Large-scale Canonical Polyadic Decomposition via Regular Tensor Sampling

Charilaos I. Kanatsoulis
Dept. ECE, Univ. of Minnesota
Minneapolis, MN USA
kanat003@umn.edu

Nicholas D. Sidiropoulos
Dept. ECE, Univ. of Virginia
Charlottesville, VA, USA
nikos@virginia.edu

***Abstract*—Tensor decomposition models have proven to be effective analysis tools in various applications, including signal processing, machine learning, and communications, to name a few. Canonical polyadic decomposition (CPD) is a very popular model, which decomposes a higher order tensor signal into a sum of rank 1 terms. However, when the tensor size gets big, computing the CPD becomes a lot more challenging. Previous works proposed using random (generalized) tensor sampling or compression to alleviate this challenge. In this work, we propose using a regular tensor sampling framework instead. We show that by appropriately selecting the sampling mechanism, we can simultaneously control memory and computational complexity, while guaranteeing identifiability at the same time. Numerical experiments with synthetic and real data showcase the effectiveness of our approach.**

***Index Terms*—tensor, big data, large-scale, canonical polyadic decomposition, PARAFAC, identifiability**

I. Introduction

In the era of data deluge, multi-dimensional data, also known as tensors, are ubiquitous in a plethora of engineering tasks and data analytics. Tensor decompositions are essential tools in understanding, analyzing and processing multi-dimensional data. Tensors and tensor decompositions find applications in various fields including signal processing [1], machine learning [2]–[5], data mining [6], [7], remote sensing [8], [9], medical imaging [10] and communications [11], to name a few. For example the *canonical polyadic decomposition* (CPD) is used to co-cluster high dimensional data [5], for mining purposes, or to fuse images with different resolutions in order to produce a super-resolution image [8].

Large-scale, multidimensional tensors have emerged in various engineering domains, facilitated by the rapid development in data acquisition and integration. Tensors with millions or billions of entries are common in numerous fields. A raw fMRI scan, for instance, can be represented as a dense complex tensor with dimensions $10,000 \times 500 \times 2,000$ which corresponds to 10 billion non-zero complex entries. The NELL dataset [12], which represents real world knowledge base data, is a $26 \times 26 \times 48$ million tensor with 144 million non-zero entries. Standard CPD methods, which are computationally intensive and memory demanding, have difficulty in operating with big data tensors. For an $I \times J \times K$ tensor of rank F each iteration of the popular alternating least squares (ALS) method requires IJF additional memory and $IJF + IJKF$ flops for dense or $IJF + 2Fm$ for sparse tensors, where m is the number of non-zero entries. It is therefore clear that computing the CPD of large scale tensors is challenging.

This challenge has motivated numerous works in developing efficient algorithms for computing the CPD of big data tensors. A first class of algorithms focused in efficiently computing the CPD of big sparse tensors [13]–[16]. The work in [13], for example, uses a random sampling mechanism to compute the non-negative CPD of sparse tensors, whereas [15] uses a novel, memory friendly sparse tensor data structure in conjunction with parallel implementation. The idea of random sampling has also been used in computing more general tensor structures. The work in [17] uses randomized block sampling to update only a subset of affected variables at each iteration, thus mitigating the computational burden. Tensor compression is another idea which has been used instead of sampling. First compression was applied via the higher-order singular value decomposition (HOSVD) [18], followed by [19], [20], which create compressed versions of the big tensor by multiplying it with random and pseudo-random matrices respectively. Finally the idea of computing the CPD of an incomplete version of the big tensor has been considered in [21].

Albeit the number and variety of works in computing the decomposition of large-scale tensors, there are still remaining challenges that need to be addressed. First, it is often the case that model identifiability is not discussed, especially in works that use sampling to facilitate the computation. Note that model identifiability is important, since it guarantees that the solution of the computationally lighter problem is the same as the solution of the original one in the noiseless case (or, fixing residuals). Furthermore, although there exist various effective algorithms for big sparse tensors, this is not the case for big and dense ones, which leaves room for additional improvement. Finally, a number of existing works exhibit significant performance drop, when real, noisy data are involved and thus there is need for alternative approaches.

In the current work we address the aforementioned challenges and propose a regular tensor sampling framework to compute the CPD of large-scale tensors. This paper follows our recent work in [10], which studies the problem of regular sampling and reconstruction of tensor signals and proves that under certain conditions, tensor completion from regular

Supported in part by U.S. NSF grants IIS-1447788 and IIS-1704074.

samples is doable. This paper proposes two new multi-modal regular sampling mechanisms, which are identifiable, i.e., an optimal solution is guaranteed to provide the true factors, and accomplish significant speed-up. Furthermore, we develop a lightweight algorithm to perform the CPD computation and verify its effectiveness via synthetic and real data simulations.

II. Tensor Algebra Preliminaries

In this section we briefly present some tensor algebra preliminaries to facilitate the upcoming discussion. We focus on third-order tensors, although the analysis can be easily extended to higher order tensors. The readers are referred to [22], [23] for detailed discussion.

A third order tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{I\times J\times K}$ is a three-way array indexed by (i,j,k). It has three modes: rows $\underline{\boldsymbol{X}}(:,j,k)$, columns $\underline{\boldsymbol{X}}(i,:,k)$ and fibers $\underline{\boldsymbol{X}}(i,j,:)$ and three types of slabs: horizontal $\underline{\boldsymbol{X}}(i,:,:)$, vertical $\underline{\boldsymbol{X}}(:,j,:)$ and frontal $\underline{\boldsymbol{X}}(:,:,k)$. Any tensor can be realized as a sum of three-way outer products, i.e.,

$$\underline{\boldsymbol{X}}(i,j,k) = \sum_{f=1}^{F} \boldsymbol{A}(i,f)\boldsymbol{B}(j,f)\boldsymbol{C}(k,f) \tag{1}$$

The expression in (1) is called the polyadic decomposition (PD). If F is minimal, the decomposition is known as *canonical polyadic decomposition* (CPD) or parallel factor analysis (PARAFAC) and F is the *rank* or *CP-rank* of the tensor. For brevity we use the following notation to denote the CPD of a third-order tensor, $\underline{\boldsymbol{X}} = [\![\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}]\!]$. A pivotal difference between tensors and matrices is that the CPD is unique under mild conditions. The following theorem establishes CPD identifiability of a third-order tensor:

Theorem 1: [24] Let $\underline{\boldsymbol{X}} = [\![\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}]\!]$ with $\boldsymbol{A}: I\times F$, $\boldsymbol{B}: J\times F$, and $\boldsymbol{C}: K\times F$. The decomposition $\underline{\boldsymbol{X}} = [\![\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}]\!]$ is essentially unique with CP rank F if $k_{\boldsymbol{A}}+k_{\boldsymbol{B}}+k_{\boldsymbol{C}} \geq 2F+2$. Essential uniqueness means that $\boldsymbol{A}$, $\boldsymbol{B}$, $\boldsymbol{C}$ are identifiable up to column permutation and scaling/counter-scaling. Here $k_{\boldsymbol{A}}$ denotes the Kruskal rank of a matrix, i.e., the largest integer $k_{\boldsymbol{A}}$ such that *any* $k_{\boldsymbol{A}}$ columns of $\boldsymbol{A}$ are linearly independent.

A tensor can be represented as a matrix form using tensor *matricization*. There are three ways to matricize/unfold a third-order tensor. The following operation, for instance, stacks the fibers of tensor $\underline{\boldsymbol{X}}$ as rows of matrix $\boldsymbol{X}^{(3)}$:

$$\boldsymbol{X}^{(3)} := [\mathrm{vec}(\underline{\boldsymbol{X}}(:,:,1)), \mathrm{vec}(\underline{\boldsymbol{X}}(:,:,2)), \ldots, \mathrm{vec}(\underline{\boldsymbol{X}}(:,:,K))],$$

where '$\mathrm{vec}(\cdot)$' is the vectorization operator. One can see that:

$$\boldsymbol{X}^{(3)} = (\boldsymbol{B}\odot\boldsymbol{A})\boldsymbol{C}^T \in \mathbb{C}^{IJ\times K}, \tag{2}$$

where $\odot$ denotes the Khatri-Rao product. The superscript (3) denotes that the unfolding is performed on the third mode of the tensor.

An important operation in tensor algebra is the mode product, which multiplies a matrix to a tensor in one mode. A joint mode-1, mode-2, and mode-3 product of a third-order tensor is written as:

$$\tilde{\underline{\boldsymbol{X}}} = \underline{\boldsymbol{X}} \times_1 \boldsymbol{P}_1 \times_2 \boldsymbol{P}_2 \times_3 \boldsymbol{P}_3 \tag{3}$$

where "$\times_1$", "$\times_2$", "$\times_3$" denote the operations that multiply each column, row and fiber of $\underline{\boldsymbol{X}}$ with $\boldsymbol{P}_1$, $\boldsymbol{P}_2$, and $\boldsymbol{P}_3$ respectively. The mode product is reflected in the PD of the tensor, i.e., tensor $\tilde{\underline{\boldsymbol{X}}}$ in (3) admits the following PD:

$$\tilde{\underline{\boldsymbol{X}}} = [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{P}_3\boldsymbol{C}]\!].$$

III. Sampling in multiple modes

We begin our discussion by presenting the sampling schemes, used to compute the CPD of large-scale tensors. We propose two sampling mechanisms, which operate on multiple modes of the tensor, and provide identifiability analysis of the sampling model.

A. Combining slab and fiber sampling

The first sampling mechanism, for CPD computation purposes, combines slab and fiber sampling. In particular, we propose to subsample a subset of frontal slabs $\mathcal{S}_f \subseteq \{1,\ldots,K\}$, along with a subset of fibers, defined by rows $\mathcal{S}_r \subseteq \{1,\ldots,I\}$ and columns $\mathcal{S}_c \subseteq \{1,\ldots,J\}$. Then two sub-sampled tensors are formed:

$$\underline{\boldsymbol{Y}}_1 = \underline{\boldsymbol{X}}(\mathcal{S}_r,\mathcal{S}_c,:) = \underline{\boldsymbol{X}} \times_1 \boldsymbol{P}_1 \times_2 \boldsymbol{P}_2 \tag{4}$$

$$\underline{\boldsymbol{Y}}_2 = \underline{\boldsymbol{X}}(:,:,\mathcal{S}_f) = \underline{\boldsymbol{X}} \times_3 \boldsymbol{P}_3, \tag{5}$$

where $\boldsymbol{P}_1 \in \mathbb{R}^{I_1\times I}$, $\boldsymbol{P}_2 \in \mathbb{R}^{J_1\times J}$, $\boldsymbol{P}_3 \in \mathbb{R}^{K_2\times K}$ are row, column and fiber selection matrices corresponding to $\mathcal{S}_r$, $\mathcal{S}_c$, $\mathcal{S}_f$ respectively. An illustration of this sampling technique is depicted in Fig. 1. Note that the samples are

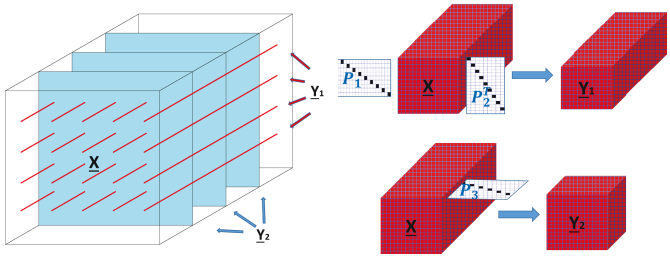


Fig. 1: Combination of fiber and frontal slab sampling.

not drawn arbitrarily. In contrast with [13], [17] the sampling is not random. On the contrary regular and highly structured schemes are preferred since they are simpler to implement.

Now, let $\underline{\boldsymbol{X}} = [\![\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}]\!]$ and $\underline{\boldsymbol{Y}}_1 \in \mathbb{R}^{I_1\times J_1\times K}$, $\underline{\boldsymbol{Y}}_2 \in \mathbb{R}^{I\times J\times K_2}$ as defined in (4). Then it holds:

$$\underline{\boldsymbol{Y}}_1 = [\![\boldsymbol{A}(\mathcal{S}_r,:), \boldsymbol{B}(\mathcal{S}_c,:), \boldsymbol{C}]\!] = [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{C}]\!] \tag{6a}$$

$$\underline{\boldsymbol{Y}}_2 = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}(\mathcal{S}_f,:)]\!] = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_3\boldsymbol{C}]\!] \tag{6b}$$

Identifiability of the model is established, under the conditions of the following theorem:

Theorem 2: Let $\underline{\boldsymbol{X}} \in \mathbb{R}^{I\times J\times K}$, with CPD $\underline{\boldsymbol{X}} = [\![\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}]\!]$. Assume that $\boldsymbol{A}^\star,\boldsymbol{B}^\star,\boldsymbol{C}^\star$ satisfy the equations in (6). Then, $\boldsymbol{A}^\star = \boldsymbol{A}\boldsymbol{\Pi}\boldsymbol{\Lambda}_1$, $\boldsymbol{B}^\star = \boldsymbol{B}\boldsymbol{\Pi}\boldsymbol{\Lambda}_2$, and $\boldsymbol{C}^\star = \boldsymbol{C}\boldsymbol{\Pi}\boldsymbol{\Lambda}_3$, where $\boldsymbol{\Pi}$ is a permutation matrix and $\boldsymbol{\Lambda}_i$ is a full rank diagonal matrix such that $\boldsymbol{\Lambda}_1\boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_3 = \boldsymbol{I}$, provided that $2F+2 \leq k_{\boldsymbol{A}^\star}+k_{\boldsymbol{B}^\star}+k_{\boldsymbol{P}_3\boldsymbol{C}^\star}$ and $\boldsymbol{P}_2\boldsymbol{B}^\star \odot \boldsymbol{P}_1\boldsymbol{A}^\star$ has full column rank.

The proof of Theorem 2 is relegated to the journal version, due to space limitation. The main insight is that if $\underline{\boldsymbol{Y}}_2$ admits a unique CPD, under Theorem 1, one can identify $\boldsymbol{A}$, $\boldsymbol{B}$ up to common permutation and scaling. Then $\boldsymbol{C}$ can be obtained from $\underline{\boldsymbol{Y}}_1$, via a linear system of equations, if $\boldsymbol{P}_2\boldsymbol{B}^\star \odot \boldsymbol{P}_1\boldsymbol{A}^\star$ has full column rank.

B. Fiber sampling in multiple modes

We also propose a fiber sampling mechanism in multiple modes which can further reduce the complexity of computing the CPD. In particular, fiber samples are taken along different modes of the tensor, i.e. rows, columns and fibers are jointly sampled from $\underline{\boldsymbol{X}}$, as illustrated in Fig. 2. Following similar

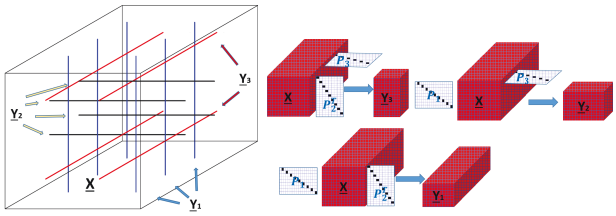


Fig. 2: Multi-mode fiber sampling.

analysis as before we deduce:

$$\underline{\boldsymbol{Y}}_1 \in \mathbb{R}^{I_1 \times J_1 \times K} = \underline{\boldsymbol{X}}\left(\mathcal{S}_r, \mathcal{S}_c, :\right) = \underline{\boldsymbol{X}} \times_1 \boldsymbol{P}_1 \times_2 \boldsymbol{P}_2$$
$$= [\![\boldsymbol{A}\left(\mathcal{S}_r, :\right), \boldsymbol{B}\left(\mathcal{S}_c, :\right), \boldsymbol{C}]\!] = [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{C}]\!] \quad (7a)$$
$$\underline{\boldsymbol{Y}}_2 \in \mathbb{R}^{I_2 \times J \times K_2} = \underline{\boldsymbol{X}}\left(\mathcal{S}_r, :, \mathcal{S}_f\right) = \underline{\boldsymbol{X}} \times_1 \boldsymbol{P}_1 \times_3 \boldsymbol{P}_3$$
$$= [\![\boldsymbol{A}\left(\mathcal{S}_r, :\right), \boldsymbol{B}, \boldsymbol{C}\left(\mathcal{S}_f, :\right)]\!] = [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_3\boldsymbol{C}]\!] \quad (7b)$$
$$\underline{\boldsymbol{Y}}_3 \in \mathbb{R}^{I \times J_3 \times K_3} = \underline{\boldsymbol{X}}\left(:, \mathcal{S}_c, \mathcal{S}_f\right) = \underline{\boldsymbol{X}} \times_2 \boldsymbol{P}_2 \times_3 \boldsymbol{P}_3$$
$$= [\![\boldsymbol{A}, \boldsymbol{B}\left(\mathcal{S}_c, :\right), \boldsymbol{C}\left(\mathcal{S}_f, :\right)]\!] = [\![\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{P}_3\boldsymbol{C}]\!] \quad (7c)$$

As far as identifiability is concerned, we have the following theorem:

Theorem 3: Let $\underline{\boldsymbol{X}} \in \mathbb{R}^{I \times J \times K}$, with CPD $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$. Assume that $\boldsymbol{A}^\star, \boldsymbol{B}^\star, \boldsymbol{C}^\star$ satisfy the equations in (7). Then, $\boldsymbol{A}^\star = \boldsymbol{A}\boldsymbol{\Pi}\boldsymbol{\Lambda}_1$, $\boldsymbol{B}^\star = \boldsymbol{B}\boldsymbol{\Pi}\boldsymbol{\Lambda}_2$, and $\boldsymbol{C}^\star = \boldsymbol{C}\boldsymbol{\Pi}\boldsymbol{\Lambda}_3$, where $\boldsymbol{\Pi}$ is a permutation matrix and $\boldsymbol{\Lambda}_i$ is a full rank diagonal matrix such that $\boldsymbol{\Lambda}_1\boldsymbol{\Lambda}_2\boldsymbol{\Lambda}_3 = \boldsymbol{I}$, provided that $2F + 2 \leq k_{\boldsymbol{P}_1\boldsymbol{A}^\star} + k_{\boldsymbol{P}_2\boldsymbol{B}^\star} + k_{\boldsymbol{C}^\star}$ and $\boldsymbol{P}_3\boldsymbol{C}^\star \odot \boldsymbol{P}_2\boldsymbol{B}^\star$, $\boldsymbol{P}_2\boldsymbol{B}^\star \odot \boldsymbol{P}_1\boldsymbol{A}^\star$ have full column rank, or
$2F+2 \leq k_{\boldsymbol{P}_1\boldsymbol{A}^\star} + k_{\boldsymbol{B}^\star} + k_{\boldsymbol{P}_3\boldsymbol{C}^\star}$ and $\boldsymbol{P}_2\boldsymbol{B}^\star \odot \boldsymbol{P}_1\boldsymbol{A}^\star$, $\boldsymbol{P}_2\boldsymbol{B}^\star \odot \boldsymbol{P}_3\boldsymbol{C}^\star$ have full column rank, or
$2F+2 \leq k_{\boldsymbol{A}^\star} + k_{\boldsymbol{P}_2\boldsymbol{B}^\star} + k_{\boldsymbol{P}_3\boldsymbol{C}^\star}$ and $\boldsymbol{P}_3\boldsymbol{C}^\star \odot \boldsymbol{P}_1\boldsymbol{A}^\star$, $\boldsymbol{P}_2\boldsymbol{B}^\star \odot \boldsymbol{P}_1\boldsymbol{A}^\star$ have full column rank.
In a nutshell, the above theorem states that the multi-mode fiber sampling model is identifiable if one of the subsampled tensors admits a unique CPD, under Theorem 1. The other two subtensors do not need to admit unique CPD's as long as they satisfy certain full column rank conditions. For example, factor $\boldsymbol{C}$ can be identified from the CPD of $\underline{\boldsymbol{Y}}_1$. Then $\boldsymbol{B}$, $\boldsymbol{A}$ are computed from $\underline{\boldsymbol{Y}}_2$, $\underline{\boldsymbol{Y}}_3$ respectively, as solutions to linear system of equations.

IV. Algorithmic framework

The first part of our approach selects an appropriate mechanism, which samples the given tensor, such that the CPD identifiability is maintained. In this section we develop an algorithmic framework which exploits the sampling pattern and reduces the computational and memory complexity of the CPD problem.

A three step approach is being followed for both sampling mechanisms.

Case of slab-fiber sampling: In the first step the CPD of $\underline{\boldsymbol{Y}}_2$ is computed and factors $\boldsymbol{A}$, $\boldsymbol{B}$ are obtained. Then factor $\boldsymbol{C}$ is computed as the solution of the following linear system:

$$\boldsymbol{Y}_1^{(3)} = (\boldsymbol{P}_2\boldsymbol{B} \odot \boldsymbol{P}_1\boldsymbol{A})\boldsymbol{C}^T \quad (8)$$

Finally the following estimator is employed:

$$\underset{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}}{\text{minimize}} \;\; \|\underline{\boldsymbol{Y}}_1 - [\![\boldsymbol{P}_1\boldsymbol{A}, \boldsymbol{P}_2\boldsymbol{B}, \boldsymbol{C}]\!]\|_F^2 + \|\underline{\boldsymbol{Y}}_2 - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{P}_3\boldsymbol{C}]\!]\|_F^2, \quad (9)$$

Problem 9 is non-convex and NP-hard in general. To handle it we employ a *block coordinate descent (BCD)* approach with exact line search, which admits lightweight updates.

Case of multi-mode fiber sampling: The first step computes the CPD of $\underline{\boldsymbol{Y}}_1$ which obtains factor $\boldsymbol{C}$. Note that for multi-mode fiber sampling the CPD of $\underline{\boldsymbol{Y}}_2$ or $\underline{\boldsymbol{Y}}_3$, can be computed instead, with similar analysis. Step 2 computes the remaining factors, e.g. $\boldsymbol{A}$, $\boldsymbol{B}$, as solutions to the following system of linear equations:

$$\boldsymbol{Y}_2^{(2)} = (\boldsymbol{P}_3\boldsymbol{C} \odot \boldsymbol{P}_1\boldsymbol{A})\boldsymbol{B}^T \quad (10a)$$
$$\boldsymbol{Y}_3^{(1)} = (\boldsymbol{P}_3\boldsymbol{C} \odot \boldsymbol{P}_2\boldsymbol{B})\boldsymbol{A}^T \quad (10b)$$

Finally, step 3 solves the following problem as before:

$$\underset{\boldsymbol{A},\boldsymbol{B},\boldsymbol{C}}{\text{minimize}} \left\|\underline{\boldsymbol{Y}}_1 - \left[\!\!\left[\boldsymbol{P}_1^{(1)}\boldsymbol{A}, \boldsymbol{P}_2^{(1)}\boldsymbol{B}, \boldsymbol{C}\right]\!\!\right]\right\|_F^2 +$$
$$\left\|\underline{\boldsymbol{Y}}_2 - \left[\!\!\left[\boldsymbol{P}_1^{(2)}\boldsymbol{A}, \boldsymbol{P}_2^{(2)}\boldsymbol{B}, \boldsymbol{C}\right]\!\!\right]\right\|_F^2 + \left\|\underline{\boldsymbol{Y}}_3 - \left[\!\!\left[\boldsymbol{P}_1^{(3)}\boldsymbol{A}, \boldsymbol{P}_2^{(3)}\boldsymbol{B}, \boldsymbol{C}\right]\!\!\right]\right\|_F^2 \quad (11)$$

The FIber-Slab Tensor sampling algorithm (FIST) is presented in Algorithm 1.

Algorithm 1: FIST

Input: $\underline{\boldsymbol{X}}$, F.
Select sampling mechanism
Sample $\underline{\boldsymbol{X}}$ and generate $\underline{\boldsymbol{Y}}_1$, $\underline{\boldsymbol{Y}}_2$, $\underline{\boldsymbol{Y}}_3$.
Case Slab-fiber sampling:
1) $\boldsymbol{A}, \boldsymbol{B} \leftarrow$ CPD($\underline{\boldsymbol{Y}}_2$)
2) $\boldsymbol{C} \leftarrow$ solve (8).
3) If $\|\underline{\boldsymbol{X}} - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]\|_F >$ threshold:
Solve (9) using BCD with exact line search.
Case multimode fiber sampling:
1) $\boldsymbol{C} \leftarrow$ CPD($\underline{\boldsymbol{Y}}_1$)
2) $\boldsymbol{A}, \boldsymbol{B} \leftarrow$ solve (10).
3) If $\|\underline{\boldsymbol{X}} - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]\|_F >$ threshold:
Solve (11) using BCD with exact line search.

V. Simulations

In this section we showcase the effectiveness of the proposed framework with simulated experiments involving synthetically generated and real tensors. All simulations are performed in MATLAB on a Linux server with 8 3.6GHz cores and 32GB RAM.

The baseline algorithms used for comparison are:

`CPD`: The CPD of the original tensor $\underline{\boldsymbol{X}}$ is computed using Tensorlab's CPD command [25]. The stopping criterion is maximum number of iterations equal to 50, which empirically are sufficient to give a good CPD fit.

`Randomized Block Sampling` (`RBS`) [17]: Tensorlab's implementation is being used and the algorithm is tested for different block sizes.

`Paracomp` [19]: Author's implementation is being used with three anchor rows between the compressed factors to reconcile for permutation and scaling mismatches. The CPD of the compressed tensors is performed with 50 iterations of Tensorlab's algorithm.

$\texttt{FIST}_1$, $\texttt{FIST}_2$: The two proposed approaches for slab-fiber and multi-fiber sampling respectively. We run step 1 with 50 iterations of Tensorlab's algorithm and the CPD stopping criterion is maximum number of iterations equal to 50. The threshold is set equal to $10^{-2}\|\underline{\boldsymbol{X}}\|_F$ and the stopping criterion for step 3 is maximum number of iterations equal to 5.

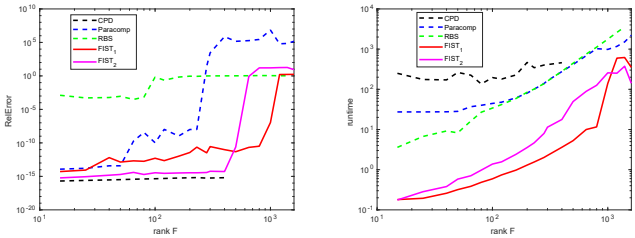
To assess the performance of each algorithm we measure the CPD relative error defined as:

$$\texttt{RelError} = \frac{\|\underline{\boldsymbol{X}} - [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]\|_F}{\|\underline{\boldsymbol{X}}\|_F},$$

where the subscript F is used to denote the frobenius norm of a tensor. We also measure the `runtime` of each algorithm.

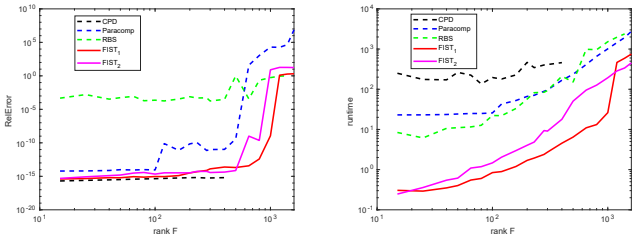
A. Synthetic experiments

The first set of experiments uses synthetically generated third-order tensors. In particular, we generate tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{1000\times1000\times1000}$ by randomly drawing the CPD factors $\boldsymbol{A} \in \mathbb{R}^{1000\times F}$, $\boldsymbol{B} \in \mathbb{R}^{1000\times F}$, $\boldsymbol{C} \in \mathbb{R}^{1000\times F}$ from a zero-mean unit-variance Gaussian distribution and synthesize the tensor as $\underline{\boldsymbol{X}} = [\![\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}]\!]$. We vary the rank F from 15 to 1600 and record the `RelError` and `runtime` for all the competing methods. Two scenarios are considered. In the first, the sampling/compression ratio $r = \frac{\#\text{measurements}}{IJK}$, for our method as well as `Paracomp`, is in the order of 10^{-3} and for the second in the order of 10^{-2}. Then for $\texttt{FIST}_1$ $K_2 = 2, 5$ for the two scenarios and $I_1 = J_1$ are chosen such that $I_1 J_1 > F + 10$. Regarding $\texttt{FIST}_2$ $I_1 = I_2 = J_1 = J_3 = 40, 50$ and $K_2 = K_3$ are chosen such that $I_1 K_2 > F + 10$. As far as `Paracomp` is concerned, the compressed subtensors, for the two scenarios, are chosen to be of size $50\times50\times50$ or $100\times100\times100$ and their number is $n = 22$ and $n = 11$ respectively, so that the final system is overdetermined. The block sizes in `RBS` are chosen equal to `Paracomp` for fair comparison. The performance of the competing methods for the two scenarios is presented in Fig. 3, 4 respectively. In terms of `RelError` $\texttt{FIST}_1$, $\texttt{FIST}_2$ work the best and `Paracomp` comes second (for small ranks). As far as `runtime` is concerned $\texttt{FIST}_1$ is the fastest, while



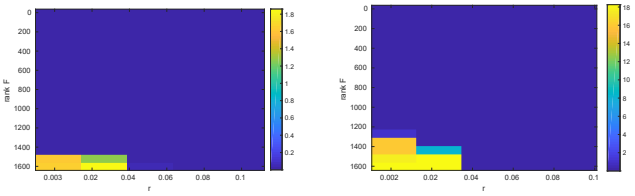
(a) `RelError` (b) `runtime`

Fig. 3: Scenario 1



(a) `RelError` (b) `runtime`

Fig. 4: Scenario 2



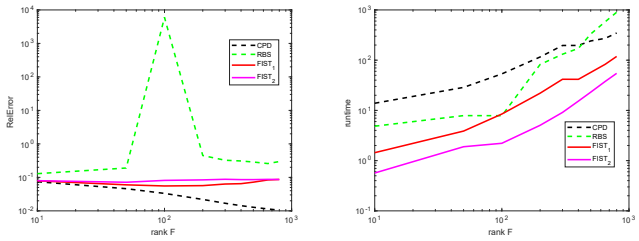
(a) $\texttt{FIST}_1$ (b) $\texttt{FIST}_2$

Fig. 5: F vs r

$\texttt{FIST}_2$ comes second. Note that both of them are at least an order of magnitude faster than the competing algorithms. The `RBS` algorithm exhibits a stable performance. We should also mention the direct CPD on the full tensors runs out of memory for rank greater than 500. We also vary the sampling ratio r from 0.002 to 0.1 for our proposed methods. Fig 5 presents the `RelError` for $\texttt{FIST}_1$ and $\texttt{FIST}_2$ as a function of F and r. The results show that the proposed methods work well for a wide range of ranks and sampling ratios.

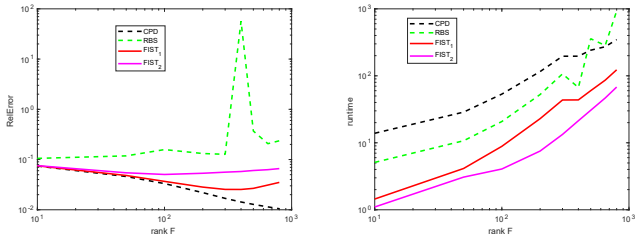
B. Real experiments

Finally we test the proposed approach with real data tensors. To this end we use the Cuprite hyperspectral image (HSI) from the AVIRIS platform [26], which is represented as a third order tensor $\underline{\boldsymbol{X}} \in \mathbb{R}^{512\times614\times187}$. Note that in HSI's factor $\boldsymbol{C}$ is generally ill-conditioned, due to the low rank matrix structure HSI's admit. In particular the condition number of Cuprite HSI for different ranks ranges from 10^4 to 10^8. We vary the rank from 10 to 800 and consider again two scenarios: In the first $I_1 = J_1 = 40$, $K_2 = 2$ for $\texttt{FIST}_1$ and $I_1 = I_2 = J_1 = J_3 = K_2 = K_3 = 50$ for $\texttt{FIST}_2$, whereas the blocksize of `RBS` is $10 \times 10 \times 10$. In the second scenario $I_1 = J_1 = 40$, $K_2 = 5$ for $\texttt{FIST}_1$, $I_1 = I_2 = J_1 = J_3 = 100$, $K_2 = K_3 = 50$ for $\texttt{FIST}_2$ and `RBS` block is $20 \times 20 \times 20$. Note that for `RBS` block sizes greater than 30 the runtime is worse than `CPD`. The



(a) `RelError` (b) `runtime`

Fig. 6: Real scenario 1



(a) `RelError` (b) `runtime`

Fig. 7: Real scenario 2

performance of `Paracomp` for all sizes and ranks was giving `RelError` greater than 10 and thus is omitted. The reason is that with real noisy data reconciling for permutation and scaling mismatches becomes very cumbersome. The results are presented in Fig. 6, 7. Same conclusions can be derived again. The proposed $\texttt{FIST}_1$, $\texttt{FIST}_2$ are faster and more accurate than the competitors.

VI. Conclusion

In this paper we studied the task of computing the CPD of large-scale tensors. We proposed two sampling mechanisms that operate on different modes of the tensor. The sampling is regular and does not need to follow any stochasticity. We also established the identifiability of the proposd model and developed an efficient algorithmic framework to handle the problem. Simulations with synthetic and real experiments showcase the effectiveness of the approach.

References

[1] X. Fu, N. D. Sidiropoulos, J. H. Tranter, and W.-K. Ma, "A factor analysis framework for power spectra separation and multiple emitter localization," *IEEE Transactions on Signal Processing*, vol. 63, no. 24, pp. 6581–6594, 2015.

[2] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 2773–2832, 2014.

[3] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 79–86.

[4] P. A. Traganitis and G. B. Giannakis, "Parafac-based multilinear subspace clustering for tensor data," in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Dec 2016, pp. 1280–1284.

[5] E. E. Papalexakis, N. D. Sidiropoulos, and R. Bro, "From k-means to higher-way co-clustering: Multilinear decomposition with sparse latent factors," *IEEE transactions on signal processing*, vol. 61, no. 2, pp. 493–506, 2013.

[6] B. W. Bader, R. A. Harshman, and T. G. Kolda, "Temporal analysis of semantic graphs using asalsan," in *Seventh IEEE international conference on data mining (ICDM 2007)*. IEEE, 2007, pp. 33–42.

[7] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 2, p. 16, 2017.

[8] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and W. Ma, "Hyperspectral super-resolution: A coupled tensor factorization approach," *IEEE Transactions on Signal Processing*, vol. 66, no. 24, pp. 6503–6517, Dec 2018.

[9] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and W.-K. Ma, "Hyperspectral super-resolution: Combining low rank tensor and matrix structure," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018, pp. 3318–3322.

[10] C. I. Kanatsoulis, X. Fu, N. D. Sidiropoulos, and M. Akçakaya, "Tensor completion from regular sub-Nyquist samples," *arXiv preprint arXiv:1903.00435*, 2019.

[11] N. D. Sidiropoulos and G. Z. Dimic, "Blind multiuser detection in w-cdma systems with large delay spread," *IEEE Signal Processing Letters*, vol. 8, no. 3, pp. 87–89, 2001.

[12] A. Carlson, J. Betteridge, B. Kisiel, B. Settles, E. R. Hruschka, and T. M. Mitchell, "Toward an architecture for never-ending language learning," in *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[13] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Parcube: Sparse parallelizable tensor decompositions," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2012, pp. 521–536.

[14] U. Kang, E. Papalexakis, A. Harpale, and C. Faloutsos, "Gigatensor: scaling tensor analysis up by 100 times-algorithms and discoveries," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2012, pp. 316–324.

[15] S. Smith, N. Ravindran, N. D. Sidiropoulos, and G. Karypis, "Splatt: Efficient and parallel sparse tensor-matrix multiplication," in *2015 IEEE International Parallel and Distributed Processing Symposium*. IEEE, 2015, pp. 61–70.

[16] B. W. Bader and T. G. Kolda, "Efficient matlab computations with sparse and factored tensors," *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 205–231, 2007.

[17] N. Vervliet and L. De Lathauwer, "A randomized block sampling approach to canonical polyadic decomposition of large-scale tensors," *IEEE Journal of Selected Topics in Signal Processing*, vol. 10, no. 2, pp. 284–295, 2016.

[18] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM journal on Matrix Analysis and Applications*, vol. 21, no. 4, pp. 1253–1278, 2000.

[19] N. D. Sidiropoulos, E. E. Papalexakis, and C. Faloutsos, "Parallel randomly compressed cubes: A scalable distributed architecture for big tensor decomposition," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 57–70, 2014.

[20] B. Yang, A. Zamzam, and N. D. Sidiropoulos, "Parasketch: Parallel tensor factorization via sketching," in *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 2018, pp. 396–404.

[21] N. Vervliet, O. Debals, L. Sorber, and L. De Lathauwer, "Breaking the curse of dimensionality using decompositions of incomplete tensors: Tensor-based scientific computing in big data analysis," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 71–79, 2014.

[22] N. D. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3551–3582, 2017.

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

[24] J. B. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear algebra and its applications*, vol. 18, no. 2, pp. 95–138, 1977.

[25] N. Vervliet, O. Debals, L. Sorber, M. Van Barel, and L. De Lathauwer, "Tensorlab v3. 0, march 2016," *URL: http://www. tensorlab. net*.

[26] G. Vane, R. O. Green, T. G. Chrien, H. T. Enmark, E. G. Hansen, and W. M. Porter, "The airborne visible/infrared imaging spectrometer (aviris)," *Remote sensing of environment*, vol. 44, no. 2-3, pp. 127–143, 1993.