



Dictionary-based tensor-train sparse coding

Abdelhak Boudehane, Yassine Zniyed, Arthur Tenenhaus, Laurent Le Brusquet, Remy Boyer

► **To cite this version:**

Abdelhak Boudehane, Yassine Zniyed, Arthur Tenenhaus, Laurent Le Brusquet, Remy Boyer. Dictionary-based tensor-train sparse coding. 28th European Signal Processing Conference, EU-SIPCO'20, Jan 2021, Amsterdam, Netherlands. hal-02865345

HAL Id: hal-02865345

<https://hal.univ-lille.fr/hal-02865345>

Submitted on 11 Jun 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

DICTIONARY-BASED TENSOR-TRAIN SPARSE CODING

Abdelhak Boudehane* Yassine Zniyed† Arthur Tenenhaus* Laurent Le Brusquet* Remy Boyer‡

* Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France.

† CRAN, Université de Lorraine, CNRS, Vandoeuvre-lès-Nancy, France

‡ Centre de Recherche en Informatique, Signal et Automatique de Lille, Université de Lille 1, Villeneuve-d'Ascq, France

Abstract—Multidimensional signal processing is receiving a lot of interest recently due to the wide spread appearance of multidimensional signals in different applications of data science. Many of these fields rely on prior knowledge of particular properties, such as sparsity for instance, in order to enhance the performance and the efficiency of the estimation algorithms. However, these multidimensional signals are, often, structured into high-order tensors, where the computational complexity and storage requirements become an issue for growing tensor orders. In this paper, we present a sparse-based Joint dimensionality Reduction And Factors rEtrieval (JIRAFE). More specifically, we assume that an arbitrary factor admits a decomposition into a redundant dictionary coded as a sparse matrix, called the sparse coding matrix. The goal is to estimate the sparse coding matrix in the Tensor-Train model framework.

Index Terms—Sparse coding, High-order tensors, Tensor train, Constrained tensor decomposition, Fast algorithms.

I. INTRODUCTION

Sparse coding has been, for a long time, a common topic in different signal processing fields, such as sound source separation [1], digital sampling [2] and feature extraction [3]. The growing interest of multidimensional signals, such as in sensor array processing [4] and the biomedical field [5], led to the need for efficient frameworks to deal with these particular data structures. The multidimensional signals are, in this work, structured into high-order tensors [6]. The tensor network frameworks developed for high-order tensor decomposition into a graph-connected low order tensors are suitable for applications where the dimensionality/order is an issue, since many methods have been proposed to overcome the so called “curse of the dimensionality” [7]. This problem occurs when the order of the tensor increases, leading to an exponential increase of the number of the tensor’s entries, which means an exponential increase for computational/memory requirements and low convergence rate. Moreover, tensor decomposition models present many beneficial properties, such as uniqueness for the canonical polyadic decomposition (CPD) [8]. Uniqueness means that it exists only one possible set of factors that generates the tensor up to mild ambiguities. However, the estimation problem of the factors is usually done using iterative algorithms such as gradient-based approach in [9], using a non-linear conjugate gradient (NCG) algorithm, or alternating least squares (ALS) in [10] and their variants. These algorithms may encounter some convergence problems,

especially in high-order tensor cases and in the presence of noise [11], where they require a high number of iterations and they present a low convergence rate, in addition to the presence of outliers. When a priori knowledge on the data structure is available, considering structural constraints in the estimation scheme can help to both, reduce the computational cost and help changing the ill-posed estimation problem into a better-posed one [12]. Several constraints can be found in the literature, such as the sparsity [13] or the structured factors constraints[14] to mention a few.

In the sequel, a dictionary parameterized factor constraint is considered. This constraint considers that at least one factor of the CP tensor is parameterized by a linear combination of atoms from a dictionary. A binary column-sparse coding matrix selects the atoms in the dictionary. However, the high-order tensor case is still an open problem, which limits the tensor structures that we may consider. In order to deal with this problem, we propose a methodological approach based on Joint dimensionality Reduction And Factors rEtrieval (JIRAFE) scheme [15]. This scheme is based on the tensor-train (TT) model [16], adding sparsity constraint, to estimate the sparse coding matrix in the case of a high-order tensors.

Notations: Vectors, matrices and tensors are represented by \mathbf{x} , \mathbf{X} and \mathcal{X} , respectively. The symbols $(\cdot)^T$ and $(\cdot)^\dagger$ denote, respectively, the transpose and the Moore–Penrose inverse matrix. The Frobenius norm is defined by $\|\cdot\|_F$. The n -mode product is denoted by \times_n and the tensor contraction by \times_q^p . The tensor $\mathcal{I}_{Q,R}$, is a hypercube with ones on the diagonal and zero otherwise, denotes the identity tensor of size $R \times R \times \dots \times R$. The matrix $\mathbf{X}^{(k_0)}$ of size $N_{k_0} \times N_1 \dots N_{k_0-1} N_{k_0+1} \dots N_Q$ refers to the k_0 -mode unfolding of \mathcal{X} of size $N_1 \times \dots \times N_Q$.

II. DICTIONARY-BASED PARAMETERIZED CANONICAL POLYADIC DECOMPOSITION

A. Canonical polyadic decomposition

A Q -order rank- R CPD tensor $\mathcal{X} \in \mathbb{C}^{N_1 \times N_2 \times \dots \times N_Q}$ is given by (1) as

$$\mathcal{X} = \mathcal{I}_{Q,R} \times_1 \mathbf{P}_1 \times_2 \dots \times_Q \mathbf{P}_Q, \quad (1)$$

where $\mathbf{P}_q \in \mathbb{C}^{N_q \times R}$, for $1 \leq q \leq Q$, are called the factor matrices.

Many methods that are found in the literature, including

alternating least squares, allow the estimation of these factors by minimizing the following objective function

$$f(\mathbf{P}_1, \dots, \mathbf{P}_Q) = \|\mathcal{X} - \mathcal{I}_{Q,R} \times_1 \mathbf{P}_1 \times_2 \dots \times_Q \mathbf{P}_Q\|_F^2. \quad (2)$$

B. Sparse dictionary-parameterized factors

For a fixed index k_0 , the k_0 -th factor \mathbf{P}_{k_0} is dictionary-based, *i.e.*, the columns of \mathbf{P}_{k_0} are obtained by a linear combination of the atoms of a dictionary. It is written as

$$\mathbf{P}_{k_0} = \mathbf{D}\mathbf{S}, \quad (3)$$

where \mathbf{D} is a known over-complete dictionary matrix, of size $N_{k_0} \times M$, whose columns are called the atoms, with $M \gg N_{k_0}$. The sparse matrix \mathbf{S} , of size $M \times R$, is called the selection matrix and it is used to select columns of the factor \mathbf{P}_{k_0} from the dictionary. In the following, we will consider a particular binary structure for \mathbf{S} where only one element is non-zero in each column. Hence, we can write the sparsity constraint on the columns of the sparse coding matrix as $\|\mathbf{s}_i\|_0 = 1$, for $i \in \{1, \dots, R\}$ [17]. The problem in this case, is to determine the support of the sparse coding matrix \mathbf{S} , *i.e.*, the row index of the non-zero elements.

The objective function of this dictionary-based CPD problem is, then, defined as in [17] where the authors propose an objective function by adding the sparsity constraint to the CPD objective function defined in (2). This problem is referred to as Dictionary based CPD (DCPD).

C. Limitations

The problem presented in the previous section encounters convergence problems due to the projection of the parameterized factor on the dictionary [17]. However, the major limitation is witnessed in the high-order tensors case due to the curse of dimensionality. Moreover, in order to estimate the sparse coding matrix \mathbf{S} , the estimation of all the factors \mathbf{P}_q , for $q \in \{1, \dots, Q\}$, is inevitable, which makes the computational and storage cost expensive when only the estimation of the dictionary-parameterized factor is targeted.

In the next section, we will propose a new scheme allowing the estimation of matrix \mathbf{S} , without the need to estimate all the factors, and which is based on a semi-iterative methodology.

III. DICTIONARY-BASED JIRAFE

In this section, we propose a new scheme based on TT decomposition, as a solution to mitigate the curse of dimensionality, using a modified JIRAFE scheme [15, 18] to take into account the sparsity. The JIRAFE framework is composed of two steps. A first dimensionality reduction step that mitigates the original high-order tensor \mathcal{X} into 3 order tensors called TT-cores. In the second step, and given the known structure of the TT-cores, different solutions can be considered to retrieve the factors \mathbf{P}_q . In this paper, we will consider the decomposition of the k_0 -th TT-core and estimate the factor \mathbf{P}_{k_0} that is dictionary-based (*i.e.* estimate the sparse coding matrix \mathbf{S}).

A. Tensor Train model

Let us assume a Q -order CPD tensor \mathcal{X} of canonical rank R , where all the factors \mathbf{P}_q are full column rank. In addition, the k_0 -th factor \mathbf{P}_{k_0} is assumed to be parametrized, *i.e.*, written in terms of a dictionary \mathbf{D} and a sparse matrix \mathbf{S} as in (3).

The equivalence result between the CP model and the TT model in [15] allows us to write the CP tensor \mathcal{X} following a TT format as:

$$\mathcal{X} = \mathbf{G}_1 \times_2 \mathbf{G}_2 \times_3 \dots \times_{k_0} \mathbf{G}_{k_0} \times_{k+1} \dots \times_{Q-1} \mathbf{G}_{Q-1} \times_Q \mathbf{G}_Q,$$

with

$$\begin{aligned} \mathbf{G}_q &= \mathcal{I}_{3,R} \times_1 \mathbf{M}_{q-1} \times_2 \mathbf{P}_q \times_3 \mathbf{M}_q^{-T}, \\ \mathbf{G}_{k_0} &= \mathcal{I}_{3,R} \times_1 \mathbf{M}_{k_0-1} \times_2 \mathbf{D}\mathbf{S} \times_3 \mathbf{M}_{k_0}^{-T}, \\ \mathbf{G}_1 &= \mathbf{P}_1 \mathbf{M}_1^{-1}, \\ \mathbf{G}_Q &= \mathbf{M}_{Q-1} \mathbf{P}_Q^T, \end{aligned}$$

where \mathbf{G}_q is the q -th TT-core tensor of size $R \times N_q \times R$, and R is the TT-rank. The matrices \mathbf{M}_q , of size $R \times R$, are the change-of-basis matrices for $1 \leq q \leq Q-1$ that appear when extracting the dominant singular sub-spaces [19]. The identifiability problem is solved when the matrices \mathbf{M}_q are estimated.

It is to be mentioned that the rank of the TT-cores is equivalent to the canonical rank of tensor \mathcal{X} [15].

B. Proposed scheme

As mentioned in the previous section, the first step of JIRAFE scheme is the TT-decomposition which is done by minimizing the objective function

$$f(\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_Q) = \|\mathcal{X} - \mathbf{G}_1 \times_2 \mathbf{G}_2 \times_3 \dots \times_Q \mathbf{G}_Q\|_F^2,$$

using algorithms such as the tensor-train singular value decomposition (TT-SVD) [16] or the TT hierarchical SVD (TT-HSVD) [19]. As a result we estimate the TT-cores including the k_0 -th TT-core whose decomposition allows us to estimate the k_0 -th factor. Hence, we can define the second step, which is the estimation of the k_0 -th factor from the k_0 -th TT core by minimizing the objective function

$$\begin{aligned} \tilde{f}(\mathbf{P}_{k_0}, \mathbf{S}) &= \|\mathbf{G}_{k_0} - \mathcal{I}_{3,R} \times_1 \mathbf{M}_{k_0-1} \times_2 \mathbf{P}_{k_0} \times_3 \mathbf{M}_{k_0}^{-T}\|_F^2 \\ &\quad + \|\mathbf{P}_{k_0} - \mathbf{D}\mathbf{S}\|_F^2 \\ &\text{subject to } \|\mathbf{s}_i\|_0 = 1. \end{aligned} \quad (4)$$

Processing only the k_0 -th TT-core \mathbf{G}_{k_0} related to the factor \mathbf{P}_{k_0} allows us to mitigate the high Q -order DCPD problem into a smaller/easier 3-order DCPD problem expressed in (4). In fact, the estimation of the selection matrix \mathbf{S} is done in two steps: first we estimate the TT-core \mathbf{G}_{k_0} using a TTD algorithm. Then, we estimate the factor \mathbf{P}_{k_0} using a simple Tri-ALS algorithm [20] in which simultaneous orthogonal

matching pursuit (SOMP) [21] is involved in order to estimate the sparse coding matrix \mathbf{S} using \mathbf{P}_{k_0} and the dictionary matrix \mathbf{D} , as explained in algorithm 1. In our case, since only one element is non zero in each column of the sparse coding matrix ($\|s_i\|_0 = 1$), the SOMP algorithm can be reduced to simply computing the angle between the factor vectors and the dictionary atoms.

Algorithm 1 D-JIRAFE

Input: Tensor \mathcal{X} , dictionary \mathbf{D} and rank R
Output: Sparse coding matrix \mathbf{S}

1: TT-cores estimation
 $[\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_{Q-1}, \mathbf{G}_Q] = \mathbf{TTD}(\mathcal{X}, R)$

2: **Repeat**

Estimation of \mathbf{P}_{k_0} from \mathcal{G}_{k_0} using one **Tri-ALS** iteration as follows

$$\begin{aligned} \mathbf{M}_{k_0-1} &= \mathbf{G}_{k_0}^{(1)}[(\mathbf{M}_{k_0}^{-T} \odot \mathbf{P}_{k_0})^T]^\dagger \\ \mathbf{P}_{k_0} &= \mathbf{G}_{k_0}^{(2)}[(\mathbf{M}_{k_0}^{-T} \odot \mathbf{M}_{k_0-1})^T]^\dagger \\ \mathbf{M}_{k_0}^{-T} &= \mathbf{G}_{k_0}^{(3)}[(\mathbf{P}_{k_0} \odot \mathbf{M}_{k_0-1})^T]^\dagger \end{aligned}$$

Estimation of matrix \mathbf{S}

$$\mathbf{S} = \mathbf{SOMP}(\mathbf{D}, \mathbf{P}_{k_0})$$

Update \mathbf{P}_{k_0} using the estimated \mathbf{S}

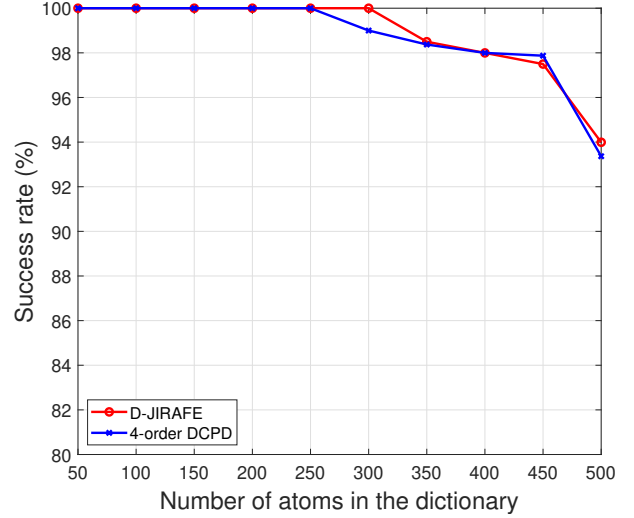
$$\mathbf{P}_{k_0} = \mathbf{D}\mathbf{S}$$

Until fit ceases to improve or maximum iterations exhausted

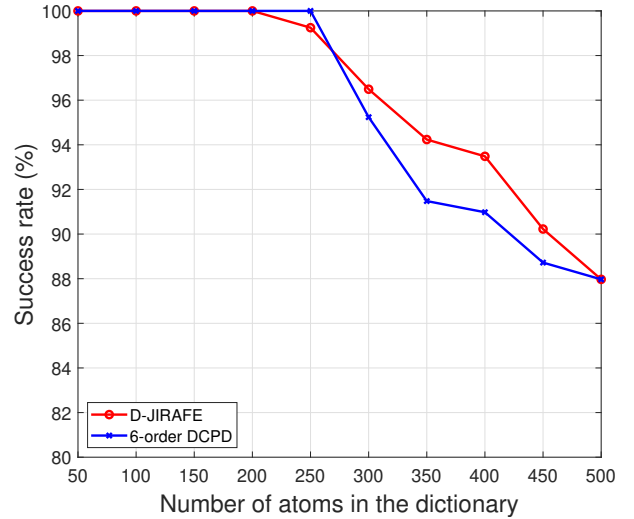
For the applications where only the dictionary-based factor is targeted, algorithm 1 is sufficient. However, in order to estimate the rest of the factors, we take advantage of the coupling between the TT-cores [22], *i.e.*, the change of basis matrix \mathbf{M}_{k_0-1} is used with the TT-core \mathcal{G}_{k_0-1} in a Bi-ALS in order to estimate the factor \mathbf{P}_{k_0-1} . The Bi-ALS is a 3-order ALS to decompose a 3-order tensor but with a known (pre-estimated) matrix. It is to be mentioned that a closed-form solution could be used instead of Bi-ALS [15]. In the same way, the matrix \mathbf{M}_{k_0} is used to estimate \mathbf{P}_{k_0+1} in parallel. Hence, the parallel estimation of the factors on the right and left side is done by using the change-of-base matrices estimated in the previous steps as shown in Fig. 1.

IV. NUMERICAL RESULTS AND DISCUSSION

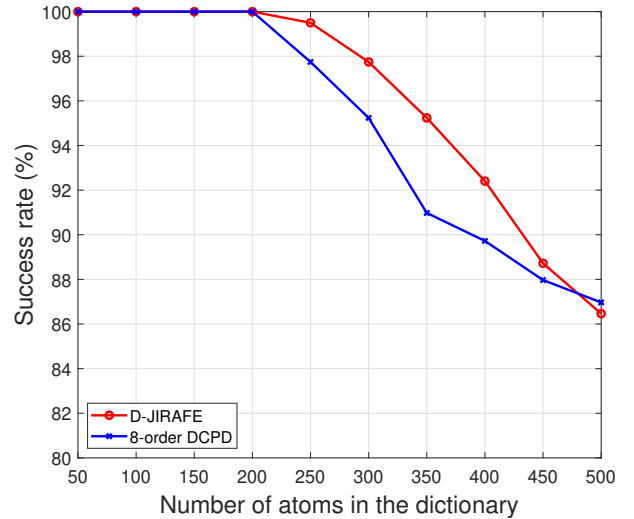
In this section, we will present the numerical results obtained by applying D-JIRAFE on 4-order, 6-order and 8-order tensors and we compare it to the DCPD [17]. The dimensions of the tensor are $N_q = 10$ for $q \in \{1, \dots, k_0 - 1, k_0 + 1, \dots, Q\}$, while $N_{k_0} = 30$. The canonical rank of the tensor is 3. The simulations were performed on a computer equipped with an *Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz (8 CPUs)*, 2.1GHz processor and 32Gb RAM.



(a) Order 4



(b) Order 6



(c) Order 8

Fig. 2: Success rate (%) in function of the number of atoms in the dictionary (SNR = 20dB)

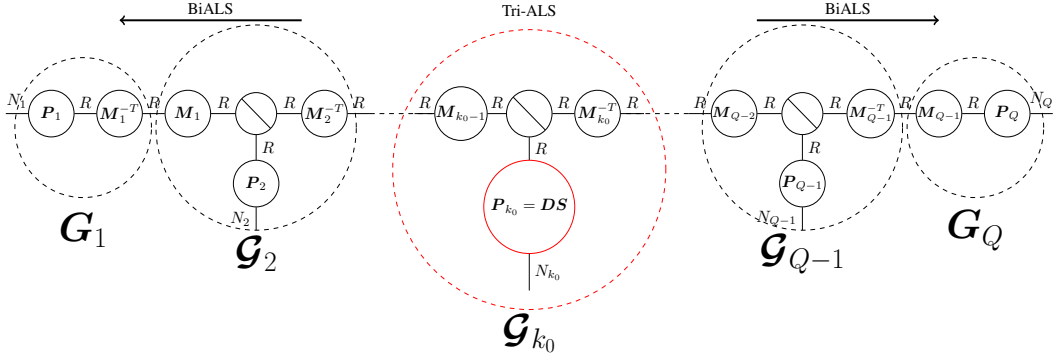


Fig. 1: The graph modelization in this figure represents the TT-model, where each bow represents a dimension. Hence, the nodes that are connected with 3 bows are 3-d identity tensors, while the nodes connected with only 2 bows are the matrices. The dashed lines encircle the 3-d TT-core tensors.

TABLE I: Execution time in function of the number of atoms in the dictionary (order 6, SNR=20dB)

N# of atoms	D-JIRAFE $\times 10^{-3}$	DCPD $\times 10^{-3}$	Gain
50	10.06 (s)	67.18 (s)	6.67
250	12.62 (s)	75.6 (s)	5.99
500	12.38 (s)	76.75 (s)	6.2

TABLE II: Execution time in function of the order for 250 atoms in the dictionary (SNR=20dB)

Tensor order	D-JIRAFE $\times 10^{-3}$	DCPD $\times 10^{-3}$	Gain
4	5.20 (s)	7.96 (s)	1.53
6	12.62 (s)	75.6 (s)	5.99
8	222.2 (s)	5151.6 (s)	23.18

Following the frameworks in [6, 17], we use a harmonic dictionary D ($N_{k_0} \times M$) containing the modes $\{\mathbf{d}_1, \dots, \mathbf{d}_M\}$ with

$$\mathbf{d}_m = [1, \exp\{j2\pi f_m\}, \dots, (\exp\{j2\pi f_m\})^{N_{k_0}-1}]^T,$$

and $f_m \in (0, 1)$ is the m -th frequency. We vary the size of the dictionary (number of atoms) from 50 to 500 with a step of 50. The sparse coding matrix is column-sparse and binary valued, while the non-parameterized factor matrices follow a standard complex normal distribution. White Gaussian noise is, then, added in order to have a signal to noise ratio of 20dB and the results are generated from 1000 trials. For the simulations, we use the Tensorlab package for Matlab [23]. We compare the two frameworks in terms of success rate, *i.e.*, the number of the trials in which the framework succeed to estimate the exact support on the total number of trials, as well as the execution time where we define a gain by dividing the execution time of DCPD on the execution time of D-JIRAFE.

As we can see in Fig. 2, the success rate of our approach and the DCPD is identical for small number of atoms. In fact, the variation remains too small and insignificant when we increase the number of atoms in the dictionary. Thereby,

when we increase the order from 4 to 8, the success rate does not show any significant changing. A slight decrease is seen in both algorithms as we increase the order. In fact, this is due to the appearance of some outliers as the order increases. On the other hand, Tab. I shows the execution time of the DCPD and D-JIRAFE in function of the number of atoms in the case of a 6-order tensor. The ratio of the execution time between the two methods remains invariant when we increase the number of atoms. However, Tab. II shows that, as the order of the tensor increases, the ratio of the execution time increases in a very significant way in the favor of D-JIRAFE.

V. CONCLUSION

Sparse coding is a common signal processing topic that founds applications in many fields. Nevertheless, the multidimensional signals presented by the recent application fields has unveiled many challenges, as data structures are becoming more massive in terms of dimensions and order. In this paper, we presented a sparse-coding problem in the context of high-order tensors. We have shown it was necessary to estimate all the factors in order to estimate the selection support of a sparse factor, which is computationally expensive. Thus, we proposed a scheme in order to cope the problem of dimensionality, in this case, by mitigating the high order tensor into a 3-order tensors called the TT-cores. As a result, the estimation of the sparse factor becomes a simple 3-order tensor problem, considering the TT-core relative to the sparse factor. The simulation results show that, as we increase the order, the ratio in terms of execution time, between our approach, and the approach that we took as a reference, increases in a significant way. This makes it possible to deal with high-order tensors with a lower computational complexity.

REFERENCES

- [1] T. Virtanen, "Separation of sound sources by convolutive sparse coding," *ISCA Tutorial and Research Workshop (ITRW) on Statistical and Perceptual Audio Processing*, 2004.

- [2] R. Rubinstein, A. M. Bruckstein, and M. Elad, "Dictionaries for sparse representation modeling," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 1045–1057, 2010.
- [3] H. Liu, C. Liu, and Y. Huang, "Adaptive feature extraction using sparse coding for machinery fault diagnosis," *Mechanical Systems and Signal Processing*, vol. 25, no. 2, pp. 558 – 574, 2011.
- [4] M. Viberg and B. Ottersten, "Sensor array processing based on subspace fitting," *IEEE Transactions on Signal Processing*, vol. 39, no. 5, pp. 1110–1121, 1991.
- [5] R. Cárdenes, R. de Luis-García, and M. Bach-Cuadra, "A multidimensional segmentation evaluation for medical image data," *Computer Methods and Programs in Biomedicine*, vol. 96, no. 2, pp. 108 – 124, 2009.
- [6] S. Sahnoun, E. H. Djermoune, D. Brie, and P. Comon, "A simultaneous sparse approximation method for multi-dimensional harmonic retrieval," *Signal Processing*, vol. 131, pp. 36 – 48, 2017.
- [7] A. Cichocki, N. Lee, I. Oseledets, A. Phan, Q. Zhao, D. P. Mandic *et al.*, *Tensor networks for dimensionality reduction and large-scale optimization: Part 1 low-rank tensor decompositions*. Now Publishers, Inc., 2016, vol. 9, no. 4-5.
- [8] R. A. Harshman, "Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis," *UCLA Working Papers in Phonetics*, vol. 16, pp. 1–84, 1970.
- [9] E. Acar, T. G. Kolda, and D. M. Dunlavy, "All-at-once Optimization for Coupled Matrix and Tensor Factorizations," in *9th Workshop on Mining and Learning with Graphs, San Diego, CA*, 2011.
- [10] S. Bahargam and E. E. Papalexakis, "A constrained coupled matrix-tensor factorization for learning time-evolving and emerging topics," *CoRR*, vol. abs/1807.00122, 2018.
- [11] V. de Silva and L. H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM J. Matrix Analysis Applications*, vol. 30, pp. 1084–1127, 2006.
- [12] N. Vervliet, O. Debals, and L. De Lathauwer, "Exploiting efficient representations in large-scale tensor decompositions," *SIAM Journal on Scientific Computing*, vol. 41, no. 2, pp. A789–A815, 2019.
- [13] S. Zubair and W. Wang, "Tensor dictionary learning with sparse tucker decomposition," in *2013 18th International Conference on Digital Signal Processing (DSP)*, 2013, pp. 1–6.
- [14] A. Koochakzadeh, P. Sarangi, and P. Pal, "Mixed factor structured tensor decomposition via solving quadratic equations," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, 2018, pp. 647–651.
- [15] Y. Zniyed, R. Boyer, A. L. F. de Almeida, and G. Favier, "High-order tensor estimation via trains of coupled third-order CP and Tucker decompositions," *Linear Algebra and its Applications*, vol. 588, pp. 304–337, 2020.
- [16] I. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.
- [17] J. E. Cohen and N. Gillis, "Dictionary-based tensor canonical polyadic decomposition," *IEEE Transactions on Signal Processing, Institute of Electrical and Electronics Engineers*, pp. 66 (7), pp.1876–1889., 2018.
- [18] Y. Zniyed, R. Boyer, A. L. F. de Almeida, and G. Favier, "High-order CPD estimator with dimensionality reduction using a tensor train model," *26th European Signal Processing Conference (EUSIPCO)*, pp. 2613–2617, 2018.
- [19] Y. Zniyed, R. Boyer, A. de Almeida, and G. Favier, "A tt-based hierarchical framework for decomposing high-order tensors," *to appear in SIAM Journal on Scientific Computing*, 2020.
- [20] H. A. Kiers, "An alternating least squares algorithm for parafac2 and three-way dedicom," *Computational Statistics & Data Analysis*, vol. 16, no. 1, pp. 103–118, 1993.
- [21] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Simultaneous sparse approximation via greedy pursuit," *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5, pp. v/721–v/724 Vol. 5, 2005.
- [22] A. Boudehane, Y. Zniyed, A. Tenenhaus, L. L. Brusquet, and R. Boyer, "Breaking the curse of dimensionality for coupled matrix-tensorfactorization," *CAMSAP*, 2019.
- [23] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab 3.0," <https://www.tensorlab.net>, 2016.