Identifying Stable Components of Matrix/Tensor Factorizations via Low-Rank Approximation of Inter-Factorization Similarity

Simon Van Eyndhoven*†, Nico Vervliet*, Lieven De Lathauwer*‡ and Sabine Van Huffel*†
*KU Leuven, Department of Electrical Engineering (ESAT), STADIUS Center for Dynamical Systems, Signal Processing and Data Analytics, Kasteelpark Arenberg 10, 3001 Leuven, Belgium
†imec, Leuven, Belgium ‡ KU Leuven Kulak, Group Science, Engineering and Technology, 8500 Kortrijk, Belgium
Email: simon.vaneyndhoven@esat.kuleuven.be

***Abstract*—Many interesting matrix decompositions / factorizations, and especially many tensor decompositions, have to be solved by non-convex optimization-based algorithms, that may converge to local optima. Hence, when interpretability of the components is a requirement, practitioners have to compute the decomposition (e.g. CPD) many times, with different initializations, to verify whether the components are reproducible over repetitions of the optimization. However, it is non-trivial to assess such reliability or stability when multiple local optima are encountered. We propose an efficient algorithm that clusters the different repetitions of the decomposition according to the local optimum that they belong to, offering a diagnostic tool to practitioners. Our algorithm employs a graph-based representation of the decomposition, in which every repetition corresponds to a node, and similarities between components are encoded as edges. Clustering is then performed by exploiting a property known as *cycle consistency*, leading to a low-rank approximation of the graph. We demonstrate the applicability of our method on realistic electroencephalographic (EEG) data and synthetic data.**

I. Introduction

Decompositions or factorizations of matrices or tensors (multiway arrays) into a number of components are useful in a variety of signal processing, data mining and machine learning applications, in which the matrix or tensor represents a certain multivariate signal or dataset. Matrix decompositions such as non-negative matrix factorization (NMF) or independent component analysis (ICA) have attracted a lot of interest in the past decades and are used in numerous applications of telecommunications, biomedical signal processing, blind source separation (BSS), exploratory data analysis, etc. [1]. Tensors, which are generalizations of matrices to higher orders, have also found their way into these domains [2]. Several extensions of matrix decompositions to higher order exist, such as the (non-negative) canonical polyadic decomposition (CPD) and the $(L_r, L_r, 1)$-decomposition [2], [3]. Coupled matrix/matrix, matrix/tensor and tensor/tensor decompositions are used in data fusion, i.e., when multiple datasets which are similar in at least one mode are available [1], [4]. In many cases, decompositions of (noisy) data have no closed-form solution and have to be computed iteratively, i.e., by means of an optimization algorithm. Whereas algebraic algorithms exist for various decompositions, they can only be used when the data exactly follows the imposed structure, or to initialize an optimization algorithm (see e.g. [5] and references therein). In many of the discussed examples (e.g. ICA, NMF, CPD), the optimization problem is non-convex and only convergence to a local optimum can be guaranteed, e.g. when minimizing the residual between the observed data and a factor model thereof (CPD, NMF), or when maximizing a measure of independence between component time courses (ICA). This can pose a problem if one wants to interpret the components of the decomposition, since these components may vary substantially between local optima, even though the associated values of the optimized cost function need not be significantly different. Furthermore, in some applications, because of a considerable influence by noise and/or artifacts, the optimum with the lowest cost value is not guaranteed to yield the most meaningful result. Consider, for example, the computation of a CPD of tensor-valued data that are heavily corrupted by noise and/or artifacts. When fitting the CPD model by minimizing the mean squared error, one or more of the CPD components may model variance that is due to the noise or strong artifacts, neglecting 'true' sources of interest that remain undiscovered. In this case, another solution, in which the artifacts are fully captured in the residual, and where all CPD components model activity of interest, may attain a worse cost value, but may be much more interpretable and meaningful.

From the problem statement above, we distill a practical need to assess the reproducibility of (non-convex) matrix or tensor decompositions. In the current context, this means accurately clustering repetitions of the associated optimization problems in terms of their locally optimal solutions for further interpretation.

A well-known method that attempts such clustering is ICASSO [6], which was designed to assess the reproducibility of ICA decompositions, but whose concepts may be used for other matrix or tensor factorizations as well, e.g. as in [7]. The usefulness of this method depends on the user-defined number of clusters, which can be difficult to estimate if the

The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013) / ERC Advanced Grant: BIOTENSORS (n° 339804). This paper reflects only the authors' views and the Union is not liable for any use that may be made of the contained information. This work was supported by the Fonds de la Recherche Scientifique – FNRS and the Fonds Wetenschappelijk Onderzoek – Vlaanderen under EOS Project n° 30468160 (SeLMA). NV is supported by Internal Funds KU Leuven (PDM/18/146).

factorization is not very stable.

Related approaches exist in non-signal processing-related contexts. A parallel can be drawn between assessing reproducibility of non-convex decompositions and establishing correspondences (maps or transformations) between (points on) multiple objects. The latter problem appears in many applications, e.g., fusing partially overlapping images of a certain scene [8], extracting structure from motion [9], finding dense correspondences between collections of 3D shapes [10], etc. E.g., for the case of image fusion, a link must be established between points in two images if these correspond to the same physical point. Similarly, we aim to identify whether a component appears in two distinct repetitions of the decomposition. While it is feasible to estimate *local* (pairwise) maps between objects, it is not trivial to extract the *global* information (over all repetitions) on the correspondences, e.g. because pairwise maps may be noisy and may lead to ambiguities, especially when similar points on the same objects are present [11]. However, starting from a collection of noisy pairwise maps, techniques known as joint object matching or map synchronization may exploit constraints to retrieve corrected maps that reveal the global structure [12], [13]. Several algorithms have been devised for this objective, of which most treat the problem in a graph framework: objects are treated as nodes, and the input pairwise maps are transformations residing on the edges between them [10]–[13]. In this paper, we propose to use map synchronization principles to identify reliable components of the CPD and assess which components co-occur frequently. Under this setting, the 'right' number of clusters appears naturally from the problem representation and synchronization constraints, fostering a correct interpretation of the components. We demonstrate our technique on the decomposition of electroencephalographic (EEG) data of multiple subjects, and compare its performance to that of ICASSO.

II. Methods

A. (Non-convex) matrix and tensor factorizations

We denote scalars, vectors, matrices and tensors by lower case (e.g. a), lower case boldface (e.g. $\mathbf{a}$), upper case boldface (e.g. $\mathbf{A}$) and bold upper case calligraphic letters (e.g. $\mathcal{A}$), respectively. An Mth order tensor $\mathcal{A} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_M}$ is a multiway array which holds data varying over M modes (e.g. sensors, time points, frequencies, ...) with dimensions I_1, I_2, ... , I_M, respectively. A tensor may be (approximately) decomposed or factorized as a sum of rank-1 terms, where every rank-1 term is an outer product ($\otimes$) of M vectors, which is known as the parallel factor analysis (PARAFAC) or CPD model [1]–[3]. For an Mth order tensor $\boldsymbol{\mathcal{T}}$, this gives:

$$\boldsymbol{\mathcal{T}} = \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \mathbf{a}_r^{(2)} \otimes \ldots \otimes \mathbf{a}_r^{(M)} + \boldsymbol{\mathcal{E}}_x \,, \tag{1}$$

where any column vector $\mathbf{a}_r^{(m)} \in \mathbb{R}^{I_n}$ is the mode-m factor vector of the rth component or term, and $\boldsymbol{\mathcal{E}}_x$ is a residual tensor. As there is no closed-form solution for the factor matrices $\mathbf{A}^{(m)} = \left[\mathbf{a}_1^{(m)}\, \mathbf{a}_2^{(m)} \ldots \mathbf{a}_R^{(m)}\right]$, $m = 1 \ldots M$ that minimize the Frobenius norm $\|\boldsymbol{\mathcal{E}}_x\|_F^2$ in (1), optimization algorithms are commonly used to iteratively update a cost function, e.g. of the form

$$J(\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(M)}) = \left\| \boldsymbol{\mathcal{T}} - \sum_{r=1}^{R} \mathbf{a}_r^{(1)} \otimes \ldots \otimes \mathbf{a}_r^{(M)} \right\|_F^2 \tag{2}$$

In data fusion, multiple tensors and/or matrices are decomposed simultaneously, and the cost function is a linear combination of fit terms similar to those in (2) [1], [4], [14]. As discussed in section I, many cost functions for tensor and matrix decompositions are non-convex, and hence the optimization algorithm may converge to different local minima, depending on the initialization for the factor matrices. When the components' factor vectors $\mathbf{a}_r^{(m)}$ are to be interpreted, e.g. in BSS, it is therefore wise to repeat the optimization several times with different initializations, and verify which components seem stable over several runs (and therefore interpretable) and which seem be specific to a local solution.

B. Inter-factorization similarity as a graph

To assess the reproducibility or stability of a factorization (in R terms), we need to investigate the similarity of components across all repetitions of the decomposition with different initialization, up to permutation and scaling. A graph representation can accommodate for such similarity relationships. Following the notation in [11], we treat every repetition $n = 1 \ldots N$ of the decomposition as an object $S_n \in \mathcal{S}$ with R points. An observation graph $\mathcal{G} = (\mathcal{S}, \mathcal{E})$ can then be constructed by connecting any two repetitions S_i and S_j by an edge $(i, j) \in \mathcal{E}$ if at least one component is shared between them. We consider the mapping $S_j \mapsto S_i$ on the edge (i, j) as an assignment matrix $\mathbf{X}_{ij} \in \{0, 1\}^{R \times R}$, which indicates which components of S_j match which components of S_i[1]. The entire graph can then be represented by a block matrix $\mathbf{X} \in \{0, 1\}^{NR \times NR}$ as follows:

$$\mathbf{X} = \begin{bmatrix} \mathbf{I} & \mathbf{X}_{12} & \cdots & \mathbf{X}_{1N} \\ \mathbf{X}_{12}^{\mathrm{T}} & \mathbf{I} & \cdots & \mathbf{X}_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{X}_{1N}^{\mathrm{T}} & \mathbf{X}_{2N}^{\mathrm{T}} & \cdots & \mathbf{I} \end{bmatrix} \tag{3}$$

The submatrices $\mathbf{X}_{ij}$ encode local, pairwise correspondence. If $(i', j') \notin \mathcal{E}$, i.e., no component is shared between $S_{i'}$ and $S_{j'}$, $\mathbf{X}_{i'j'}$ is an all-zero matrix in (3). Note that composite maps, e.g. $S_k \mapsto S_j \mapsto S_i$ can be constructed by matrix multiplication:

$$\mathbf{X}_{ik} = \mathbf{X}_{ij}\mathbf{X}_{jk} \tag{4}$$

$$= \mathbf{X}_{ij}\mathbf{X}_{kj}^{\mathrm{T}} \tag{5}$$

[1]If the reproducibility between decompositions of varying rank is of interest, the transformation matrices are described by non-square matrices $\mathbf{X}_{ij} \in \{0, 1\}^{|S_i| \times |S_j|}$. This does not alter the remainder of the method.

C. Repeated decompositions exhibit cycle consistency

If the optimization routine converges to the same local optimum in several repetitions, the associated maps are permutation matrices. Together with (5), this immediately leads to the following key observation: composite maps $S_i \mapsto S_j \mapsto S_k \mapsto \cdots \mapsto S_i$ along cycles in the graph should be equal to the identity map. Such a *cycle consistency* property can be exploited as a constraint to correct noisy input maps $\mathbf{X}_{ij}$, a technique which is known as map synchronization [10]–[13]. We now aim to offer an intuition for this consistency from three different perspectives. Firstly, note that in a valid graph representation as in (3), consistency holds for *any* cycle that can be constructed along the edges in $\mathcal{E}$, and as such it constitutes in fact a quite restrictive set of constraints of the form $\mathbf{X}_{i\ldots} \cdots \mathbf{X}_{kj}\mathbf{X}_{ji} = \mathbf{I}$ [10]. Secondly, consider the factor matrices $\mathbf{A}_n^{(m)}$ of an arbitrary mode m of all repetitions n, and stack these in a fat matrix $\mathbf{A}^{(m)} = \begin{bmatrix}\mathbf{A}_1^{(m)} \mathbf{A}_2^{(m)} \ldots \mathbf{A}_N^{(m)}\end{bmatrix}$. A component that appears in multiple repetitions $n_1 \subset n$ then contributes columns, which are the same (up to scaling ambiguity), to the associated $\mathbf{A}_{n_1}^{(m)}$. It is then clear that $\text{rank}(\mathbf{A}^{(m)}) = \tilde{R} \geqslant R$, where $\tilde{R}$ is the number of distinct components that are found in all repetitions n of the decomposition. This establishes a link between cycle consistency and low-rank structure. Lastly, consider that all repetitions S_i are (partial) instances of a universe S_u holding all $\tilde{R}$ distinct components as 'templates' [11]. Using (4)–(5), we can write any map $S_j \mapsto S_i$ as a composite map with the universe as a hub, such that $\mathbf{X}_{ij} = \mathbf{X}_{iu}\mathbf{X}_{uj} = \mathbf{X}_{iu}\mathbf{X}_{ju}^{\mathrm{T}}, \forall (i,j) \in \mathcal{E}$. This reveals that the inter-factorization similarity matrix $\mathbf{X}$ admits a positive semi-definite rank-$\tilde{R}$ decomposition [10]–[12]:

$$\mathbf{X} = \mathbf{X}_u\mathbf{X}_u^{\mathrm{T}} = \begin{bmatrix}\mathbf{X}_{1u} \\ \mathbf{X}_{2u} \\ \vdots \\ \mathbf{X}_{Nu}\end{bmatrix} \begin{bmatrix}\mathbf{X}_{1u}^{\mathrm{T}} & \mathbf{X}_{2u}^{\mathrm{T}} & \cdots & \mathbf{X}_{Nu}^{\mathrm{T}}\end{bmatrix} \quad (6)$$

In realistic scenarios, ambiguities and noise in the (local) input maps $\mathbf{X}_{ij}$ can destroy the (global) cycle consistency and hence the low-rankness in (6) is only approximate. Our goal is now to retrieve $\mathbf{X}_u$ from the imperfect inter-factorization similarity matrix $\mathbf{X}$, in order to assign each individual factorization's component to the correct universal component in S_u and assess the decomposition's reproducibility.

D. Clustering through low-rank graph approximation

For a valid (i.e. noiseless) graph encoding, every row of the block $\mathbf{X}_{iu}$ (corresponding to a component of factorization i) consists of all zeros except for a one at the index of the universal component to which that row is assigned. All rows that are assigned to the same universal component are then equal and form a 'clique'. Hence, if we can find an estimate $\hat{\mathbf{X}}_u$ of $\mathbf{X}_u$, up to an orthogonal transformation, we can cluster its rows in order to categorize the components [12]. As in [12], [13], we estimate $\mathbf{X}_u$ as the leading $\tilde{R}$ eigenvectors of $\mathbf{X}$, weighted by the square root of their eigenvalues, and propose to use hierarchical clustering based on average linkage, a strategy which is also used in ICASSO [6]. The number of distinct components $\tilde{R}$, which also equals the dimension of the resulting feature space, can be found by truncating at the largest relative drop $\frac{\lambda_r - \lambda_{r+1}}{|\lambda_r| + |\lambda_{r+1}|}$ in the eigenvalue spectrum [12]. To further investigate which components often co-occur in the local solutions, we characterize each of the $r = 1 \ldots \tilde{R}$ estimated cliques by means of the set $\mathcal{S}^{(r)}$ of objects who contribute a component to the clique. The object overlap between any two cliques p and q can then be measured by the Dice coefficient $d(p,q) = 2\,\frac{|\mathcal{S}^{(p)} \cap \mathcal{S}^{(q)}|}{|\mathcal{S}^{(p)}| + |\mathcal{S}^{(q)}|}$. The $\tilde{R} \times \tilde{R}$ overlap matrix can then be used for a second stage of average linkage clustering, to find groups of cliques that share many objects. These groups indicate which components are often simultaneously present in repetitions of the optimization.

III. Experiments

We conducted two case studies, based on real data and synthetic data, to evaluate our proposed method. In both cases, we compute the similarity between the rth component of the ith run and the sth component of the jth run as

$$\sigma(r_i, s_j) = \prod_{m \in \mathcal{M}_\sigma}^{M} \sigma_m(r_i, s_j) = \left| \prod_{m \in \mathcal{M}_\sigma}^{M} \langle \mathbf{a}_{r_i}^{(m)}, \mathbf{a}_{s_j}^{(m)} \rangle \right| \quad (7)$$

Here, the factor vectors have been normalized to unit norm, and hence $\sigma_m(r_i, s_j)$ is the mode-m cosine similarity metric [3]. The product of mode-wise similarities is taken over $\mathcal{M}_\sigma$, which is the set of all modes in which component reproducibility is expected/desired[2]. Finally, we binarize the similarities using a threshold of $0.95^{|\mathcal{M}_\sigma|}$ and populate all matrices $\mathbf{X}_{ij}$. All decompositions are computed using state-of-the-art Gauss-Newton type algorithms in Tensorlab [15].

A. CPD of neuroimaging data

We analyzed multi-subject electroencephalographic (EEG) data, recorded in an MR scanner, using a pipeline similar to that of [7]. For every subject $s = 1 \ldots 12$, resting state EEG data from 15 electrodes was converted into a spectrogram $\mathcal{T}^{(s)} \in \mathbb{R}^{15 \times 40 \times 1350}$, evaluated at 40 1-Hz frequency bins and 1350 1-s windows. We wanted to extract reliable spatial and spectral signatures of modulated resting state network (RSN) activity [7]. In order to find these signatures, we sliced every $\mathcal{T}^{(s)}$ halfway along the third mode and stacked either half into two large tensors $\mathcal{T}^{\text{begin}}, \mathcal{T}^{\text{end}} \in \mathbb{R}^{15 \times 40 \times (12 \cdot \frac{1350}{2})}$, which we normalized as in [7]. By decomposing $\mathcal{T}^{\text{begin}}$ and $\mathcal{T}^{\text{end}}$ separately, we expected to find components that were common and may model true RSN modes, but also components that modeled data-specific fluctuations in either half [3], [7] that may be less interesting. We computed a rank-10 CPD of both halves 100 times and applied our algorithm to cluster the resulting components in the spatial and spectral mode (between the halves, no temporal correspondence is expected). We used `cpd_nls` [15] and initialized all factor matrices

[2]E.g. in the case of the $(L_r, L_r, 1)$-decomposition, a rotational ambiguity in the first two modes remains. Hence, similarities may not be computed for these modes separately, but for entire frontal slices, 'absorbing' the ambiguity.

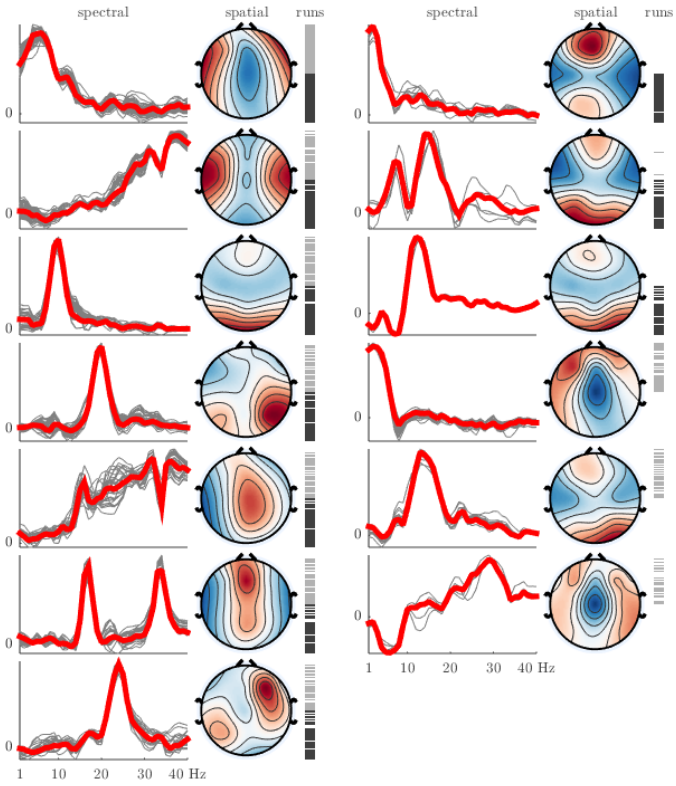


Fig. 1. Our algorithm successfully clusters spatio-spectral components that appear in nearly all runs of the decomposition and are hence reliable (left column), but also distinguishes components that are specific to the decomposition of each half (light/dark grey) of the data (right column). The mean spectra are superimposed in red. The variability of spatial signatures around their mean (plotted) was comparably small as that of the spectra.

using i.i.d. Gaussian variables [15]. We found that $\tilde{R} = 275$, though most of the spectral energy was concentrated in the first few eigenvalues of $\mathbf{X}$, i.e., many of these individual templates appeared only in a few runs, and a small set of templates accounted for most of the runs' components. As shown in Fig.1, we found a subset of seven components that were encountered in nearly every run (left column), and two subsets of three components each, which were only found in $\mathcal{T}^{\text{begin}}$ or $\mathcal{T}^{\text{end}}$, respectively (right column). From this split-half experiment, we may conclude that the latter two subsets of components are less reliable, in the sense that they seem specific to a data segment, whereas the common components may be further inspected. E.g., in the third component we can identify typical alpha (10 Hz) power increases in the occipital regions, at the back of the head. Note, however, that some common components need not be RSN-related: e.g. the sixth component seems to capture two harmonics that may be due to residual MR scanner artifacts.

We also attempted to cluster the components directly based on $\mathbf{X}$, as in ICASSO. Since ICASSO leaves the choice of number of clusters to the user, we varied this number (which we will also denote by $\tilde{R}$) between $\tilde{R} = R = 10$ (i.e., assuming that every run of the decomposition will return the same components) and $\tilde{R} = 400$, in increments of 10. To evaluate the clustering performance, we kept track of 1) the

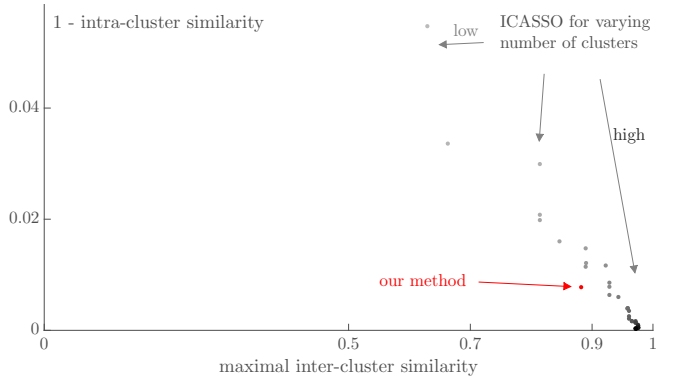


Fig. 2. When grouping EEG tensor decomposition components in clusters, our algorithm realizes a favourable tradeoff between high intra-cluster similarity and low maximal inter-cluster similarity. This is thanks to the proper (automatic) estimation of the number of clusters $\tilde{R}$ (darker is higher), in an $\tilde{R}$-dimensional component space, which is not possible with ICASSO.

average similarity σ of components within the same cluster and 2) the maximal average similarity σ between all components of a cluster and all components of another cluster. These two metrics can respectively indicate (per cluster) whether a cluster contains too much variability and would better be split in several smaller, more compact clusters, and whether two clusters are too similar and would better be merged. In Fig.2, we set out the median of these values (over clusters) against each other (closer to the origin is better). By means of the low-rank graph framework, our proposed method automatically achieves a good operating point, where the components within every cluster are mutually very similar, and the undesired inter-cluster similarity is lower than for most choices of the hyperparameter $\tilde{R}$ in ICASSO. For low values of $\tilde{R}$, the inevitable variability of the components over runs was neglected, and dissimilar components were wrongly grouped together. On the other hand, when $\tilde{R}$ was overestimated, coherent clusters were wrongfully split into smaller clusters. For a few choices of $\tilde{R}$ (ca. 60–100), ICASSO found the same division in (main) clusters as in Fig.1, though this required much more computation and user interaction due to the iterative nature of the tuning procedure.

B. Coupled matrix-tensor factorization of synthetic data

We generated synthetic multimodal data according to a coupled matrix-tensor factorization model (CMTF) [14]. In this case, $\mathcal{M}_\sigma$ comprises the modes of both the tensor *and* the matrix. For the tensor $\mathcal{T} \in \mathbb{R}^{20\times100\times40}$, factor matrices of rank 7 were sampled from an exponential distribution. All columns of the mode-1 factor matrix were shared with the matrix $\mathbf{M} \in \mathbb{R}^{20\times200}$, which also contained an unshared rank-1 term. The mode-2 factor matrix of $\mathbf{M}$ was generated by AR(1)-model with coefficient 0.9 and white gaussian noise as innovations. We added exponentially distributed and white gaussian noise to both datasets, respectively, with an SNR of 0 dB. Using `sdf_nls` [4], [15], we computed a CMTF of the data with five components 100 times, with non-negativity constraints on the factor matrices of $\mathcal{T}$. Each time, $\mathcal{T}$'s factor matrices were initialized as random factor matrices, uniformly

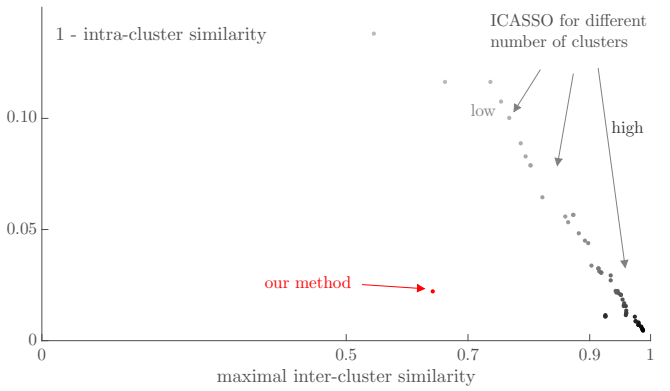


Fig. 3. Also on synthetic multimodal data, the proposed method successfully groups components of a coupled matrix-tensor factorization in clusters which are more coherent than those of ICASSO.

distributed between 0 and 1, after which $\mathbf{M}$'s second factor matrix was initialized with a pseudo-inverse. We wish to look at the effect of mismodeling the data, i.e., underestimating the number of of components (five instead of seven). The $(5 \cdot 100) \times (5 \cdot 100)$ similarity matrix $\mathbf{X}$ had an estimated rank $\tilde{R} = 191$, indicating that the variability was relatively high (approximately 191 distinct components out of 500 in total). Nevertheless, a small set of seven components appeared relatively frequently (at least in five runs). For ICASSO, we varied the number of clusters from 5 to 300, in steps of 5. In comparison of both methods, a similar pattern of intra- and inter-cluster similarity as in the previous experiment emerged (see Fig.3). The proposed method again finds (much) more coherent clusters than ICASSO, i.e., with a low inter-cluster similarity and a high intra-cluster similarity. We want to stress that 'successful clustering' is not a guarantee of the truthfulness of the found components. We observed that several ground-truth components were almost never captured in an estimated component of a CMTF run. On the other hand, the ground-truth components that were sucessfully found in some repetitions (with a relatively high similarity of approximately 0.5) were most of the time successfully clustered over repetitions.

IV. Discussion

We have proposed a method to effectively assess the stability of components of coupled or uncoupled matrix and tensor factorizations, in case a non-convex optimization algorithm has to be repeated with multiple initializations. Inspired by earlier work in the area of geometric object matching, we have cast the problem as a structured graph clustering problem, in which runs of the optimization routine are nodes and the similarity of the resulting components prescribe the edges between them. By leveraging a cycle consistency constraint, a low-rank approximation of this inter-factorization similarity graph can then be computed, which allows an effective clustering of factorization components. We envision that this method may be helpful to data mining practitioners who wish to verify the reproducibility of components without going through the trouble of manual alignment. Two typical (and potentially overlapping) use cases are 1) the estimation and interpretation of individual components from data when there is concern about their dependency on the algorithm's initialization 2) assessing commonality (or stationarity) of components between two distinct, but related datasets, such as in split-half experiments. We tested our method on the tensor decomposition of a real EEG dataset and the coupled matrix-tensor factorization of synthetic data, and showed that it is capable of finding meaningful and accurate grouping patterns over the repetitions of the decomposition. We highlighted its strengths in comparison to an existing algorithm which was developed for ICA.

References

[1] D. Lahat, T. Adalı *et al.*, "Multimodal data fusion: An overview of methods, challenges, and prospects," *Proceedings of the IEEE*, vol. 103, no. 9, pp. 1449–1477, 2015.

[2] N. D. Sidiropoulos, L. De Lathauwer *et al.*, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

[3] R. Bro, "PARAFAC. Tutorial and applications," *Chemometrics and intelligent laboratory systems*, vol. 38, no. 2, pp. 149–171, 1997.

[4] L. Sorber, M. Van Barel *et al.*, "Structured data fusion," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 4, pp. 586–600, 2015.

[5] M. Sørensen, I. Domanov *et al.*, "Coupled canonical polyadic decompositions and (coupled) decompositions in multilinear rank-$(L_{r,n}, L_{r,n}, 1)$ terms—Part II: Algorithms," *SIAM Journal on Matrix Analysis and Applications*, vol. 36, no. 3, pp. 1015–1045, 2015.

[6] J. Himberg, A. Hyvärinen *et al.*, "Validating the independent components of neuroimaging time series via clustering and visualization," *Neuroimage*, vol. 22, no. 3, pp. 1214–1222, 2004.

[7] R. Mareček, M. Lamoš *et al.*, "Multiway array decomposition of EEG spectrum: Implications of its stability for the exploration of large-scale brain networks," *Neural computation*, vol. 29, no. 4, pp. 968–989, 2017.

[8] D. Huber, "Automatic three-dimensional modeling from reality," Ph.D. dissertation, Carnegie Mellon University, Pittsburgh, PA, 2002.

[9] C. Zach, M. Klopschitz *et al.*, "Disambiguating visual relations using loop constraints," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 1426–1433.

[10] Q.-X. Huang and L. Guibas, "Consistent shape maps via semidefinite programming," in *Proceedings of the Eleventh Eurographics/ACMSIGGRAPH Symposium on Geometry Processing*. Eurographics Association, 2013, pp. 177–186.

[11] Y. Chen, L. Guibas *et al.*, "Near-optimal joint object matching via convex relaxation," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 100–108.

[12] C. Bajaj, T. Gao *et al.*, "SMAC: simultaneous mapping and clustering using spectral decompositions," in *International Conference on Machine Learning*, 2018, pp. 334–343.

[13] Y. Shen, Q. Huang *et al.*, "Normalized spectral map synchronization," in *Advances in Neural Information Processing Systems*, 2016, pp. 4925–4933.

[14] E. Acar, E. E. Papalexakis *et al.*, "Structure-revealing data fusion," *BMC bioinformatics*, vol. 15, no. 1, p. 239, 2014.

[15] N. Vervliet, O. Debals *et al.*, "Tensorlab 3.0—Numerical optimization strategies for large-scale constrained and coupled matrix/tensor factorization," in *2016 50th Asilomar Conference on Signals, Systems and Computers*. IEEE, 2016, pp. 1733–1738.