Signal Processing 144 (2018) 77–86

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Non-negative sub-tensor ensemble factorization (NsTEF) algorithm. A new incremental tensor factorization for large data sets.

Vincent Vigneron[a,*], Andreas Kodewitz[a], Michele Nazareth da Costa[b], Ana Maria Tome[c], Elmar Langlang[d]

[a] IBISC, Univ Evry, Université Paris-Saclay, 91025, Evry, France
[b] DSPCom Laboratory, University of Campinas (UNICAMP), PO Box 6101, 13083-852, Campinas/SP, Brazil
[c] Departamento de Electronica, Telecomunicacoes e Informática, Universidade de Aveiro, Portugal
[d] Institut für Biophysik und physikalische Biochemie, University of Regensburg, Universitätsstrasse 31, D-93040 Regensburg, Germany

ARTICLE INFO

Article history:
Received 18 December 2016
Revised 9 August 2017
Accepted 10 September 2017
Available online 22 September 2017

Keywords:
Non-negative tensor decomposition
CANDECOMP/PARAFAC Decomposition
Matrix factorization
NTF
Incremental algorithm
Learning method

ABSTRACT

In this work we present a novel algorithm for nonnegative tensor factorization (NTF). Standard NTF algorithms are very restricted in the size of tensors that can be decomposed. Our algorithm overcomes this size restriction by interpreting the tensor as a set of sub-tensors and by proceeding the decomposition of sub-tensor by sub-tensor. This approach requires only one sub-tensor at once to be available in memory.

1. Introduction

Since the pioneering works [42,52], non-negative matrix factorization (NMF) has attracted much interest in the context of several applications such as image and signal processing, computer vision, data analysis, blind source separation [12,22,27,30,32,42,45,53,69]. Particularly in image processing, the associated constraints are desirable to retain the non-negative characteristics of the original data, since the pixel values of the basis images essentially share this feature, leading to a natural meaning regarding the underlying components. As a result, we can, for instance, better represent a face as a linear combination of basis images by NMF in contrast with classical methods such as principal component analysis (PCA) [42,45].

Furthermore, NMF can be viewed as an implicit sparse representation of the input data [27,30,42], which allows representing local features of distributed parts over a human face such as eyes, nose and mouth, and, consequently, learning features of images in face recognition applications. Broadly speaking, a subspace representation by non-negative factorizations makes possible to determinate hidden structures and characteristics inherent to an object class of the input data set, which is helpful in object recognition, detection of semantic features of text documents and of spectral characteristics of hyperspectral images, among others [1,12,22,27,42,46,53,62,69].

Tensorial approaches naturally arise from multilinear structures or multidimensional data, and NMF has been extended to higher-order tensors by the non-negative tensor factorizations (NTFs). The NTF was firstly introduced [6] by imposing non-negative constraints over the matrix factors of the well-known decomposition called CANDECOMP/PARAFAC (or, shortly CP) [5,24]. Analogously, a non-negative version of the Tucker decomposition [61] has also been presented and is referred to here as non-negative Tucker decomposition (NTD) [3], representing a more complex model, as the core tensor could be dense and the matrix factors not necessarily have the same number of columns. An interesting advantage of the NTF/NTD is that, in general, tensor decompositions are essentially unique under mild conditions, as opposed to NMF. More precisely, the uniqueness issue associated with the NTD/NTF happens when the factors are not sufficiently sparse [70].

Almost all NMF algorithms can be generalized or extended to non-negative tensor factorizations by the use of unfolding matri-

* Corresponding author.
E-mail addresses: vincent.vigneron@ibisc.univ-evry.fr (V. Vigneron), andreas.kodewitz@ibisc.univ-evry.fr (A. Kodewitz), nazareth@decom.fee.unicamp.br (M.N. da Costa), ana@ua.pt (A.M. Tome), Elmar.Lang@biologie.uni-regensburg.de (E. Langlang).

https://doi.org/10.1016/j.sigpro.2017.09.012
0165-1684/

ces of the higher-order tensor or by a multi-layer strategy (multi-factor model) [12,17]. A very popular multiplicative update (MU) method [42] can be derived, regarding gradient descent methods, by solving the following optimization problem

$$(\mathbf{A}^*, \mathbf{B}^*) = \arg\max_{A,B} f(\mathbf{A}, \mathbf{B}) = \arg\max_{A,B} \frac{1}{2}\|\mathbf{Y} - \mathbf{AB}\|_F^2. \quad (1)$$

The MU rule with a simple projection to the non-negative space at each updating step is given by

$$b_{n,p} \leftarrow b_{n,p}\left[\frac{\mathbf{A}^{\mathrm{T}}\mathbf{Y}}{\mathbf{A}^{\mathrm{T}}\mathbf{AB}}\right]_{n,p,+}, \qquad a_{m,n} \leftarrow a_{m,n}\left[\frac{\mathbf{YB}^{\mathrm{T}}}{\mathbf{ABB}^{\mathrm{T}}}\right]_{m,n,+}, \quad (2)$$

which has a simple and easy implementation despite presenting slow convergence [49].

A basic approach to NMF, called alternating non-negative least squares (ANLS), is driven by alternating least squares (ALS) techniques [12] based on the alternating minimization of the cost function (1) with respect to the nonnegativity-constrained matrices **A** and **B** separately. However, it does not necessarily lead to global minimization. Several algorithms have been proposed based on the ANLS framework with the purpose of accelerating and overcoming the unstable convergence properties of the standard ANLS and, also, becoming more robust to noise [12] by including penalty terms on the cost function (1) to add supplementary or to preserve constraints on **A** and **B** as nonnegativity, sparsity and smoothness, leading to generalized NMF methods [11,29,30,53,64].

The hierarchical ALS (HALS) algorithm [11] is an alternative method to ALS based on an optimization of a set of local cost functions, which updates each column of **A** and **B** instead of directly computing the whole matrices at each iterative step. This method is simple and often used for multi-layer models to improve performance; furthermore, it is efficient for large-scale NMF [12,17]. Another fast algorithm in the ANLS framework, referred to as ANLS-BPP, was proposed in [35], and employs the block principal pivoting (BPP) and active set methods [41]. The ANLS-BPP technique can outperform the HALS mainly when the matrix factors are sparse.

It is interesting to remark that the optimization problem given by (1) can also be formulated in terms of the Kullback–Leibler divergence [43,45] or other divergences [8,10,67] instead of the Frobenius norm. A problem that arises from the processing of large-scale or ill-conditioned data is the slow convergence, mainly for the MU methods, and the increase of computational complexity and memory requirements. An efficient way to reduce the complexity and to improve the performance of the NTF/NTD is to include a pre-processing step based on low-rank approximation techniques, as proposed in [69,70].

We present in this paper a novel algorithm for NTF and sparse NTF adapted to higher-order tensor decomposition with one large dimension. Algorithms for NTF presented in the past were often restricted in the size of tensors that can be decomposed. Algorithms designed to overcome this size restriction, for example based on *block wise decomposition*, require a frequent access to partitions of the whole data of the tensor. The presented algorithm in contrary requires only a minimum of data access and is even capable to start a decomposition before the whole tensor is known. In comparative tests the algorithm has proved to be competitive with state of the art algorithms. NsTEF is an incremental algorithm as incremental PCA [65] or INMF proposed by Bucak and Gunsel [4] but it deals with tensors, not with matrices. An important advantage of tensor decompositions over standard matrix approach is the model uniqueness; if it exists, is unique [9], which leads to an interesting benefit of our method.

The rest of the paper is organized as follows: related works are presented in Section 2 where problems encountered using standard NTF algorithms are detailed; Section 3 introduces the proposed algorithm, named NsTEF; we present the experimental results in Section 4; finally, we conclude this paper in Section 6.

Notation

N-th order tensors (for $N \geq 3$), matrices (second-order tensors), vectors (first-order tensors), and scalars (zero-order tensors) are respectively denoted by calligraphic ($\mathcal{A}, \mathcal{B}, \ldots$), boldface uppercase ($\mathbf{A}, \mathbf{B}, \ldots$), boldface lower-case ($\mathbf{a}, \mathbf{b}, \ldots$), and lower-case letters ($a, b, \ldots$). Each element of an N-order tensor $\mathcal{A}$ is denoted by $a_{i_1,i_2,\cdots,i_N}$. A tensor $\mathcal{A}$ is called non-negative if all its elements are non-negative, *i.e.* $a_{i_1,i_2,\cdots,i_N} \geq 0$. For non-negative real tensors we use the short hand notation $\mathcal{A} \geq 0$ and $\mathcal{A} \in \mathbb{R}_+$. $\mathbf{A}_{i_1\cdot\cdot} \in \mathbb{R}^{I_2 \times I_3}$, $\mathbf{A}_{\cdot i_2 \cdot} \in \mathbb{R}^{I_1 \times I_3}$, $\mathbf{A}_{\cdot\cdot i_3} \in \mathbb{R}^{I_1 \times I_2}$ represent the *slices* of a third-order tensor $\mathcal{A}$ constructed by fixing the mode 1, 2, and 3, respectively. Any higher-order tensor can be represented by *matrix unfoldings* from the rearrangement of its elements into a matrix from the *matrix slicings* by fixing one mode. Consider for example a third-order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, we can define three different matrix unfoldings: $\mathbf{A}_{I_1 \times I_2 I_3} \triangleq \left[\mathbf{A}_{\cdot 1\cdot} \ \ldots \ \mathbf{A}_{\cdot I_2 \cdot}\right]$, $\mathbf{A}_{I_2 \times I_3 I_1} \triangleq \left[\mathbf{A}_{\cdot\cdot 1}^{\mathrm{T}} \ \ldots \ \mathbf{A}_{\cdot\cdot I_3}^{\mathrm{T}}\right]$ and $\mathbf{A}_{I_3 \times I_1 I_2} \triangleq \left[\mathbf{A}_{1\cdot\cdot}^{\mathrm{T}} \ \ldots \ \mathbf{A}_{I_1\cdot\cdot}^{\mathrm{T}}\right]$ to represent the same tensor $\mathcal{A}$. By convention, the indexes placed more to the left vary slower and the ones placed more to the right vary faster. The Kronecker, Khatri–Rao, Hadamard and outer products are denoted by $\otimes, \diamond, \bullet$ and $\circ$ respectively. The trace of **A** is denoted by Tr(**A**).

Definition 1. The n-mode product of a tensor $\mathcal{G} \in \mathbb{R}^{I_1 \times \cdots \times I_n \times \cdots \times I_N}$ and a matrix $\mathbf{A} \in \mathbb{R}^{J_n \times I_n}$ is an $(I_1 \times \cdots \times I_{n-1} \times J_n \times I_{n+1} \times \cdots \times I_N)$-tensor given by

$$[\mathcal{G} \times_n \mathbf{A}]_{i_1,\ldots,i_{n-1},j_n,i_{n+1},\ldots,i_N} \triangleq \sum_{i_n=1}^{I_n} g_{i_1,\ldots,i_n,\ldots,i_N} a_{j_n,i_n}, \text{ for all index values.} \quad (3)$$

The n-mode product is a compact form to represent linear transformations involving tensors and (3) can be rewritten in terms of matrix unfoldings by fixing the n-th mode as follows

$$\mathcal{X} = \mathcal{G} \times_n \mathbf{A} \quad \Leftrightarrow \quad \mathbf{X}_{(n)} = \mathbf{AG}_{(n)}, \quad (4)$$

where $\mathbf{X}_{(n)}$ and $\mathbf{G}_{(n)}$ denote the matrix unfolding of $\mathcal{X}$ and $\mathcal{G}$ associated with the n-th mode.

2. Tensor models

Tensor decompositions were first discussed in 1927 by Hitchcock [28]. In the late 1960s tensor decompositions were rediscovered by Tucker [61], Carroll and Chang [5], and Harshman [24] respectively named Tucker decomposition, canonical decomposition (CANDECOMP), and parallel factors analysis (PARAFAC). The two last models, referred to herein as CP, were independently developed in psychometrics and phonetics, however both correspond to the same decomposition and the names report to different features of this model. Tucker model is a general version of the well-known CP model and was also applied in psychometrics. A particular case of this decomposition can be viewed as a multilinear generalization of the singular value decomposition (SVD) for higher-order tensors later introduced by Lathauwer [40]. Tensor decompositions appear today in various fields including image and signal processing, clustering analysis, data compression, blind source separation, direction of arrival estimation, hyperspectral imaging and others [2,51,55,62].

2.1. CANDECOMP/PARAFAC (CP) model

The CP model decomposes a tensor as a minimal sum of rank-one tensors, which can be defined in a concise form and denoted