & & C_4 & C_5 & C_6 \\ & & & C_7 & C_8 & C_9 \\ & B_1 & B_2 & B_3 & & \\ & B_4 & B_5 & B_6 & & \\ & B_7 & B_8 & B_9 & & \\ A_1 & A_2 & A_3 & & & \\ A_4 & A_5 & A_6 & & & \\ A_7 & A_8 & A_9 & & & \end{matrix} . \tag{3.28}$$

We want to represent the distance between the tensor  $\mathcal{A}$  and its rank-1 approximations in a three dimensional space so that we can visualize the behavior 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  $\mathbf{a} = \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix}, \mathbf{b} = \begin{pmatrix} \cos\alpha \\ \sin\alpha \\ \sin\alpha \end{pmatrix}$  and  $\mathbf{c} = \begin{pmatrix} \cos\beta \\ \sin\beta \\ \sin\beta \end{pmatrix}$ .

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,  $\alpha$  and  $\beta$ . Then outer product of these vector produce a tensor such that

$$\begin{aligned}
 \mathbf{a} \circ \mathbf{b} \circ \mathbf{c} &= \begin{pmatrix} 1 \\ -1 \\ -1 \end{pmatrix} \circ \begin{pmatrix} \cos\alpha \\ \sin\alpha \\ \sin\alpha \end{pmatrix} \circ \begin{pmatrix} \cos\beta \\ \sin\beta \\ \sin\beta \end{pmatrix} \\
 &= \begin{matrix} \cos\alpha\sin\beta & \sin\alpha\sin\beta & \sin\alpha\sin\beta \\ -\cos\alpha\sin\beta & -\sin\alpha\sin\beta & -\sin\alpha\sin\beta \\ -\cos\alpha\sin\beta & -\sin\alpha\sin\beta & -\sin\alpha\sin\beta \end{matrix} \\
 &= \begin{matrix} \cos\alpha\sin\beta & \sin\alpha\sin\beta & \sin\alpha\sin\beta \\ -\cos\alpha\sin\beta & -\sin\alpha\sin\beta & -\sin\alpha\sin\beta \\ -\cos\alpha\sin\beta & -\sin\alpha\sin\beta & -\sin\alpha\sin\beta \end{matrix} \quad (3.29) \\
 &= \begin{matrix} \cos\alpha\cos\beta & \sin\alpha\sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\cos\beta & -\sin\alpha\cos\beta & -\sin\alpha\cos\beta \\ -\cos\alpha\cos\beta & -\sin\alpha\cos\beta & -\sin\alpha\cos\beta \end{matrix}
 \end{aligned}$$

We find the value of  $f(\mathbf{a}, \mathbf{b}, \mathbf{c})$  according equation (3.27), by putting the value of equation (3.28) and (3.29).

$$\begin{aligned}
 f(\mathbf{a}, \mathbf{b}, \mathbf{c}) &= (A_1 - \cos\alpha\cos\beta)^2 + (A_2 - \sin\alpha\sin\beta)^2 + (A_3 - \sin\alpha\cos\beta)^2 \\
 &+ (A_4 + \cos\alpha\cos\beta)^2 + (A_5 + \sin\alpha\cos\beta)^2 + (A_6 + \sin\alpha\cos\beta)^2 \\
 &+ (A_7 + \cos\alpha\cos\beta)^2 + (A_8 + \sin\alpha\cos\beta)^2 + (A_9 + \sin\alpha\cos\beta)^2 \\
 &+ (B_1 - \cos\alpha\sin\beta)^2 + (B_2 - \sin\alpha\sin\beta)^2 + (B_3 - \sin\alpha\sin\beta)^2 \\
 &+ (B_4 + \cos\alpha\sin\beta)^2 + (B_5 + \sin\alpha\sin\beta)^2 + (B_6 + \sin\alpha\sin\beta)^2 \\
 &+ (B_7 + \cos\alpha\sin\beta)^2 + (B_8 + \sin\alpha\sin\beta)^2 + (B_9 + \sin\alpha\sin\beta)^2 \\
 &+ (C_1 - \cos\alpha\sin\beta)^2 + (C_2 - \sin\alpha\sin\beta)^2 + (C_3 - \sin\alpha\sin\beta)^2 \\
 &+ (C_4 + \cos\alpha\sin\beta)^2 + (C_5 + \sin\alpha\sin\beta)^2 + (C_6 + \sin\alpha\sin\beta)^2 \\
 &+ (C_7 + \cos\alpha\sin\beta)^2 + (C_8 + \sin\alpha\sin\beta)^2 + (C_9 + \sin\alpha\sin\beta)^2. \quad (3.30)
 \end{aligned}$$

We can sketch a graph of the minimizing function given above by running through MATLAB code (see Appendix-1(e)). For above numerical tensor, which is convert to rank-1 tensor and the final fit of the approximation is 0.99991, we have able to sketch a three dimensional graph of the rank-1 approximation. (see figure-3)

## 4 Conclusion

Here is the brief outline over the outcomes of the purposed research work: Section I is an introductory one and reveals the development of CP-decomposition and its use in various fields of sciences. In section II, we have discussed variety of vector, matrix and tensor products along with appropriate examples. Also, the MATLAB codes, corresponding to these products are also mentioned in Appendix-I. Third section with its subsequent sub-sections has been dealt with rank-1 approximation of a third order tensor  $\mathcal{A}$  using ALS method manually. We have also calculated best lower rank approximation to a 3rd order tensor  $\mathcal{A}$ . For this purpose, we have calculated rank-1 approximation of the 3rd order tensor  $\mathcal{A}$  manually which is  $\mathcal{D}_1$  and then calculated the same (which is  $\mathcal{D}_2$ ) using of MATLAB codes developed by [26] and [28]. The value for the distance between the tensor  $\mathcal{A}$  and its approximations is the same in both cases, i.e.  $\|\mathcal{A} - \mathcal{D}_1\|_F = \|\mathcal{A} - \mathcal{D}_2\|_F = 0.0046$ . Finally, we have computed the final fit of approximation which is 0.99991, and can be considered much closer to 1. This closeness shows that the best one rank approximation of a 3rd order tensor reveals quite good agreement with the original 3rd order tensor  $\mathcal{A}$ . A pictographic view of this approximation is also drawn in fig. (3).

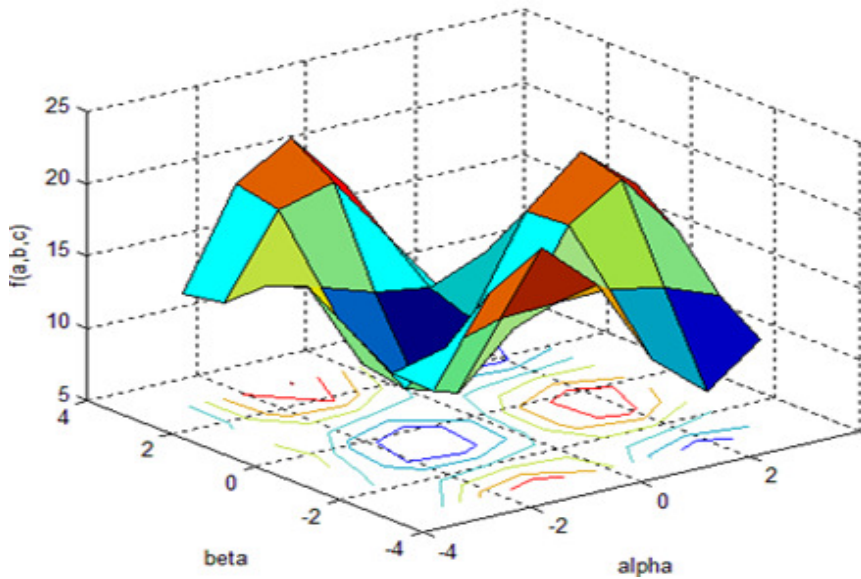


Figure 3. 3D-graph of rank-1 approximation of 3rd-order tensor

(a) Tensor times Matrix (tensor-matrix multiplication):

```
% Assume a (3 by 9)-matrix, which is used to make a tensor.
M = [1 2 3 4 5 6 7 8 9;10 11 12 13 14 15 16 17 18; 19 20 21 22 23 24 25 26 27];
T = tensor(M, [3,3,3]) %A third order (3 by 3 by 3)-tensor.
A = [1 2 3;4 5 6;7 8 9]%We assume a matrix A.
P1 = ttm(T,A,1) %Tensor Ist mode product with matrix A.
P2 = ttm(T,A,2) % Tensor IInd mode product with matrix A.
P3 = ttm(T,A,3) % Tensor IIIrd mode product with matrix A.
```

(b) Tensor times Vector (tensor-vector multiplication):

```
%Assume a (3 by 9)-matrix, which is used to make a tensor.
M = [1 2 3 4 5 6 7 8 9;10 11 12 13 14 15 16 17 18; 19 20 21 22 23 24 25 26 27];
T = tensor(M, [3,3,3]) % A third order (3 by 3 by 3)-tensor.
a = [4;5;6]% we assume a vector a.
P1 = ttv(T,a,1)% Vector multiplication with Ist mode tensor unfolding.
P2 = ttv(T,a,2)% Vector multiplication with IInd mode tensor unfolding.
P3 = ttv(T,a,3)% Vector multiplication with IIIrd mode tensor unfolding.
```

(c) Outer Product of three vectors:

```
a=[1;2;3],b=[4;5;6],c=[7;8;9]%Let we have three vector a,b and c.
T1 = a*(kron(c,b))'% T1 is first way outer product of three vectors a,b and c.
T = tensor(T1, [3,3,3])% A third order tensor produced by T1.
T2 = b*(kron(c,a))'%T2 is second way outer product of three vectors a,b and c.
T = tensor(T2, [3,3,3])% A third order tensor produced by T2.
T3 = c*(kron(b,a))'%T3 is third way outer product of three vectors a,b and c.
T = tensor(T3, [3,3,3])% A third order tensor produced by T3.
```

(d) Best Rank-1 Approximation (Parafac decomposition):

```
M = [1 2 3 4 5 6 7 8 9;10 11 12 13 14 15 16 17 18; 19 20 21 22 23 24 25 26 27];
%this matrix denotes the 1-mode matricization of the given tensor
T = tensor(M, [3,3,3]) %This code builds up the tensor from matrix M.
B = parafac_als(T,1) % This code computes the approximation.
```

(e) MATLAB code for the Minimizing Function Graph:

```
% First take all the element of the same tensor, which solved by hand in section 3.
A1=1; A2=2; A3=3; A4=2; A5=4; A6=6; A7=3; A8=6; A9=9;
B1=2; B2=4; B3=6; B4=4; B5=8; B6=12; B7=6; B8=12; B9=18;
C1=3; C2=6; C3=9; C4=6; C5=12; C6=18; C7=9; C8=18; C9=27;
% Produce a matrix by above elements.
T1 = [A1,A2,A3,A4,A5,A6,A7,A8,A9; B1,B2,B3,B4,B5,B6,B7,B8,B9; C1,C2,C3,C4,C5,C6,C7,C8,C9];
```

```

T = tensor(T1,[3,3,3]) %Produce a third order tensor by T1.
[x,y] = meshgrid([-pi:0.99991:pi]);
%Take meshgrid for final fit of approximation and then construct minimizing function f.
f =(A1 - ((cos(x)).*(cos(y)))).^2 + (A2 - ((sin(x)).*(sin(y)))).^2+
(A3 + ((sin(x)).*(cos(y)))).^2 + (A4 + ((cos(x)).*(cos(y)))).^2 +
(A5 + ((sin(x)).*(cos(y)))).^2 + (A6 - ((sin(x)).*(cos(y)))).^2+
(A7 + ((cos(x)).*(cos(y)))).^2 + (A8 + ((sin(x)).*(cos(y)))).^2+
(A9 + ((sin(x)).*(cos(y)))).^2 +(B1 - ((cos(x)).*(sin(y)))).^2 +
(B2 - ((sin(x)).*(sin(y)))).^2+ (B3 + ((sin(x)).*(sin(y)))).^2 +
(B4 + ((cos(x)).*(sin(y)))).^2 + (B5 - ((sin(x)).*(sin(y)))).^2 +
(B6 - ((sin(x)).*(sin(y)))).^2 + (B7 + ((cos(x)).*(sin(y)))).^2 +
(B8 + ((sin(x)).*(sin(y)))).^2+ (B9 + ((sin(x)).*(sin(y)))).^2 +
(C1 - ((cos(x)).*(sin(y)))).^2 + (C2 - ((sin(x)).*(sin(y)))).^2+
(C3 + ((sin(x)).*(sin(y)))).^2 + (C4 + ((cos(x)).*(sin(y)))).^2 +
(C5 - ((sin(x)).*(sin(y)))).^2 + (C6 - ((sin(x)).*(sin(y)))).^2 +
(C7 + ((cos(x)).*(sin(y)))).^2 + (C8 + ((sin(x)).*(sin(y)))).^2+
(C9 + ((sin(x)).*(sin(y)))).^2;
%we use x for alpha and y for beta.
surf(x,y,f)

```

S. No.	Notation used	Notations meaning
1.	$\circ$	Outer product of two vector/matrix/tensor
2.	$\otimes$	Kronecker product
3.	$\mathcal{A}, \mathcal{D}, \mathcal{C}, \mathcal{P}$	Tensor
4.	$A, B, C, \dots, M$	Matrices
5.	$a, b, \dots; \alpha, \beta, \dots; a_1, a_2, \dots$	Scalars (i.e. t is used for transpose of matrix and vector)
6.	$\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots$	Vectors
7.	$I_1, I_2, \dots; I, J, K, \dots$	Indices
8.	$*$ , $\cdot$	Product
9.	$\times_n$	n-mode product (i.e. n-mode product of a tensor $A$ with a matrix $A$ is $\mathcal{A} \times_n A$ )
10.	$\mathbb{R}$	Real vector space
11.	QCF	Quadratic cast function
12.	ALS	Alternative least Square
13.	$\odot$	Khatri-Rao Product

## References

- [1] F. L. Hitchcock: *The expression of a tensor or a polyadic as a sum of products*, J. Math. Phys., **6**(1927), pp 164-189.
- [2] F. L. Hitchcock: *Multiple invariants and generalized rank of a p-way matrix or tensor*, J. Math. Phys., **7** (1927), pp 39-79.
- [3] G. Eckart and G. Young, "The approximation of one matrix by another of lower rank", Psychometrika, **1** (1936), pp. 211-218.
- [4] R. A. Harshman: *Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multimodal factor analysis*, UCLA Working Papers in Phonetics, **16** (1970), pp. 1-84.
- [5] 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.
- [6] J. D. Carroll and J. J. Chang: "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition", Psychometrika, **35** (1970), pp. 283-319.
- [7] J. Mcks: "Topographic components model for event-related potentials and some biophysical considerations", IEEE Trans. Biomed. Engrg., **35** (1988), pp. 482-484.
- [8] H. A. L. Kiers "Towards a standardized notation and terminology in multiway analysis", J. Chemometrics, **14** (2000), pp. 105-122.
- [9] R. B. Cattell, "Parallel proportional profiles and other principles for determining the choice of factors by rotation", Psychometrika, **9** (1944), pp. 267-283.
- [10] C. J. Appellof and E. R. Davidson "Strategies for analyzing data from video fluorometric monitoring of liquid chromatographic effluents", Anal. Chem., **53** (1981), pp. 2053-2056.
- [11] C. M. Andersen and R. Bro "Practical aspects of PARAFAC modeling of fluorescence excitation-emission data", J. Chemometrics, **17** (2003), pp. 200-215.
- [12] N. Sidiropoulos, R. Bro, and G. Giannakis "Parallel factor analysis in sensor array processing", IEEE Trans. Signal Process., **48** (2000), pp. 2377-2388.
- [13] N. Sidiropoulos and R. Budampati "Khatri-Rao space-time codes", IEEE Trans. Signal Process., **50** (2002), pp. 2396-2407.
- [14] N. Sidiropoulos, G. Giannakis, and R. Bro "Blind PARAFAC receivers for DS-CDMA systems", IEEE Trans. Signal Process., **48** (2000), pp. 810-823.
- [15] A. H. Andersen and W. S. Rayens "Structure-seeking multilinear methods for the analysis of fMRI data", NeuroImage, **22** (2004), pp. 728-739.
- [16] E. Martnez-Montes, P. A. Valdes-Sosa, F. Miwakeichi, R. I. Goldman, and M. S. Cohen "Concurrent EEG/fMRI analysis by multiway partial least squares", NeuroImage, **22** (2004), pp. 1023-1034.

- [17] F. Miwakeichi, E. Martinez-Montes, P. A. Valdes-Sosa, N. Nishiyama, H. Mizuhara, and Y. Yamaguchi "Decomposing EEG data into space-time-frequency components using parallel factor analysis", *NeuroImage*, 22 (2004), pp. 1035-1045.
- [18] E. Acar, C. A. Bingol, H. Bingol, R. Bro, and B. Yener "Multiway analysis of epilepsy tensors", *Bioinformatics*, 23 (2007), pp. 110-118.
- [19] M. De Vos, L. De Lathauwer, B. Vanrumste, S. Van Huffel, and W. Van Paesschen "Canonical decomposition of ictal scalp EEG and accurate source localisation: Principles and simulation study", *Comput. Intelligence Neurosci.*, 2007 (2007), pp. 1-8.
- [20] M. De Vos, A. Vergult, L. De Lathauwer, W. De Clercq, S. Van Huffel, P. Dupont, A. Palmi, and W. Van Paesschen "Canonical decomposition of ictal scalp EEG reliably detects the seizure onset zone", *NeuroImage*, 37 (2007), pp. 844-854.
- [21] A. Stegeman "Comparing Independent Component Analysis and the PARAFAC Model for Artificial Multisubject fMRI Data", Tech. report, Heymans Institute for Psychological Research, University of Groningen, The Netherlands, 2007.
- [22] B. W. Bader, M. W. Berry, and M. Browne "Discussion tracking in Enron email using PARAFAC, in Survey of Text Mining: Clustering, Classification, and Retrieval, 2nd ed.", M. W. Berry and M. Castellanos, eds., Springer, 2007, pp. 147-162.
- [23] A. Shashua and A. Levin "Linear image coding for regression and classification using the tensor-rank principle", in *CVPR 2001: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2001, pp. 42-49.
- [24] M. Morup, L. K. Hansen, C. S. Herrmann, J. Parnas, and S. M. Arnfred "Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG", *NeuroImage*, 29 (2006), pp. 938-947.
- [25] M. Morup, L. K. Hansen, and S. M. Arnfred, ERPWAVELAB "A toolbox for multi-channel analysis of timefrequency transformed event related potentials", *J. Neurosci. Methods*, 161 (2007), pp. 361-368.
- [26] 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.
- [27] B. W. Bader and T. G. Kolda "Tensor Decompositions and Applications", *SIAM Review*, Vol. 51. No 3. (2009) pp. 455-500.
- [28] L. De Lathauwer, B. De Moor, and J. Vandewalle "On the best rank-1 and rank-(R1;R2; : : : ;RN) approximation of higher-order tensors", *SIAM J. matrix Anal. Appl.*, 21 (2000), pp. 1324-1342.

**Author information**

Vineet Bhatt, Department of Mathematics, HNB Garhwal University, Campus Badshahi Thaul, Tehri Garhwal-249199, Uttarakhand, India.

S. Kumar, Department of Mathematics, Government Post-Graduate College, New Tehri, Tehri Garhwal, Pin: 249001, Uttarakhand, India.

E-mail: drsandeepkumarmath@hotmail.com

Received: February 7, 2013

Accepted: July 7, 2013