Joint Tensor Compression for Coupled Canonical Polyadic Decompositions

Jérémy E Cohen, Rodrigo Cabral Farias, Pierre Comon

Joint Tensor Compression for Coupled Canonical Polyadic Decompositions

Jeremy Emile Cohen, Rodrigo Cabral Farias and Pierre Comon
GIPSA-lab
11 rue des mathematiques
Grenoble, France
firstname.name@gipsa-lab.fr

***Abstract*—To deal with large multimodal datasets, coupled canonical polyadic decompositions are used as an approximation model. In this paper, a joint compression scheme is introduced to reduce the dimensions of the dataset. Joint compression allows to solve the approximation problem in a compressed domain using standard coupled decomposition algorithms. Computational complexity required to obtain the coupled decomposition is therefore reduced. Also, we propose to approximate the update of the coupled factor by a simple weighted average of the independent updates of the coupled factors. The proposed approach and its simplified version are tested with synthetic data and we show that both do not incur substantial loss in approximation performance.**

I. Introduction

Among a plethora of multivariate models, low rank tensors emerged as natural joint models to approximate latent multilinear quantitative information contained in blocks of data with multiple diversities. They became standard models in chemometrics, antenna array processing, sociology, medical imaging, higher order statistics, hyperspectral imaging, and quantum mechanics [1], [2], [3], [4], [5]. For example, time series of hyperspectral images can be analyzed using low rank tensors [5] with diversities corresponding to spatial, spectral and temporal variations.

Why tensors are useful. It is possible to treat these data with matrix-based methods such as principal components analysis, linear regression, nonnegative matrix factorization. But by doing so, intricate multilinear information contained in the tensor is lost.

Apart from their ability to mine for patterns on data without losing multilinear information, the second reason why tensors are so important in data analysis is the existence of powerful theoretical results linking tensor decomposition and measured physical phenomena. For example, consider a given tensor as an hyper-matrix with three indices or more. It is then tempting to generalize the singular value decomposition (SVD) to tensors. A vastly used generalization is the canonical polyadic decomposition (CPD), invented by Hitchcock in 1927, also called CANDECOMP/PARAFAC decomposition.

Research funded by ERC "DECODA" no. 320594, with european program FP7/2007-2013.

The strength of the CPD is that its parameters may be obtained uniquely from the data, so that constraints such as orthogonality need not be imposed as in the SVD. In other words, the CPD of a tensor hopefully recovers physically meaningful parameters from data, such as spectra in chemometrics or abundances in hyperspectral images.

Why coupling is useful. Among many challenges of tensor data analysis, two different topics became especially important: coupling multiple decompositions and speeding up computations by dimensionality reduction. With ever-increasing data storage and ever-reducing sensor costs, in the last decade, the size of data to be analyzed has sky-rocketed. In particular, for tensors, increasing dimensions of all modes may result in an enormous increase in computation time and data storage requirements. On the other hand, this also means that data are now obtained through multiple sensors recording different modalities of the same phenomenon. From these multiple datasets, it is then natural to try to extract common information. This common information can be exploited by coupling the the datasets models through shared latent variables. For instance, EEG and MEG both capture different spatial fingerprints from the same time-evolving sources in the brain. For this reason, joint decompositions have been used extensively in this context [6]. Instances of coupled tensor decompositions can also be found in chemometrics or array processing, among others [7].

Why compression is useful. To handle dimensionality reduction without destroying the multilinear structure of the data, standard tools of matrix algebra can be used [1], [8] as explained in section II. Works have been proposed to extend this compression method to more difficult models including constraints [9], but little has been done for compressing coupled data sets when the decomposition model is a coupled CPD. The goal in this paper is not to deal with extremely huge tensors, but to show that dimensionality reduction can be applied before applying a coupled CPD to tensor data, and that this compression may not deteriorate the coupling relationship if properly computed. A first idea of joint dimentionality reduction can be found in [8], but not in the context of tensors coupled through their decomposition. For the sake of clarity, only the case of tensors having identical factors in one mode of their decompositions will be studied in this paper, but we still shed light on issues with more flexible couplings, which

recently appeared in the literature [10], [11].

Contents of the paper. After recalling some basics on tensor algebra, a theoretical framework for joint dimensionality reduction when computing a coupled CPD is presented. A consensus algorithm is then described to compute the compressed CPD, and simulation results are shown to illustrate the power of the suggested compression method. Finally, we discuss extensions of this work to more general couplings, which appear to be more difficult problems.

II. Tensor Decompositions

A. Definitions

A tensor $\boldsymbol{\mathcal{T}}$ is a multilinear application from a Cartesian product space of n_1 vector spaces to another Cartesian product of n_2 vector spaces. If bases are fixed in each of these product spaces, $\boldsymbol{\mathcal{T}}$ can be seen as an array with $n = n_1 + n_2$ dimensions. We say that $\boldsymbol{\mathcal{T}}$ is of order n. Thus a matrix is exactly a tensor of order 2 [3].

Similarly to the matrix algebra, one of the main challenges of tensor data analysis is to identify appropriate bases of representation for the data with respect to the information to be extracted. To do so, two generalizations of the SVD have been developed, namely the CPD and the Tucker model [12]. For CPD, the tensor should be diagonal in the new basis whereas in the Tucker model, orthogonality is imposed among the basis vectors, but the resulting tensor is not diagonal. Note that for matrices, both constraints can be achieved at the same time with the SVD, this is not the case for higher order tensors.

To change the basis of representation of a tensor, it is sufficient to change the basis of each vector space in the Cartesian product spaces defining the tensor. Therefore, for a third order tensor, three matrices can be used to define this basis multilinear transformation, and the following notation can be used:

$$\boldsymbol{\mathcal{T}} = (\boldsymbol{U} \otimes \boldsymbol{V} \otimes \boldsymbol{W})\, \boldsymbol{\mathcal{T}}' \tag{1}$$

where $\otimes$ here denotes tensor product of linear operators on each mode. It is not the Kronecker product in the general case [13]. Another usual notation is the n-mode product [14].

In the case of a second order tensor (matrix) $\boldsymbol{M}$, for well chosen orthonormal basis $\boldsymbol{U}$ and $\boldsymbol{V}$ such that $\boldsymbol{M}'$ is a diagonal matrix with the least number of nonzero elements in the diagonal, this yields the SVD: $\boldsymbol{M} = (\boldsymbol{U} \otimes \boldsymbol{V})\, \boldsymbol{M}' = \boldsymbol{U}\boldsymbol{M}'\boldsymbol{V}^T$.

B. CPD

If one wishes to express the tensor as a diagonal tensor in a new basis, the multilinear operator transforming the tensor into this new basis defines the CPD:

$$\boldsymbol{\mathcal{T}} = (\boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{C})\, \boldsymbol{\mathcal{L}} = \sum_{r=1}^{R} \lambda_r\, \boldsymbol{A}_{:,r} \otimes \boldsymbol{B}_{:,r} \otimes \boldsymbol{C}_{:,r} \tag{2}$$

where $\boldsymbol{\mathcal{L}}$ is a diagonal tensor and $\otimes$ in the right-hand side can be understood as an outer product, R is the minimum number of non-zero values in $\boldsymbol{\mathcal{L}}$ and is called tensor rank (it actually coincides with usual rank when $\boldsymbol{\mathcal{T}}$ is a matrix). Matrices $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{C}$ are sometimes called the factors of $\boldsymbol{\mathcal{T}}$ and are usually not orthogonal.

The important feature of the CPD is that it supposedly explains the data stored in $\boldsymbol{\mathcal{T}}$ with a multilinear model where each way, or modality, interacts with each other in a multiplicative fashion. These interactions are decomposed in R terms, each characterizing a simple component of the complex interaction between the modalities. Since laws of physics are often approximately linear, the CPD is commonly used to retrieve parameters of a physical model.

Under mild sufficient conditions, often met in practice, this decomposition is unique [15], [3]. However, in practice, data are corrupted by noise, which typically has a much higher rank than noiseless data. Thus, computing a meaningful tensor decomposition becomes a low rank approximation problem, which turns out to be ill-posed in general, but is still tractable [16], [17].

C. High Order Singular Value Decomposition

If orthogonality is imposed on the transformation matrices of higher order tensors ($n > 2$), picking the minimum number of vectors to describe each new basis corresponds the High Order SVD (HOSVD) [8]. This decomposition can be obtained exactly with the algorithm below, which is also sufficiently robust to noise. Let $\boldsymbol{T}_{(i)}$ be the matrix obtained by unfolding $\boldsymbol{\mathcal{T}}$ along the i-th mode,

$$\begin{cases} \boldsymbol{U}\boldsymbol{N_1} = \text{SVD}\left(\boldsymbol{T}_{(1)}\right) \\ \boldsymbol{V}\boldsymbol{N_2} = \text{SVD}\left(\boldsymbol{T}_{(2)}\right) \\ \boldsymbol{W}\boldsymbol{N_3} = \text{SVD}\left(\boldsymbol{T}_{(3)}\right) \end{cases} \qquad \boldsymbol{\mathcal{T}} = (\boldsymbol{U} \otimes \boldsymbol{V} \otimes \boldsymbol{W})\, \boldsymbol{\mathcal{G}} \tag{3}$$

The obtained tensor $\boldsymbol{\mathcal{G}}$ is called the core, and the dimensions R_1, R_2 and R_3 of the columns of $\boldsymbol{U}$, $\boldsymbol{V}$ and $\boldsymbol{W}$ are the multilinear ranks of $\boldsymbol{\mathcal{T}}$. It is easily checked that multilinear ranks are always smaller than or equal to tensor rank. Thus algorithm (3) can be used for compression, since no information of interest will be lost by truncating the SVDs up to R singular values. Compressed tensor $\boldsymbol{\mathcal{G}}$ is of dimensions $R \times R \times R$ and its CPD is related to the CPD of $\boldsymbol{\mathcal{T}}$ through a simple change of basis,

$$\boldsymbol{\mathcal{T}} = (\boldsymbol{U} \otimes \boldsymbol{V} \otimes \boldsymbol{W})\, \boldsymbol{\mathcal{G}} = (\boldsymbol{U}\boldsymbol{A}_c \otimes \boldsymbol{V}\boldsymbol{B}_c \otimes \boldsymbol{W}\boldsymbol{C}_c)\, \boldsymbol{\mathcal{L}} \tag{4}$$

$$\boldsymbol{A}_c = \boldsymbol{U}^T\boldsymbol{A},\ \boldsymbol{B}_c = \boldsymbol{V}^T\boldsymbol{B},\ \boldsymbol{C}_c = \boldsymbol{W}^T\boldsymbol{C} \tag{5}$$

In the noisy case, compression is lossy but often considered of high precision [8]. The multilinear ranks R_1, R_2 and R_3 should be set a little higher than R to limit compression loss.

As a main disadvantage, HOSVD cannot be used for parameter estimation, due to the unrealistic orthogonality constraints on the factors and on the core. Still, it is a powerful tool for denoising and compression. In this paper, we consider HOSVD only as a compression tool and, in the following section, we show how it can be used to compress jointly coupled tensors.

III. Compression of coupled tensors

A. Coupling two tensors

Suppose that two datasets are acquired in the form of tensors $\mathcal{T}$ and $\mathcal{T}'$, and that these datasets correspond to measurements of the same phenomenon. Then a way to write the similarity between the tensors is to impose a common factor in their CPD. Considering white additive zero-mean Gaussian measurement noise yields the following coupled models:

$$\begin{cases} \mathcal{T} = (\boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{C})\, \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{E}} \\ \mathcal{T}' = \left(\boldsymbol{A}' \otimes \boldsymbol{B}' \otimes \boldsymbol{C}'\right) \boldsymbol{\mathcal{L}}' + \boldsymbol{\mathcal{E}}' \\ \boldsymbol{C} = \boldsymbol{C}' + \boldsymbol{\Gamma},\ \boldsymbol{\Gamma}_{ij} \sim \mathcal{N}(0, \sigma_c^2) \end{cases} \tag{6}$$

where $\boldsymbol{\mathcal{E}}$ and $\boldsymbol{\mathcal{E}}'$ are the independent measurement noises with variances σ_n^2 and $\sigma_n'^2$, and σ_c^2 stands for the variance of a random difference between $\boldsymbol{C}$ and $\boldsymbol{C}'$, which allows coupling to be more flexible.

For reasons of space, in what follows we develop a joint compression scheme for the exact coupling case, that is when $\boldsymbol{C} = \boldsymbol{C}'$. After presenting simulation results, we discuss the difficulties encountered with the more general case of flexible couplings [11].

B. Independent or joint compression ?

A direct approach to compress two coupled tensors is to compress each of them independently, and then use the information of the coupling only in the CPD computation stage. There are two issues with this method. First, two independent compressions make no use of the fact that a constraint links the factors. Thus it is sub-optimal.

Indeed, suppose $\boldsymbol{W}$ and $\boldsymbol{W}'$ are two bases for the third mode of $\mathcal{T}$ and $\mathcal{T}'$. Then exact coupling on the third mode means that $span(\boldsymbol{W}) = span(\boldsymbol{W}')$. Thus only one basis $\boldsymbol{W}_j$ is needed to compress the two tensors.

Second, since data are noisy, compression bases will be estimated with some error. However, the coupling relationship is written for the true factors, so that compressing independently will destroy the coupling relationship. Indeed, defining $\boldsymbol{W}\boldsymbol{C}_c = \boldsymbol{C}$ for noiseless compression and $\hat{\boldsymbol{W}}^T \boldsymbol{C} = \hat{\boldsymbol{C}}_c$ for noisy compression,

$$\boldsymbol{W}\boldsymbol{C}_c = \boldsymbol{W}'\boldsymbol{C}_c' \text{ but } \hat{\boldsymbol{W}}\hat{\boldsymbol{C}}_c \neq \hat{\boldsymbol{W}}'\hat{\boldsymbol{C}}_c'. \tag{7}$$

However, if the same basis is used for compressing the two tensors, then the estimation error on $\boldsymbol{W}_j$ is reduced. Moreover it does not affect the coupling model:

$$\boldsymbol{C} = \boldsymbol{C}' \Rightarrow \hat{\boldsymbol{W}}_j^T \boldsymbol{C} = \hat{\boldsymbol{W}}_j^T \boldsymbol{C}' \equiv \boldsymbol{C}_c = \boldsymbol{C}_c' \tag{8}$$

It is clear that the best basis to jointly compress both tensors $\boldsymbol{W}_j$ is obtained by a truncated SVD of the stacked unfolding matrices. If noise levels are different, then they can be taken into account when stacking the unfolding matrices:

$$\left[\frac{\boldsymbol{T}_{(3)}}{\sigma_n}\ ,\ \frac{\boldsymbol{T}'_{(3)}}{\sigma'_n}\right] = \boldsymbol{W}_j \boldsymbol{\Sigma_j} \boldsymbol{Q}_j \tag{9}$$

By choosing an orthonormal basis, the observation noise is still white in both tensors. Moreover we have shown that the coupling model is direct even in the compressed space. Thus, the compressed optimization problem to solve (6) with no noise in the coupling is the following

$$\begin{array}{ll} \text{minimize} & \Upsilon = (1/\sigma_n^2) \left\| \boldsymbol{\mathcal{G}} - (\boldsymbol{A}_c \otimes \boldsymbol{B}_c \otimes \boldsymbol{C}_c)\, \boldsymbol{\mathcal{L}} \right\|_F^2 + \\ & \quad + (1/{\sigma_n'}^2) \left\| \boldsymbol{\mathcal{G}}' - (\boldsymbol{A}_c' \otimes \boldsymbol{B}_c' \otimes \boldsymbol{C}_c')\, \boldsymbol{\mathcal{L}}' \right\|_F^2 \\ \text{w.r.t.} & \boldsymbol{A}_c, \boldsymbol{B}_c, \boldsymbol{C}_c, \boldsymbol{\mathcal{L}}, \boldsymbol{A}_c', \boldsymbol{B}_c', \boldsymbol{C}_c', \boldsymbol{\mathcal{L}}' \\ \text{subject to} & \boldsymbol{C}_c = \boldsymbol{C}_c' \\ \text{where} & \mathcal{T} = (\boldsymbol{U} \otimes \boldsymbol{V} \otimes \boldsymbol{W}_j)\, \boldsymbol{\mathcal{G}} \\ & \mathcal{T}' = (\boldsymbol{U}' \otimes \boldsymbol{V}' \otimes \boldsymbol{W}_j)\, \boldsymbol{\mathcal{G}}' \end{array} \tag{10}$$

IV. Alternating algorithm with weighted averages

When computing the CPD of a tensor, there are basically two main categories of method. Altenating least squares (ALS) [1], which relies on the fact that Υ is quadratic with respect to each factor in order to sequentially update each factor until convergence. On the other hand, a gradient-based approach can be used to update all the variables at once. Even though a wide literature discusses advantages and disadvantages of the two methods, they basically suffer from the same drawback of finding only local minima at best.

A gradient-based all-at-once optimization method is described in [18], [7] while an alternating method when the coupling is flexible is presented in [11]. For coupling with equality between factors, standard ALS [1] has to be modified to update $\boldsymbol{C}_c$ by solving the linear system

$$\boldsymbol{C}_c \left[\frac{\boldsymbol{A}_c^{\mathsf{T}} \boldsymbol{A}_c * \boldsymbol{B}_c^{\mathsf{T}} \boldsymbol{B}_c}{\sigma_n^2} + \frac{\boldsymbol{A'}_c^{\mathsf{T}} \boldsymbol{A'}_c * \boldsymbol{B'}_c^{\mathsf{T}} \boldsymbol{B'}_c}{{\sigma'}_n^2} \right] = \frac{\boldsymbol{G}^{(3)} (\boldsymbol{A}_c \odot \boldsymbol{B}_c)}{\sigma_n^2} + \frac{\boldsymbol{G'}^{(3)} (\boldsymbol{A}_c' \odot \boldsymbol{B}_c')}{{\sigma'}_n^2} \tag{11}$$

where $*$ denotes entry-wise product and $\odot$ denotes Khatri-Rao product.

Having in mind possible extensions of coupled decomposition to N datasets with large N, we may want to process the datasets in a distributed fashion. In this context, the update of the coupled factor is the only operation which cannot be parallelized through the different datasets. A straightforward option to solve (11) in a distributed way is to evaluate the matrices $\boldsymbol{A}_c^{\mathsf{T}} \boldsymbol{A}_c * \boldsymbol{B}_c^{\mathsf{T}} \boldsymbol{B}_c$ and $\boldsymbol{G}^{(3)} (\boldsymbol{A}_c \odot \boldsymbol{B}_c)$ in parallel, then apply a weighted consensus algorithm [19] with weights given by $1/\sigma_n^2$ to retrieve the sums and, finally, solve the system at each processing node. Another approximate solution is to assume that each step of uncoupled ALS generates an unbiased estimate of $\mathrm{vec}\,(\boldsymbol{C}_c)$ with known covariances. We can then obtain a better estimate by merging the independent estimators with a scalar best linear unbiased estimator (BLUE). For two datasets, assuming estimation covariances $\boldsymbol{D}$ and $\boldsymbol{D}'$, the scalar BLUE $\boldsymbol{C}_c^\star$ of $\boldsymbol{C}_c$ is given by

$$\boldsymbol{C}_c^\star = \left(\mathrm{Tr}\left(\boldsymbol{D}\right)^{-1}\boldsymbol{C}_c + \mathrm{Tr}\left(\boldsymbol{D}'\right)^{-1}\boldsymbol{C}_c'\right)/(\mathrm{Tr}\left(\boldsymbol{D}\right)^{-1}+\mathrm{Tr}\left(\boldsymbol{D}'\right)^{-1}) \tag{12}$$

where $\mathrm{Tr}(\cdot)$ denotes the trace. In practice, estimation covariances are not available, and we can assume that $(\mathrm{Tr}\left(\boldsymbol{D}\right)/\sigma_n^2) \approx (\mathrm{Tr}\left(\boldsymbol{D}'\right)/\sigma'^2_n)$, thus leading to a simple weighted average:

$$\boldsymbol{C}_c^\star = \left((1/\sigma_n^2)\boldsymbol{C}_c + (1/\sigma'^2_n)\boldsymbol{C}_c'\right)((1/\sigma_n^2) + (1/\sigma'^2_n)) \tag{13}$$

Note that for $N \gg 2$ datasets, this weighted average can also be implemented in a distributed way using weighted consensus. Observe from the assumption on the estimation variances that for similar noise levels on the datasets, estimation performance on $\boldsymbol{C}_c$ and $\boldsymbol{C}_c'$ must be similar. From the multilinear structure, this mainly depends on the correlation structure of the columns of the other factors. As we will see in simulations for tensors with random factors, this approximation gives an estimation performance very close to (11).

V. Simulation on synthetic data

For each tensor, three factor matrices of size 50×3 are drawn independently according to a standard Gaussian distribution, but the third factor is exactly the same for both tensors. Then the tensors are normalized to have a unit Frobenius norm. Five algorithms to compute the CPD are compared. Three work in a compressed domain of size $3 \times 3 \times 3$. Compression is computed jointly, but then only two algorithms consider coupling in the compressed domain, one being the weighted average ALS described in section IV while the second one is an exact coupled ALS with update (11). The third one is an uncoupled plain ALS for each small tensor. Two other algorithms do not compress the initial tensors. One just runs two independent ALS, the other is the exact coupled ALS with update (11) but in the uncompressed domain. The total mean squared error (MSE) on the coupled factors is plotted in Figure 1, signal to noise ratio (SNR) of the first tensor is fixed to approximately $33\mathrm{dB}$ ($\sigma_n = 10^{-4}$) while the other varies from 0 to $40\mathrm{dB}$.

It appears from the simulation results that the coupling relationship is crucial in the compression. Indeed, the jointly compressed independent ALS algorithm has smaller total MSE than the uncompressed independent ALS. Since compression can also be seen as denoising, it can be concluded that joint data compression helps denoising the noisy dataset using information about the span of coupled factors contained in less noisy data. This is not intuitive, since in theory compressing should increase estimation error by reducing the amount of information contained in a data set. Moreover, even without including the coupling knowledge in the ALS algorithm, it is already efficient to simply compress jointly in order to compute small independent ALS. That is why an extension of joint compression to more complex models would be of crucial importance.

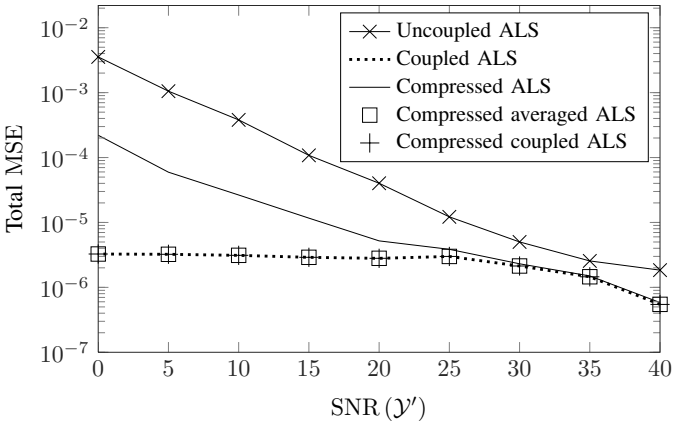

Fig. 1. Total MSE for estimation of factors $\boldsymbol{C}$ and $\boldsymbol{C}'$ for the five described algorithms. MSE is averaged on 100 realizations of the CP model and noise, each estimate is the best among six different initialization points.

VI. Extensions and Challenges

Recently, direct coupling has been extended to cover more complex interaction between datasets. In [10], only a few number of components are considered shared, while others are not a priori related. In a previous work, we proposed a Bayesian framework to cover noisy and non-linear relations between the factors, although only the noisy linear coupling model case has been well-studied as of yet [11]. Overall, extensions concern error modeling or transformation modeling in the coupling model. Either way, obtaining a joint compression based on linear tools is difficult. Let us see why with a few simple examples.

A. Shared/Unshared components

Assume the following coupling model.

$$\boldsymbol{C} = \left[\boldsymbol{C}^\star, \tilde{\boldsymbol{C}}\right] \quad \boldsymbol{C}' = \left[\boldsymbol{C}^\star, \tilde{\boldsymbol{C}}'\right] \tag{14}$$

where $\boldsymbol{C}^\star$ has $r \leq R$ columns and stands for shared components. In the easiest scenario, r is known. Then compressing the two data sets jointly would mean that only a part of the joint basis $\boldsymbol{W}_j$ spans the shared subspace. However, since a priori $\boldsymbol{C}^\star$ is not orthogonal to $\tilde{\boldsymbol{C}}$ and $\tilde{\boldsymbol{C}}'$, what is obtained when computing the SVD as in (9) cannot be identified as shared and unshared singular vectors. Especially when the data are noisy, it is not clear whether this method is still optimal.

On the other hand, it may provide us with a tool for identifying the number of shared components, since the dimension of the span of the stacked tensors should not be greater than $2R - r$.

B. Noisy coupling

If two datasets are obtained from the same samples but with a slight change in the experimental setting, then it is possible that the shared factors $\boldsymbol{C}$ and $\boldsymbol{C}'$ have some discrepancies

w.r.t. the true underlying factor $\boldsymbol{C}^\star$ measured by a probability density. In other words,

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{C}^\star + \boldsymbol{\Gamma} \\ \boldsymbol{C}' &= \boldsymbol{C}^\star + \boldsymbol{\Gamma}' \end{aligned} \tag{15}$$

where $\boldsymbol{\Gamma}$ and $\boldsymbol{\Gamma}'$ follow two independent matrix normal distributions with diagonal covariances. Then trying to express the coupling in a direct fashion leads to additional *correlated* observation noise.

$$\begin{aligned} \boldsymbol{\mathcal{T}} &= (\boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{C}^\star)\, \boldsymbol{\mathcal{L}} + (\boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{\Gamma})\, \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{E}} \\ \boldsymbol{\mathcal{T}}' &= \left(\boldsymbol{A}' \otimes \boldsymbol{B}' \otimes \boldsymbol{C}^\star\right) \boldsymbol{\mathcal{L}}' + \left(\boldsymbol{A}' \otimes \boldsymbol{B}' \otimes \boldsymbol{\Gamma}'\right) \boldsymbol{\mathcal{L}}' + \boldsymbol{\mathcal{E}}' \end{aligned} \tag{16}$$

Now we have an *exact* coupled decomposition model where the noise is correlated. Moreover, this correlation depends on the factors to be estimated $\boldsymbol{A}$ and $\boldsymbol{B}$. This can be tackled by first computing two independent decompositions, then using the estimates of uncoupled factors to write the correlation model in (16) to find a common representation basis under this correlated noise. This is however much trickier than the joint compression presented earlier, and an optimal solution is still to be found.

C. Linear Coupling

There is however one case which directly relates to the joint compression as presented earlier. If the coupling is noiseless but expressed through a linear transformation $\boldsymbol{H}$, *i.e.*

$$\boldsymbol{C}' = \boldsymbol{H}\boldsymbol{C}, \tag{17}$$

then a simple multiplication by $\boldsymbol{H}$ of the third mode of the first tensor yields exact coupling:

$$\boldsymbol{\mathcal{Y}} = (\boldsymbol{I} \otimes \boldsymbol{I} \otimes \boldsymbol{H})\, \boldsymbol{\mathcal{T}} = \left(\boldsymbol{A} \otimes \boldsymbol{B} \otimes \boldsymbol{C}'\right) \boldsymbol{\mathcal{L}} + \boldsymbol{\mathcal{E}}'' \tag{18}$$

where $\boldsymbol{\mathcal{E}}''$ is correlated on the third mode. That is, $\mathrm{vec}(\boldsymbol{\mathcal{E}}'')$ follows a normal distribution with Kronecker product covariance $\boldsymbol{I} \boxtimes \boldsymbol{I} \boxtimes \boldsymbol{H}\boldsymbol{H}^T$. However, the left inverse of $\boldsymbol{H}$ is required to recover factor $\boldsymbol{C}$. In this scenario, the joint compression scheme presented earlier can then be applied on the transformed data.

VII. Conclusion

In this paper we have dealt with joint dimensionality reduction for coupled tensors when joint CPD is to be applied to several datasets. A simple algorithm is presented for joint compression in the exact setting, and then demonstrated on simulated data along with a simple coupling algorithm. Finally, we show how the joint compression problem raises many issues and could become an important topic among data scientists interested in multimodality.

References

[1] R. Bro, "PARAFAC, tutorial and applications," *Chemom. Intel. Lab. Syst.*, vol. 38, pp. 149–171, 1997.

[2] N. D. Sidiropoulos, R. Bro, and G. B. Giannakis, "Parallel factor analysis in sensor array processing," *IEEE Trans. Signal Process.*, vol. 48, no. 8, pp. 2377–2388, Aug. 2000.

[3] P. Comon, "Tensors: a brief introduction," *IEEE Signal Process. Magazine*, vol. 31, no. 3, pp. 44–53, May 2014, special issue on BSS. hal-00923279.

[4] L. Gauvin, A. Panisson, and C. Cattuto, "Detecting the community structure and activity patterns of temporal networks: a non-negative tensor factorization approach," *PloS One*, vol. 9, no. 1, pp. 1–13, Jan. 2014, open access, www.plosone.org.

[5] M. A. Veganzones, J. E. Cohen, R. C. Farias, J. Chanussot, and P. Comon, "Nonnegative tensor CP decomposition of hyperspectral data," *IEEE Trans. Geoscience and Remote Sensing*, 2016, accepted.

[6] H. Becker, L. Albera, P. Comon, M. Haardt, G. Birot, F. Wendling, M. Gavaret, C. G. Bénar, and I. Merlet, "EEG extended source localization: Tensor-based vs conventional methods," *NeuroImage*, vol. 96, pp. 143–157, Aug. 2014.

[7] E. Acar, T. G. Kolda, and D. M. Dunlavy, "All-at-once optimization for coupled matrix and tensor factorizations," in *MLG'11: Proceedings of Mining and Learning with Graphs*, Aug. 2011.

[8] L. D. Lathauwer and J. Vandewalle, "Dimensionality reduction in higher-order signal processing and rank-$(R_1, R_2, ..., R_N)$ reduction in multilinear algebra," *Linear Algebra Appl.*, vol. 391, pp. 31–55, 2004.

[9] J. E. Cohen, R. C. Farias, and P. Comon, "Fast decomposition of large nonnegative tensors," *IEEE Signal Process. Letters*, vol. 22, no. 7, pp. 862–866, Jul. 2015, publ. online in Nov. 2014.

[10] E. Acar, R. Bro, and A. K. Smilde, "Data fusion in metabolomics using coupled matrix and tensor factorizations," *Proc. IEEE*, vol. 103, no. 9, pp. 1602–1620, Aug. 2015.

[11] R. Cabral Farias, J. E. Cohen, and P. Comon, "Exploring multimodal data fusion through joint decompositions with flexible couplings," *IEEE Trans. Signal Process.*, May 2015, submitted, arxiv:1505.07717.

[12] L. R. Tucker, "Some mathematical notes for three-mode factor analysis," *Psychometrika*, vol. 31, pp. 279–311, 1966.

[13] J. E. Cohen, *About Notations in Multiway Array Processing*, Nov. 2015, arXiv:1511.01306.

[14] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM review*, vol. 51, no. 3, pp. 455–500, 2009.

[15] J. B. Kruskal, "Three-way arrays: Rank and uniqueness of trilinear decompositions," *Linear Algebra Appl.*, vol. 18, pp. 95–138, 1977.

[16] V. D. Silva and L.-H. Lim, "Tensor rank and the ill-posedness of the best low-rank approximation problem," *SIAM J. Matrix Anal. Appl.*, vol. 30, no. 3, pp. 1084–1127, 2008.

[17] L.-H. Lim and P. Comon, "Blind multilinear identification," *IEEE Trans. Inf. Theory*, vol. 60, no. 2, pp. 1260–1280, Feb. 2014, open access.

[18] P. Comon, X. Luciani, and A. L. F. De Almeida, "Tensor decompositions, alternating least squares and other tales," *J. Chemometrics*, vol. 23, no. 7-8, pp. 393–405, Aug. 2009.

[19] R. Olfati-Saber, A. Fax, and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proc. IEEE*, vol. 95, no. 1, pp. 215–233, Jan. 2007.