Palestine Journal of Mathematics

Vol. 6(2)(2017) , 533–541

CAS COMPUTATION OF NON-NEGATIVE TENSOR FACTORIZATION

Vineet Bhatt and Suraj Singh Chand

Communicated by Ayman Badawi

MSC 2010 Classifications:15A69, 15A23, 15A72.

Keywords and phrases: Factorization, Algorithm, ALS, NTF, Uniqueness.

Abstract. In this paper, we are studying about non-negative tensor factorization and basic concept of tensor, which is started from a single scalar to n-way array data. We are also studying some important challenges for best computation of NTF and two important algorithms for computing NTF problem. The properties and benefits of the algorithms for NTF also investigated.

1 Introduction

Tensors can be viewed as multilinear arrays or generalizations of the notion of matrices. The concept of tensor can be describe in two way, first presents the notion of a Nth-order tensor as a collection of numbers organized in a N dimensional array, whose elements require N indices to be referred to and second present a tensor as a geometrical entity that "can be expressed as a multi-dimensional array relative to a choice of basis of the particular space on which it is defined. The intuition underlying the tensor concept is inherently geometrical: as an object in and of itself, a tensor is independent of any chosen frame of reference". Both ideas clearly state that a matrix and a vector can be understood as 1st-order and 2ndorder tensors, respectively. The order of a tensor is then understood as its number of dimensions. Tensor decompositions have applications in various fields such as psychometrics [1],signal processing [3], [4] numerical linear algebra and data mining [4] etc. When the data are nonnegative, the NTF better reflects the underlying structure. With NTF it is possible to extract information from a given data set and construct lower-dimensional bases that capture the main features of the set and concisely describe the original data. NMF and its extension known as NTF are emerging techniques that have been proposed recently [2], [4]. NMF and NTF have attracted much attention and have been successfully applied to numerous data analysis problems where the components of the data are necessarily non-negative. The goal of NMF/NTF is to decompose a nonnegative data matrix into a product of lower-rank nonnegative matrices or tensors [4]. NTF are commonly computed as the solution of a nonlinear bound-constrained optimization problem. Some inherent difficulties must be taken into consideration in order to achieve good solutions. NTF problem can describe as, let a given N dimensional tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$,the NTF problem consists in finding a core tensor $\mathcal{C} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_N}$, and a set of N matrices $\mathbf{A}^{(n)} \in \mathbb{R}^{I_n \times J_n}$, $n = 1, 2, ..., N$,such that $\mathcal{A} \approx \mathcal{C} \times_1 \mathrm{A}^{(1)} \times_2 \mathrm{A}^{(2)} .... \times_N \mathrm{A}^{(N)}$, where the $N + 1$ constitutive factors $\mathcal{C}, \mathrm{A}^{(1)}, \mathrm{A}^{(2)}, ..., \mathrm{A}^{(N)}$ are required to be component wise nonnegative,i.e.,$\mathrm{A}^{(1)}, \mathrm{A}^{(2)}, ..., \mathrm{A}^{(N)} \geq 0$. We have some examples of models in which tensors are used to organize data of different aspects of the problem; we use super-indexes to denote the modalities (i.e., aspects to be measured in the problem) considered along the tensor modes and sub-indexes to indicate the data type.In [5]tensor decomposition has been used in web mining $\mathcal{A}_{clickcounts} \in \mathbb{R}^{(Users \times queries \times wbepages)}$, in [6] spectroscopy data spectroscopy data
$\mathcal{A}_{strength} \in \mathbb{R}^{(Batchnumber \times time \times spectra)}$, in [7] semantic differential data
$\mathcal{A}_{grade} \in \mathbb{R}^{(Judges \times musicpieces \times scales)}$, and in [8] image analysis
$\mathcal{A}_{imageintensity} \in \mathbb{R}^{(people \times views \times illuminations \times expressions \times pixels)}$. MÃÿrup et al. [9] give an example where nonnegative factorization is used to analyze EEG data.They are aware of the impact that collapsing several modalities into a single dimension can have in the exploration performed, and they use both NMF and NTF to fully analyze the dataset and extract different information about the problem: NTF to perform an overall analysis, and NMF to perform a local

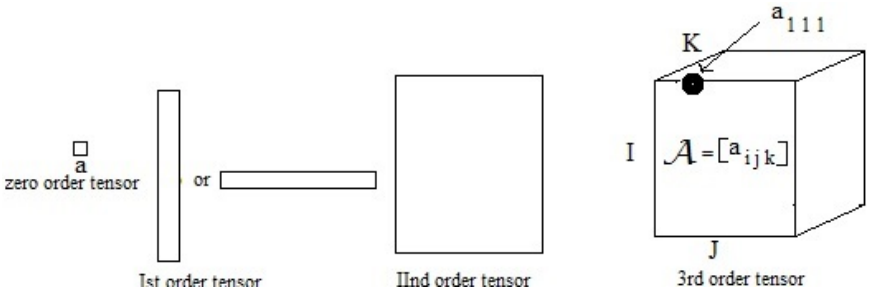

Figure 1. Visualization of tensors

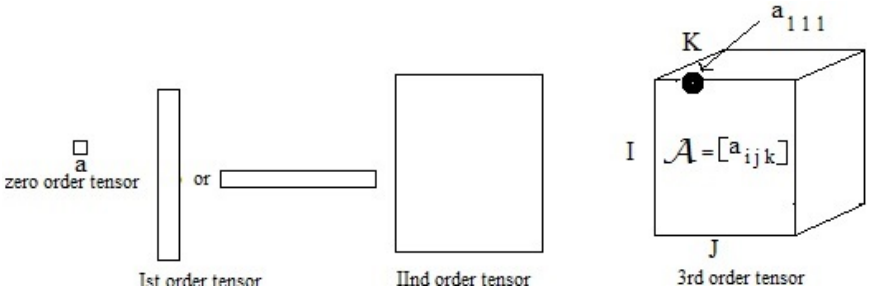

Figure 2. Fibers for matrix and tensor

one (while NMF can examine subject specific activities and search for time-frequency patterns, NTF can effectively extract the most similar activities across subjects and/or conditions). Thus, the use of tensors simplifies the organization of the data and provides an alternate insight into the problem. This paper is organized as follows; in section 2 we are studying about some knowledge of tensor like representation, definition etc and in section 3 some important algorithms.

2 Tensor's Representation, Definition and their operation [4]

In this section we are going to represent tensor by geometrical structure and their important definitions which are useful for our future development.

2.1 Tensor's Representation:

Tensors are generalization of vectors and matrixes; we can say that a scalar is a zero order tensor, a vector is a first order tensor, a matrix is a second order tensor and a cubicle and cuboidicle shapes data are higher order tensor. In fig.1, a is zero order tensor,$\mathbf{a}$ is first order tensor, A is a second order tensor and $\mathcal{A}$ is third order tensor where a_{ijk} is a element of $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$.

2.2 Subtensors:

Subtensors or sub-arrays are formed when a subset of the indices is fixed. These are the rows and columns in case of matrix. A colon is used to indicate all elements of a mode in the style of MATLAB. Thus, the i^{th} column of a matrix $A = [\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_I}]$ is formally denoted by $\mathbf{a}_{:,\mathbf{i}}$; a likewise, the i^{th} row of A is denoted by $\mathbf{a}_{\mathbf{i},:}$.

2.3 Tensor's Fiber:

A tensor fiber is a one-dimensional fragment of a tensor, obtained by fixing all indices except for one. A matrix have two types fiber, column of matrix is mode-1 fiber and row of matrix is mode-2 fiber but a third order tensor $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$ have three types of fiber; column, row and tube fibers denoted by $\mathbf{a}_{:,\mathbf{j},\mathbf{k}}$, $\mathbf{a}_{\mathbf{i},:,\mathbf{k}}$ and $\mathbf{a}_{\mathbf{i},\mathbf{j},:}$ respectively (see fig. 2).

2.4 Tensor Slice:

A two dimensional section or fragment of a tensor is called tensor slice, obtained by fixing all indices except for two indices. A third order tensor $\mathcal{A} \in \mathbb{R}^{I\times J\times K}$ have three types of slice, horizontal, lateral and frontal slices denoted by $A_{(}i ::)$,$A_{(} : j :)$ and $A_{(} :: k)$ respectively [4].

2.5 Unfolding or Matricization:

If one want to study about higher order tensor data, it is necessary to represent tensor decomposition in their matrix form. Unfolding, also known as matricization, is a process of reordering the elements of an N-th order tensor into a matrix. In [4] author have described various ways to order the fibers of tensors, therefore, the unfolding process is not unique [10].

2.6 Tensor's definition:

An Nth-order tensor is formally defined as the tensor product of N vector spaces. (a): In [11] author has defined tensors by following definition. Let $\mathbf{a_1}, \mathbf{a_2}, ..., \mathbf{a_N}$ be N Euclidean vector spaces with finite dimensions $I_1, \mathbf{I_2}, ..., \mathbf{I_N}$. Consider N vectors $\mathbf{v_n} \in \mathbf{a_n}, \mathbf{n} = \mathbf{1}, \mathbf{2}, ..., \mathbf{N}$.
We denote by $(\mathbf{v_1 o v_2 o}....\mathbf{o v_N})$ the multilinear mapping on $\mathbf{a_1} \times \mathbf{a_2} \times ... \times \mathbf{a_N}$, defined by

$$(\mathbf{v_1 o v_2 o}....\mathbf{o v_N})(\mathbf{x_1}, \mathbf{x_2}, , \mathbf{x_N}) =< \mathbf{v_1 x_1} >_{\mathbf{a_1}} < \mathbf{v_1 x_1} >_{\mathbf{a_1}} .... < \mathbf{v_N x_N} >_{\mathbf{a_N}}$$

in which $< \mathbf{v_N x_N} >_{\mathbf{a_N}}$ denotes the scalar product in $\mathbf{a_n}$, and $\mathbf{x_n}$ is an arbitrary vector in $(\mathbf{a_n}(\mathbf{1} \leq \mathbf{n} \leq \mathbf{N})$. The space generated by all the elements $(\mathbf{v_1 o v_2 o}....\mathbf{o v_N})$ is called the tensor product space of $\mathbf{a_1} \times \mathbf{a_2} \times ... \times \mathbf{a_N}$. An element of the tensor product space is called an N-th order tensor (over $\mathbf{a_1} \times \mathbf{a_2} \times ... \times \mathbf{a_N}$). (b): A tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times ...\times I_n}$ of order N is an N-way arrays where element $a_{i_1 i_2 ... i_n}$ are indexed by $i_n \in 1, 2, ..., I_n$ for $1 \leq n \leq N$ and $I_1, I_2, , I_N$ denote index upper bounds [4].

2.7 Definition of Unfolding:

Let we have a higher order tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times ...\times I_N}$, the mode-n unfolding is denoted by A_n and arranges the mode-n fibers into columns of a matrix. We can say that a tensor element $(i_1, i_2, ..., i_n)$ maps onto a matrix element i_n, j, where
$j = 1 + \sum_{p\neq n} (i_p - 1)J_p where J_p = \begin{pmatrix} 1, if p = 1 or if p = 2 and n = 1 \\ \Pi_{m\neq n}^{(p-1)} I_m, \end{pmatrix}$
Observe that in the mode-n unfolding the mode-n fibers are rearranged to be the columns of the matrix A_n. More generally, a subtensor of the tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times ...\times I_N}$, denoted by $A_{(i_n=j)}$, is obtained by fixing the n-th index to some value j. For example, a third-order tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times I_3}$ with entries $a_{(i_1,i_2,i_3)}$ and indices (i_1, i_2, i_3) has a corresponding position (i_n, j) in the mode-n unfolded matrix $A_n(n = 1, 2, 3)$ as follows
•$Mode - 1 : j = i_2 + (i_3 - 1)I_2$,
•$Mode - 2 : j = i_1 + (i_3 - 1)I_1$,
•$Mode - 1 : j = i_1 + (i_2 - 1)I_1$,
note that mode-n unfolding of a tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times ...\times I_N}$, also represents mode-1 unfolding of its permuted tensor $\mathcal{A} \in \mathbb{R}^{I_n\times I_1\times ...\times I_{n-1}\times I_{n+1}\times ...\times I_N}$, obtained by permuting its modes to obtain the mode-1 be I_n.

2.8 Special forms of Tensors:

Diagonal tensor: A diagonal tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times ...\times I_N}$, is a tensor that has nonzero elements $a_{(i_1,i_2,,i_N)}$ only when $i = i_1 = i_2 = ... = i_N$, for $1 \leq i \leq min(I_n)$. With this idea we can define the identity tensor as the diagonal tensor whose nonzero entries $a_{(i_1,i_2,,i_N)}$ are all one (see fig 3.).
Rank-one Tensor: A tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times ...\times I_N}$ of order N has rank-one if it can be written as an outer product of N vectors i.e.,$\mathcal{A} = \mathbf{a_1 o a_2 o}.\mathbf{o a_N}$, where $\mathbf{a_n} = \mathbb{R}^{(\mathbf{I_n})}$ and $a_{(i_1,i_2,,i_N)} = a_{i_1}^1 a_{i_2}^2 a_{i_N}^N$.
The rank of a tensor $\mathcal{A}$ is defined as the minimal number of rank-one tensors.
Symmetric and Super-Symmetric Tensors: For the particular case when all the N vectors $\mathbf{a^{(j)}}$

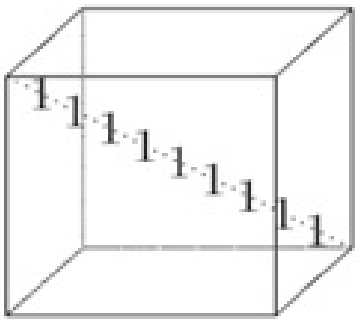

Figure 3. Identity Tensor

are equal to a vector **v**, their outer product is called a super-symmetric rank-one tensor. A super symmetric tensor has the same dimension in every mode or in general, by analogy to symmetric matrices a higher-order tensor is called super-symmetric if its entries are invariant under any permutation of their indices. Tensor can also only be symmetric in two or more modes. For example, a three-way tensor $\mathcal{A} \in \mathbb{R}^{I \times J \times K}$ is symmetric in modes one and two if all its frontal slices are symmetric.

3 Non-negative Tensor Factorization:

Let one have an N-th order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$ then the NTF problem consists in finding an N-th order core tensor $\mathcal{C} \in \mathbb{R}^{J_1 \times J_2 \times \ldots \times J_N}$, and N matrices $A^{(1)} = \mathbb{R}^{I_1 \times J_1}, A^{(2)} = \mathbb{R}^{I_2 \times J_2}, .., A^{(N)} = \mathbb{R}^{I_N \times J_N}$, whose multi-mode product approximates a given tensor $\mathcal{A}$, i.e.,

$$\mathcal{A} \approx \mathcal{C} \times_1 A^{(1)} \times_2 A^{(2)} ... \times_N A^{(N)} \tag{3.1}$$

A common approach to compute these factors [4] is to solve the nonlinear least-squares problem.

$$min_{\mathcal{C}, A^{(1)}, A^{(2)}, ..., A^{(N)}} F(\mathcal{C}, A^{(1)}, A^{(2)}, ..., A^{(N)}), \tag{3.2}$$

subject to $\mathcal{C}, A^{(1)}, A^{(2)}, ..., A^{(N)} \geq 0$. where

$$F(\mathcal{C}, A^{(1)}, A^{(2)}, ..., A^{(N)}) = \frac{1}{2} \| \mathcal{C} \times_1 A^{(1)} \times_2 A^{(2)} \times_3 ... \times_N A^{(N)} - \mathcal{A} \|_F^2 \tag{3.3}$$

In the CP model that we consider here, we are seeking an Nth-order diagonal tensor $\mathcal{C} \in \mathbb{R}^{(r \times \ldots \times r)}$ (the core tensor), and matrices that have r columns,i.e.,$A^{(n)} \in \mathbb{R}^{(I_n \times r)}, n \in 1, 2, .., N$. When N=2 in equation (3.2), the problem has reduced to NMF.

$$min_{A^{(1)}, A^{(2)}} \frac{1}{2} \| A^{(1)} C A^{(2)T} - V \|_F^2 \tag{3.4}$$

subject to $C, A^{(1)}, A^{(2)} \geq 0$

Note that C is not needed in this formulation, but we keep it as a factor to maintain the analogy with equation (3.2). We use this matrix form of the problem to illustrate particular aspects of the more general NTF problem. In this section we provide a description of the main difficulties associated with the NTF problem and explain the ideas behind the most popular approaches to solving the problem.

3.1 Some challenges in NTF problem:

The NTF problem usually appears in applications that involve a large number of variables. Some important challenges are discussed below;

Lack of Uniqueness:

The N-mode tensor factorization is never unique [21]. If it is assumed that U,V and W are nonsingular matrices, then

$$[\mathcal{C}; A^{(1)}, A^{(2)}, A^{(3)}] = [\mathcal{C} \times_1 U \times_2 V \times_3 W; A^{(1)}U^{-1}, A^{(2)}V^{-1}, A^{(3)}W^{-1}]. \tag{3.5}$$

Thus, any of the n-mode factors can be modified without affecting the product as long as the inverse modification is applied to the core tensor. This is known as scaling indeterminacy. This fact is more easily seen in the matrix case, where we have

$$WCH^T = (WD_1^{(-1)})(D_1CD_2)(HD_2^{(-T)})^T. \tag{3.6}$$

Furthermore, in the case of the CP decomposition where the elements of the diagonal tensor $\mathcal{C}$ are fixed and all equal, i.e., $(\lambda = c_1 = c_2 = .. = c_r)$, we also have permutation indeterminacy. We know that in the CP model,

$$[A^{(1)}, A^{(2)}, .., A^{(N)}] = [A^{(1)}P, A^{(2)}P, ..., A^{(N)}P]. \tag{3.7}$$

for any $r \times r$ permutation matrix P. Therefore, we could arrange the rank-1 factors in a different order, which would mean applying the same permutation P to all the columns of each factor $A^{(n)}, n = 1, 2, ...N$, and still get the same approximation. Note that if $\mathcal{C}$ is also considered a variable, this indeterminacy is always present, since it is possible to apply the permutation P to the diagonal elements of $\mathcal{C}$ and maintain the product unchanged. These two operations, scaling and permutation, are known as elementary changes. Two rank-r decompositions of a given tensor$\mathcal{A}$ are equivalent if one can be obtained from the other by elementary changes. Kruskal defines as unique a decomposition (of a tensor $\mathcal{A}$) that has rank-r and is equivalent to all other rank-r decompositions of $\mathcal{A}$. However, the lack of uniqueness in the NTF problem is insidious. In his work, Kruskal also proves that 3rd-order decompositions are not unique (according to his definition) if their ranks are large enough, although the expression (large enough) does not have a specific quantification. Despite the absence of a result concerning the maximum rank that guarantees uniqueness (up to elementary changes), it is observed that the indeterminacy increases with the size of the tensor. It has been conjectured, for example, that for an array of dimensions $r \times r \times r$, any approximation of order greater than $\frac{3r}{2} - 1$ is not unique [21]. Clearly, the presence of all these indeterminacies makes the problem ill-posed.

Degeneracy and best rank-R approximation:

A tensor is degenerate if it can be approximated arbitrarily well by a factorization of lower rank [12],[13]. The property is better understood through the use of an example. Here we outline the one used in[13], a model carefully constructed to illustrate degeneracy in a 3rd-order tensor. Consider the rank-3 tensor $\mathcal{A} \in \mathbb{R}^{n\times n\times n}$ given by $\mathcal{A} = [\widehat{\mathbf{a}}, \mathbf{b}, \mathbf{c}] + [\mathbf{a}, \widehat{\mathbf{b}}, \mathbf{c}] + [\mathbf{a}, \mathbf{b}, \widehat{\mathbf{c}}]$, where $\mathbf{a}, \mathbf{b}, \mathbf{c}$ and $\widehat{\mathbf{a}}, \widehat{\mathbf{b}}, \widehat{\mathbf{c}}$ are vectors in $\mathbb{R}$. The tensor given by the product

$$\begin{aligned}
\mathcal{A}_\epsilon = [A, B, C] = \\
[(\frac{1}{\epsilon}\mathbf{a} + \frac{\epsilon^2}{2}\hat{\mathbf{a}} \mid -\frac{1}{\epsilon}\mathbf{a} + \frac{\epsilon^2}{2}\hat{\mathbf{a}}), (\frac{1}{\epsilon}\mathbf{b} + \frac{\epsilon^2}{2}\hat{\mathbf{b}} \mid -\frac{1}{\epsilon}\mathbf{b} + \frac{\epsilon^2}{2}\hat{\mathbf{b}}), (\frac{1}{\epsilon}\mathbf{c} + \frac{\epsilon^2}{2}\hat{\mathbf{c}} \mid -\frac{1}{\epsilon}\mathbf{c} + \frac{\epsilon^2}{2}\hat{\mathbf{c}})] = \\
(\frac{1}{\epsilon}\mathbf{a} + \frac{\epsilon^2}{2}\hat{\mathbf{a}})o(\frac{1}{\epsilon}\mathbf{b} + \frac{\epsilon^2}{2}\hat{\mathbf{b}})o(\frac{1}{\epsilon}\mathbf{c} + \frac{\epsilon^2}{2}\hat{\mathbf{c}}) + (-\frac{1}{\epsilon}\mathbf{a} + \frac{\epsilon^2}{2}\hat{\mathbf{a}})o(-\frac{1}{\epsilon}\mathbf{b} + \frac{\epsilon^2}{2}\hat{\mathbf{b}})o(-\frac{1}{\epsilon}\mathbf{c} + \frac{\epsilon^2}{2}\hat{\mathbf{c}}) = \\
[\hat{\mathbf{a}}, \mathbf{b}, \mathbf{c}] + [\mathbf{a}, \hat{\mathbf{b}}, \mathbf{c}] + [\mathbf{a}, \mathbf{b}, \hat{\mathbf{c}},]
\end{aligned} \tag{3.8}$$

is a rank-2 tensor. However, we clearly have $\|\mathcal{A}_\epsilon - \mathcal{A}\|_F = \frac{\epsilon^6}{4}\|[\hat{\mathbf{a}}, \hat{\mathbf{b}}, \hat{\mathbf{c}}]\|_{\mathbf{F}}$,which decreases as $\epsilon \to 0$. Thus, $\mathcal{A}$ is a rank-3 tensor that is arbitrarily well approximated by the rank-2 tensor $\mathcal{A}_\epsilon$; in other words, we have a rank-3 tensor for which there is not a best approximation of rank 2. Extensive research has been carried out for particular types of tensors (e.g., $(I \times I \times 2)$ tensors [14]). In general, it has been proven that for any given size there is always a set of tensors that; at least for certain values of r, exhibits degeneracy [15]. This result implies that any given NTF problem may not have a solution.

Lack of bounds:

Note that the scaling indeterminacy described in 1. (Lack of Uniqueness) makes it possible to have extremely unbalanced factors, i.e., some factors have arbitrarily large norms, while others have arbitrarily small norm. For example, for any $\gamma \neq 0$.

$$\mathcal{C} \times_1 A^{(1)} \times_2 A^{(2)} \times_3 A^{(3)} = \mathcal{C} \times_1 (\gamma A^{(1)}) \times_2 (\frac{1}{\gamma} A^{(2)}) \times_3 A^{(3)} \tag{3.9}$$

Furthermore, this unbalanced variation in the elements of the factors can also be caused by degeneracy. When we try to approximate a degenerate tensor $\mathcal{A}$, we find a "best-fit" solution that gets arbitrarily closer to $\mathcal{A}$ as some (or all) its factors approach infinity. This lack of bounds has the effect of producing ill-conditioned factors whenever the original tensor is degenerate.

3.2 Algorithms for computing the NTF:

In this section, we are studying some algorithms for computing NTF problem. Almost all NMF algorithms can be extended or generalized to the various nonnegative tensor factorizations. In this section we mainly focus on two important algorithms: Multiplicative updating algorithm and alternative least squares (ALS) algorithm [4]. Multiplicative updating algorithm: This algorithm was proposed by Lee and Seung [16]. It was first method available for the NMF problem. It was extended to the NTF problem by Welling and Weber [17]. Algorithm I outlines a basic version of the multiplicative updating algorithm for the NTF problem.

Algorithm-I: Multiplicative update rule for NMF

Input: $\mathcal{A}$: N-Way tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_N}$

N initial mode-factors $A^i \geq 0$, i=1,2,..,N

while the residual norm is greater than a tolerance do

for i =1,2,...,N do

From $V_i = V_{(i)}$;

Compute $K = A^{(i)}_{\bigodot}$;

Update A^i using the rule.

$A^{(i)} \leftarrow A^{(i)} * \frac{V_i K}{A^{(i)} K^T K}$

Scale the factors to keep them bounded

end

Compute the residual

end

Properties of Multiplicative update rule:

1. First, note that the algorithm involves only the computation of the matrices $A^{(1)}, \ldots A^{(N)}$ and assumes that $\mathcal{C}$ is the identity tensor, i.e., this update rule is designed only for CP-factorizations.
2. Because each update is obtained using only multiplications, if the initial guess consists of nonnegative factors, the subsequent iterates will also be nonnegative. The algorithm thus produces nonnegative factors, as required.
3. It is guaranteed that the algorithm converges to a stationary point. This has been proved in [18].

Alternating least squares: A way of dealing with optimization problems where the objective function and constraint functions have a partially separable structure with respect to the problem's variables consists of breaking down the problem into smaller ones that can be solved one-at-a-time in sequence. The ALS approach fixes all the factors except one, and optimizes the problem with respect to the free factor. In this way, the nonlinear least-squares problem is decomposed into a sequence of smaller, linear problems. Consider the matrix case and recall that we want to solve the problem

$$min_{W,G,H \geq 0} \frac{1}{2} \parallel V - WGH^T \parallel_F^2, \tag{3.10}$$

where $V \in \mathbb{R}^{(m \times n)}, W \in \mathbb{R}^{(m \times r)}, G \in \mathbb{R}^{(r \times r)} and H \in \mathbb{R}^{(n \times r)}$. We have explicitly written the factor C (equation (3.4)) but for now, C=I. If we follow the ALS approach, the problem given by

equation (3.10) can be decomposed into two least squares problems,

$$min_{W\geq 0}\frac{1}{2}\parallel V-WH^T\parallel_F^2, and min_{H\geq 0}\frac{1}{2}\parallel V-WH^T\parallel \tag{3.11}$$

Note that each of these least squares problems is a collection of linear-squares problem. For example, the first problem in eq. (11) can be solved by the sequence of problems

$$min_{h_j\geq 0}\frac{1}{2}\parallel v_j-Wh_j^T\parallel_2^2, j=1,2,...,n. \tag{3.12}$$

Where $\mathbf{v_j}$ and $\mathbf{h_j}$ are columns vectors of H and V respectively. In the tensor case, the general nonlinear least squares problem is partitioned into the sub-problems

$$NTF_{A^{(n)}}min_{A^{(n)}}1/2\parallel [\mathcal{C};A^{(1)},...,A^{(n)},..A^{(N)}]-\mathcal{A}\parallel_F^2 \tag{3.13}$$

subject to $A^{(n)}\geq 0 for n=1,2,..,N.$ and

$$NTF_{\mathcal{C}}min_{\mathcal{C}}\frac{1}{2}\parallel [\mathcal{C};A^{(1)},...,A^{(n)},..A^{(N)}]-\mathcal{A}\parallel_F^2 \tag{3.14}$$

$subjectto\mathcal{C}\geq 0$. When solving the sub-problem associated with each matrix $A^{(n)}$, we have the option of matricizing the problem

$$min_{A^{(n)}}\frac{1}{2}\parallel A_{\bigotimes}^{(n)}C_{(n)}^TA^{(n)T}-V_{(n)}^{(T)}\parallel_F^T \tag{3.15}$$

subject to $A^{(n)}\geq 0$. by computing one row of $A^{(n)}$ at a time. In this way we can take advantage of the standard linear least-squares algorithms to solve each sub-problem. Another possibility is to vectorize the problem to solve for all the rows of $A^{(n)}$ simultaneously. To do this, recall, from matrix calculus, that for any three matrices A, X, and B of suitable size, we have

$$vec(A\times B)=(B^T\otimes A)vec(X) \tag{3.16}$$

Thus

$$vec(A_{\bigotimes}^{(n)}C_{(n)}^TA^{(n)T})=vec((A_{\bigotimes}^{(n)}C_{(n)}^T)A^{(n)}I_{I_n})=(I_{(I_n)}\otimes A_{\otimes}^{(n)}C_{(n)}^T)vec(A^{(n)T}), \tag{3.17}$$

and equation (3.15) becomes

$$min_A^{(n)}1/2\parallel (I_{(I_n)}\otimes (A_{\otimes}^{(n)}C_{(n)}^T))vec(A^{(n)T})-vec(V_{(n)}^T)\parallel_2^2 \tag{3.18}$$

subject to $A^n\geq 0$. To solve the sub-problem associated with the core tensor ??, we need to isolate $C_{(n)}$ in any of the sub-problems given by equation (3.15). We can use equation (3.16) again to get

$$vec(A^{(n)}C_{(n)}A_{\otimes}^{(n)T})=(A_{\otimes}^{(n)}\otimes A^{(n)})vec(C_{(n)}) \tag{3.19}$$

which by choosing n=1, becomes

$$vec(A^{(1)}C_{(1)}A_{\otimes}^{(n)T})=A_{\otimes}vec(C_{(1)}), \tag{3.20}$$

and leads to the optimization sub-problem

$$min_{\mathcal{C}}\frac{1}{2}\parallel A_{\otimes}vec(C_{(1)})-vec(V_{(1)})\parallel_2^2 \tag{3.21}$$

subject $\mathcal{C}\geq 0$. The ALS approach was first used in the tensor case by Bro and De Jong [19]in 1997. Some later modifications include the work of Friedlander and Hatz [20], who incorporated a regularization function and a strategy to keep the variables bounded using the l_1-norm. They also use the vectorized version of the problem and solve simultaneously for all the rows of each matrix factor, increasing the computations' efficiency. Depending on the implementation, ALS algorithms can be effective and the drawbacks faced by these algorithms are those that are inherent in the NTF problem: no guarantee of convergence to a local minimum (it could be a saddle point), great sensitivity to the initial point, and a considerable increase on the work when additional constraints are imposed.

4 Conclusion

Non-negative tensor decomposition is a fascinating emerging field of research, with many applications. In this review/study paper, we have briefly discussed tensor representation, sub-tensor, tensor fiber etc. NTF problem consists in finding an Nth order core tensor and N matrices but it is lack of uniqueness and lack of bounds. A tensor is degenerate if it can be approximated arbitrarily well by a factorization of lower rank. Multiplicative update algorithm is a best algorithm for computing NTF problem because it involves only the computation of matrices and assumes that core tensor is identity tensor and each update is obtained using only multiplications, if the initial guess consists of positive factor, the subsequent iterates will also be positive. Finally ALS algorithm for computing NTF fixes all the factors except one, and optimizes the problem with respect to the free factor and nonlinear least-squares problem is decomposed into a sequence of smaller, linear problems.

5 Acknowledgment:

The author(s) express their hearty thanks and gratefulness to Dr. K. C. Petwal, H.N.B.G.U., department of mathematics, campus Tehri and all those scientists whose masterpieces have been consulted during the preparation of the present study paper. Special thanks are due to Professor(s) J. B. Kruskal, D. Lee and H. Seung, whose fabulous research work has been entirely consulted to expose the present research article.

S. No.	Notation used	Notation's meaning
1.	$\circ$	Outer product of two vector/matrix/tensor
2.	$\otimes$	Kronecker product
3.	$\mathcal{A}$, $\mathcal{D}$, $\mathcal{C}$, $\mathcal{P}$	Higher order Tensor
4.	A, B, C, ..., M	Matrices
5.	a, b, ...; α, β, ...; a_1, a_2..	Scalars (i.e. t is used for transpose of matrix and vector)
6.	$\mathbf{a}$, $\mathbf{b}$, $\mathbf{c}$....	Vectors
7.	I_1, I_2....; I, J, K....	Indices
8.	$*$, $\cdot$	Product
9.	$\times_n$	n-mode product of tensor by matrix
10.	$\mathbb{R}$	Real vector space
11.	ALS	Alternative least Square
12.	$\odot$	Khatri-Rao Product
13.	NMF	Non-negative matrix factorization
14.	$A_{(n)}$, $C_{(n)}$...	Matricization of tensor $\mathcal{A}$, $\mathcal{C}$...

References

[1] 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, pp. 3-24, (1980).

[2] B. W. Bader and T. G. Kolda, Tensor Decompositions and Applications, SIAM Review, Vol. 51. No 3, pp. 455-500, (2009).

[3] Vineet Bhatt and S. Kumar, A Survey on PARATREE and SUSVD Decomposition Techniques and their Use in Array Signal Processing, Accepted for publication in I. J. Image, Graphics and Signal Processing, (2013).

[4] Andrzej Cichocki et al., Nonnegative matrix and tensor factorizations: Applications to exploratory multi-way data analysis and blind source separation,John Wiley and Sons, Ltd, (2009).

[5] J. T. Sun et al., Cube SVD: A novel approach to personalized web search, In A. Press, editor, 14th International Conference on World Wide Web, pp. 652-662, (2005).

[6] C. A. Andersson and R. Bro., Improving the speed of multiway algorithms, part I: Tucker3. Chemometrics and Intelligent Laboratory Systems, 42: pp. 93-103, (1998).

[7] T. Murakami and P. M. Kroonenberg., Three-mode models and individual differences in semantic differential data, Multivariate Behavioral Research, 38(2): pp. 247-283, (2003).

[8] M. A. O. Vasilescu and D. Terzopoulos., Multilinear image analysis for facial recognition, In Proceedings of the International Conference on Pattern Recognition, volume 3, pp. 511-514, (2002).

[9] M. MÃ ÿrup et. al., Decomposing the time-frequency representation of EEG using nonnegative matrix and multiway factorization, Technical report, Technical University of Denmark, (2006).

[10] Vineet Bhatt, Link between HOSVD of a Tensor and SVD of Matrix Unfolding, International Journal of Physical, Chemical and Mathematical Sciences, Vol. 2; No. 1: ISSN: 2278-683X, pp -128-139, (Jan-June 2013).

[11] L. d. Lathauwer., Signal processing based on multilinear algebra, PhD thesis, Katholieke Universiteit Leuven, September (1997).

[12] V. De Silva and L. Lim., Tensor rank and the ill-posedness of the best low-rank approximation problem, Technical Report 06-06, Stanford University, (2006).

[13] P. Paatero., Construction and analysis of degenerate parafac models, Journal of Chemometrics, 14: pp. 285-299, (2000).

[14] A. Stegeman., Degeneracy in CANDECOMP/PARAFAC explained for $p \times p \times 2$ arrays of rank p+1 or higher, Psychometrika, 71: pp. 483-501, (2006).

[15] V. De Silva and L. H. Lim., Tensor rank and ill-posedness of the best low-rank approximation problem, SIAM Journal on Matrix Analysis and Applications, (2006).

[16] D. Lee and H. Seung., Learning the parts of objects by nonnegative matrix factorization, Nature, 401: pp. 788-791, (1999).

[17] M. Welling and M. Weber., Positive tensor factorization Pattern Recognition Letters, 22: pp. 1255-1261, (2001).

[18] M. Berry. et al., Algorithms and applications for approximate nonnegative matrix factorizations, Computational Statistics and Data Analysis, 52(1): pp. 155-173, September(2007).

[19] R. Bro and S. D. Jong., A fast non-negativity-constrained least squares algorithm, Journal of Chemometrics, 11: pp. 393-401, (1997).

[20] M. P. Friedlander and K. Hatz., Computing nonnegative tensor factorizations, Technical Report TR-2006-21, University of British Columbia, Dept. of Computer Science, October (2006).

[21] J. B. Kruskal., Three-way arrays: rank and uniqueness of trilinear decompositions with application to arithmetic complexity and statistics, Linear Algebra Applications, 18: pp. 95-138, (1977).

Author information

Vineet Bhatt and Suraj Singh Chand, Department of Mathematics, G.E.H.U., Dehradun, Uttarakhand, India.
E-mail: vineet.bhatt58@gmail.com

Received: August 1, 2015.
Accepted: January 6, 2016.