Distributed Large-Scale Tensor Decomposition

André L. F. De Almeida, Alain Y. Kibangou

DISTRIBUTED LARGE-SCALE TENSOR DECOMPOSITION

André L. F. de Almeida[(1)]*, Alain Y. Kibangou*[(2)]*,*
[(1)]Department of Teleinformatics Engineering, Federal University of Ceará, Fortaleza, Brazil
[(2)] GIPSA-Lab, University Joseph Fourier, CNRS, Saint Martin d'Hères, France
E-mails: andre.almeida@ieee.org, alain.kibangou@ujf-grenoble.fr

ABSTRACT

Canonical Polyadic Decomposition (CPD), also known as PARAFAC, is a useful tool for tensor factorization. It has found application in several domains including signal processing and data mining. With the deluge of data faced in our societies, large-scale matrix and tensor factorizations become a crucial issue. Few works have been devoted to large-scale tensor factorizations. In this paper, we introduce a fully distributed method to compute the CPD of a large-scale data tensor across a network of machines with limited computation resources. The proposed approach is based on collaboration between the machines in the network across the three modes of the data tensor. Such a multi-modal collaboration allows an essentially unique reconstruction of the factor matrices in an efficient way. We provide an analysis of the computation and communication cost of the proposed scheme and address the problem of minimizing communication costs while maximizing the use of available computation resources.

Index Terms— Tensor decompositions, large-scale data, distributed computation.

1. INTRODUCTION

From Internet to large research infrastructures, the volume of data generated by our societies is continuously increasing. A deluge faced by the producers of these data as well as their users. The big data issue is a significant scientific challenge that requires deep investigations in both engineering and fundamental science. Everyone is concerned and it is urgent to get answers to questions such as how to store these huge amount of data? How to process and analyze them?

Low-rank matrix factorization has received a particular attention in recent years, since it is fundamental to a variety of mining tasks that are increasingly being applied to massive datasets. In large applications, matrix factorizations can involve matrices with billions of entries. At this massive scale, distributed algorithms for matrix factorization are essential to achieve reasonable performance [2]. However, in many disciplines, data inherently has more than two axes of variation and can be arranged as tensors (i.e. multi-way arrays). Computing tensor decompositions of multi-way datasets is particularly useful to extract hidden patterns and structure in data analytics problems. Specifically, CPD (Canonical Polyadic Decomposition) also known as PARAFAC (Parallel factor analysis) is an extension of a low rank matrix decomposition to tensors.

In order to compute CPD, several algorithms have been proposed in the literature, which can be classified into three main categories: alternating algorithms, derivative based algorithms, and non-iterative algorithms (see e.g. [3, 4, 5]). Recently, considering that the data tensor can be serially acquired or the underlying process can be time-varying, adaptive algorithms have been proposed in [6]. And more recently, assuming that the data tensor is spatially spread out, distributed algorithms have been introduced in [7] and [8]. These algorithms are based on an alternating least square (ALS) algorithm that proceeds iteratively and minimizes a criterion (that is usually quadratic) with respect to individual factors one by one. Then, distributed algorithms are obtained using the concept of average consensus that has been extensively studied in computer science and is a central topic for load balancing (with divisible tasks) in parallel computers [9].

The need for large-scale tensor computations is increasing and there is a huge gap to be filled. In contrast to large-scale matrix factorization, very few works are devoted to large-scale tensors. Two ways have been recently considered in the literature. The first one consists in exploiting sparseness of tensors. In [10], the GigaTensor algorithm is designed in order to minimize the number of floating point operations and to handle the size of intermediate data in order to overcome the intermediate data explosion problem. For large-scale tensors the intermediate data explosion problem arises when Khatri-Rao matrix products are implemented in a naive way. In [10] a smart ordering of computations is proposed and explicit computation of Khatri-Rao matrix product is avoided. In addition, a way to implement the proposed scheme in the distributed computing frameworks MapReduce and Hadoop is proposed by the authors and then, from extensive simulations, linear scalability on the number of machines is claimed. Note that the data explosion problem can also be handled by exploiting sparseness of both data and latent factors as in [11, 12, 13].

The second class of methods consists in a subdivision of the large-scale tensor into smaller ones. Then factor matrices are reconstituted from the estimated sub-factors. The work in [14] is motivated by the success of random sampling-based matrix algorithms. The large-scale tensor is under-sampled several times, then the different sub-tensors are processed in parallel and eventually the results are combined in a clever way. In order to overcome permutation indeterminacies, all the different sub-tensors are enforced to overlap in a common set of indices in all the three modes of the tensor. The merging operation is guaranteed to be successful only if the random samples (sub-tensors) fulfill CPD identifiability conditions. In [1], sub-division of the original tensor is achieved in a deterministic way; a grid decomposition is proposed. CPD is first performed for each sub-tensor in order to get a an alternative representation that allows avoiding Khatri-Rao matrix products. Then, grid-CPD performs as a decentralized computation method where at each step parallel machines have to communicate with a central server. There is no direct collaboration between the involved machines. In [15], the authors have introduced the notion

This work is partially supported by the CNPq and the FUNCAP-INRIA project TeMP.

of collaboration in the grid. In this set-up, the computation is really distributed. The central server only initiates the process by sending data to the machines in the network and merges the results at the end of the computation process. During the computation process, the central server can allocate its own resources to other tasks. Moreover, thanks to this full collaboration and by introducing overlapping between sub-tensors, uniqueness of the decomposition with indeterminacies as in the standard CPD can be guaranteed. However, collaboration was restricted to two modes and overlapping induced loss of efficiency (increase of the dimensions and the number of sub-tensors).

This paper introduces a new way for computing a grid-CPD of a large-scale data tensor that ensures essential uniqueness and perfect reconstruction of the factor matrices. The main idea of the proposed approach is to allow collaboration between the machines in the network across the three modes of the tensor. To cope with the multi-way nature of the data, multi-graphs are used to model the network. Due to direct collaboration between machines, fulfilling CPD uniqueness conditions is not required for each sub-tensor. We provide an analysis of the computation and communication cost of the proposed scheme and address the problem of minimizing communication costs while maximizing the use of available computation resources.

Notations: Vectors are written as boldface lower-case letters ($\mathbf{a}$,$\mathbf{b}$,$\cdots$), matrices as boldface upper-case letters ($\mathbf{A}$,$\mathbf{B}$,$\cdots$), and tensor as calligraphic letters ($\mathcal{X},\mathcal{Y},\cdots$). $\mathbf{A}^T$ stands for the transpose of $\mathbf{A}$ whereas $\mathbf{A}^H$ stands for its complex conjugate. The operator $\circ$ denotes the outer product between vectors, while $\odot$ stands for the Khatri-Rao (columnwise Kronecker) product. The Hadamard (element-wise) product is denoted by the $*$ symbol. For a three-way tensor $\mathcal{X} \in \mathbb{C}^{I\times J\times K}$, I, J and K are referred to as the mode-1, mode-2 and mode-3 dimensions of $\mathcal{X}$, respectively. The operator $\sqcup_1$ denotes the mode-1 concatenation of any two tensors $\mathcal{X}$ and $\mathcal{Y}$ having the same mode-2 and mode-3 dimensions. This operation is similarly defined to denote concatenation along the other modes.

2. PROBLEM SETTING

Consider a large-scale tensor $\mathcal{X} \in \mathbb{C}^{I\times J\times K}$, i.e. a tensor with a total number of entries IJK that can easily be of order of billions. The CPD of $\mathcal{X}$, denoted by $\mathcal{X} = [\![\mathbf{A},\mathbf{B},\mathbf{C}]\!]$, is given by [16, 17]:

$$\mathcal{X} = \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r, \tag{1}$$

where $\mathbf{A} = [\mathbf{a}_1,\ldots,\mathbf{a}_R] \in \mathbb{C}^{I\times R}$, $\mathbf{B} = [\mathbf{b}_1,\ldots,\mathbf{b}_R] \in \mathbb{C}^{J\times R}$ and $\mathbf{C} = [\mathbf{c}_1,\ldots,\mathbf{c}_R] \in \mathbb{C}^{K\times R}$ are the factor matrices associated with the CPD of $\mathcal{X}$, and R denotes the *tensor rank*. A sufficient condition for the essential uniqueness of the CPD was first established by Kruskal in [18], and states that $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ can be uniquely estimated (up to column permutation and scaling) if $k_{\mathbf{A}}+k_{\mathbf{B}}+k_{\mathbf{C}} \geq 2R+2$, where $k_{(\cdot)}$ denotes the Kruskal-rank, or "k-rank", of its matrix argument. Shortly, $k_{(\mathbf{A})}$ is equal to r if every set of r columns of $\mathbf{A}$ is linearly independent. In what follows, we work under the assumption that CPD uniqueness for $\mathcal{X}$ holds. Usually, the matrices $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ can be estimated by computing an approximation to the rank-R CPD of $\mathcal{X}$, i.e.:

$$\min_{\mathbf{A},\mathbf{B},\mathbf{C}} \Big\| \mathcal{X} - \sum_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \Big\|_F^2. \tag{2}$$

The traditional way for carrying out such a computation is to resort to an alternating least squares algorithm (ALS) where estimations of $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ are obtained by alternatively solving three different linear least squares (LS) problems. Despite its conceptual simplicity, ALS involved Khatri-Rao matrix products and multiplications of matrices that cannot be handled when the tensor exhibits large dimensions since computation and storage resource can easily become insufficient. Apart from complexity issues, ALS-based algorithms present a very slow convergence due to the huge number of unknowns involved in the estimation of the large factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$.

Consider a central server having at its disposal a large-scale data tensor $\mathcal{X}$ to be factorized. This server is connected to a dense network of L machines (cores or independent computers) with limited processing powers. Our goal is to perform the CPD of $\mathcal{X}$ across the network by using all the in-network computation resources in an optimal way while making the central server available to other tasks. The central server and the L machines, where computations are achieved, are nodes of a network connected through reliable high speed data links, so that data exchanges may be considered noise-free. The central server first subdivides the tensor and assigns each sub-tensor to a given machine. Once computation is achieved, the L machines send their results back to the central server where global factor matrices are built. In contrast to the grid-CPD in [1], the central server is not involved in the computation process and available to perform other tasks at the same time. The proposed set-up is therefore fully distributed while that of [1] is simply decentralized.

3. PROPOSED SOLUTION

The central server generates different (possibly overlapped) data sub-tensors $\mathcal{X}^{(\ell_1,\ell_2,\ell_3)} \in \mathbb{C}^{I_{\ell_1}\times J_{\ell_2}\times K_{\ell_3}}$, $\ell_1 = 1,\ldots,L_1$, $\ell_2 = 1,\ldots,L_2$, $\ell_3 = 1,\ldots,L_3$, of much smaller dimensions, by partitioning $\mathcal{X}$ in a dense 3-D tensor grid. The dimensions of the sub-tensors are related as follows: $I_1 + \ldots + I_{L_1} \geq I$, $J_1+\ldots+J_{L_2} \geq J$, $K_1+\ldots+K_{L_3} \geq K$. Equality in a given mode occurs if and only if the sub-tensors do not overlap in that mode. The sub-tensors $\mathcal{X}^{(\ell_1,\ell_2,\ell_3)}$ can be concatenated along different pair of modes to form mode-1, mode-2, and mode-3 sub-tensors $\mathcal{X}_1^{(\ell_1)} \in \mathbb{C}^{I_{\ell_1}\times J\times K}$, $\mathcal{X}_2^{(\ell_2)} \in \mathbb{C}^{I\times J_{\ell_2}\times K}$, $\mathcal{X}_3^{(\ell_3)} \in \mathbb{C}^{I\times J\times K_{\ell_3}}$, which are defined, respectively, as

$$\mathcal{X}_1^{(\ell_1)} = \big[\mathcal{X}_1^{(\ell_1,1)} \sqcup_3 \cdots \sqcup_3 \mathcal{X}_1^{(\ell_1,L_3)}\big] \tag{3}$$
$$\text{where} \quad \mathcal{X}_1^{(\ell_1,\ell_3)} = \big[\mathcal{X}^{(\ell_1,1,\ell_3)} \sqcup_2 \cdots \sqcup_2 \mathcal{X}^{(\ell_1,L_2,\ell_3)}\big],$$
$$\mathcal{X}_2^{(\ell_2)} = \big[\mathcal{X}_2^{(\ell_2,1)} \sqcup_1 \cdots \sqcup_1 \mathcal{X}_2^{(\ell_2,L_1)}\big] \tag{4}$$
$$\text{where} \quad \mathcal{X}_2^{(\ell_2,\ell_1)} = \big[\mathcal{X}^{(\ell_1,\ell_2,1)} \sqcup_3 \cdots \sqcup_3 \mathcal{X}^{(\ell_1,\ell_2,L_3)}\big],$$
$$\mathcal{X}_3^{(\ell_3)} = \big[\mathcal{X}_3^{(\ell_3,1)} \sqcup_2 \cdots \sqcup_2 \mathcal{X}_3^{(\ell_3,L_2)}\big] \tag{5}$$
$$\text{where} \quad \mathcal{X}_3^{(\ell_3,\ell_2)} = \big[\mathcal{X}^{(1,\ell_2,\ell_3)} \sqcup_1 \cdots \sqcup_1 \mathcal{X}^{(L_1,\ell_2,\ell_3)}\big].$$

An interesting fact resulting from the CPD trilinearity is as follows: if the CPD of $\mathcal{X}$ is essentially unique and provided that $k_{\mathbf{A}^{(\ell_1)}} = k_{\mathbf{A}}$, $k_{\mathbf{B}^{(\ell_2)}} = k_{\mathbf{B}}$, and $k_{\mathbf{C}^{(\ell_3)}} = k_{\mathbf{C}}$, each mode-$i$ sub-tensor, $i = 1,2,3$, admits an exact CPD given by:

$$\mathcal{X}_1^{(\ell_1)} = [\![\mathbf{A}^{(\ell_1)},\mathbf{B},\mathbf{C}]\!], \quad \mathcal{X}_2^{(\ell_2)} = [\![\mathbf{A},\mathbf{B}^{(\ell_2)},\mathbf{C}]\!]$$
$$\mathcal{X}_3^{(\ell_3)} = [\![\mathbf{A},\mathbf{B},\mathbf{C}^{(\ell_3)}]\!]. \tag{6}$$

Now, the central server assigns a sub-tensor $\mathcal{X}^{(\ell_1,\ell_2,\ell_3)}$ to each of the L available machines. To guarantee a one-to-one

mapping between the sub-tensors and the machines, each machine is uniquely labeled by the triad (ℓ_1, ℓ_2, ℓ_3) and is associated with the corresponding sub-tensor $\mathcal{X}^{(\ell_1,\ell_2,\ell_3)}$. Therefore the total number L of machines is equal to $L_1L_2L_3$. According to (6), from their respective sub-tensors, the machines having the same i-th coordinate, $i = 1, 2, 3$, can collaborate to estimate their common sub-factor matrix in mode-i. Therefore the adjacency relations between the machines evolve according to the mode of interest. To capture this behavior, we adopt multi-layer graphs, or simply multi-graphs, for representing the network [19, 20]. For this purpose, let $\mathbb{G}$ be a multi-layer graph which contains 3 individual graph layers $\mathbb{G}^{(i)}$, $i = 1, 2, 3$, where each layer $\mathbb{G}^{(i)}$ is a undirected graph consisting of a common vertex set V, with cardinality L, and a specific edge set $E^{(i)}$. Each layer is associated with a mode of the tensor to be decomposed. In addition, each layer is constituted with L_i connected components of L/L_i vertices, each connected component being a star graph[1]. More precisely, the nodes labeled (l_1, l_2, l_3) and (l'_1, l'_2, l'_3) belongs to the same component of $\mathbb{G}^{(i)}$ if and only if $l_i = l'_i$. They are adjacent only if one of them is an internal node and the other is a leaf of a tree. Therefore, each layer can be viewed as a set of parallel star sub-networks, each star sub-network constituting a connected component of the graph. As it is common for tensor decomposition, data processing is to be carried out in an alternating way. The layers of the multi-graph are to be considered one after the other. In time, the network topology switches periodically between parallel star subnetworks (see Fig. 1).

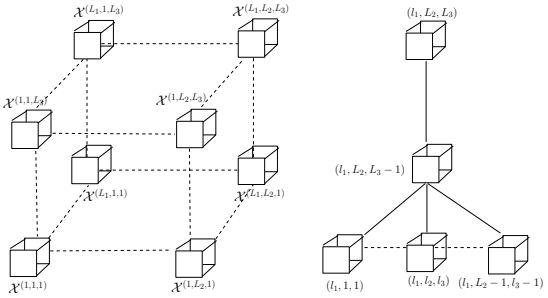

Fig. 1. Left: Subdivision of a large-scale tensor in small sub-tensors $\mathcal{X}^{(l_1,l_2,l_3}$ each one being associated to a machine labeled (ℓ_1, ℓ_2, ℓ_3). Right: ℓ_1-th star subnetwork of layer 1 associated to the tensor mode 1.

By exploiting the degrees of freedom provided by the three layers $\mathbb{G}^{(i)}$, $i = 1, 2, 3$, of the multi-graph $\mathbb{G}$, the problem consists in finding $\mathbf{A}^{(\ell_1)}$, $\ell_1 = 1, \ldots, L_1$, $\mathbf{B}^{(\ell_2)}$, $\ell_2 = 1, \ldots, L_2$, and $\mathbf{C}^{(\ell_3)}$, $\ell_3 = 1, \ldots, L_3$ by solving the three following sets of LS problems:

$$\min_{\mathbf{A}^{(\ell_1)}} \left\| \mathcal{X}_1^{(\ell_1)} - [\![\mathbf{A}^{(\ell_1)}, \mathbf{B}, \mathbf{C}]\!] \right\|_F^2, \ \ell_1 = 1, \ldots, L_1, \quad (7)$$

$$\min_{\mathbf{B}^{(\ell_2)}} \left\| \mathcal{X}_2^{(\ell_2)} - [\![\mathbf{A}, \mathbf{B}^{(\ell_2)}, \mathbf{C}]\!] \right\|_F^2, \ \ell_2 = 1, \ldots, L_2, \quad (8)$$

$$\min_{\mathbf{C}^{(\ell_3)}} \left\| \mathcal{X}_3^{(\ell_3)} - [\![\mathbf{A}, \mathbf{B}, \mathbf{C}^{(\ell_3)}]\!] \right\|_F^2, \ \ell_3 = 1, \ldots, L_3. \quad (9)$$

Each set of LS problems is associated with disjoint subsets of machines and therefore can be solved in parallel. For instance, the L_1 LS problems in mode-1 are solved independently by L_1 subsets of machines working in parallel. The same applies to the two other LS problems. Moreover, fast computations can be performed on small sub-tensors at each machine.

Let us define $\mathbf{X}_{(1)}^{(\ell_1,\ell_2,\ell_3)} \in \mathbb{C}^{J_{\ell_2}K_{\ell_3} \times I_{\ell_1}}$, $\mathbf{X}_{(2)}^{(\ell_1,\ell_2,\ell_3)} \in \mathbb{C}^{K_{\ell_3}I_{\ell_1} \times J_{\ell_2}}$, and $\mathbf{X}_{(3)}^{(\ell_1,\ell_2,\ell_3)} \in \mathbb{C}^{I_{\ell_1}J_{\ell_2} \times K_{\ell_3}}$ as the matrices obtained by unfolding ("matricization" of) the sub-tensors $\mathcal{X}_1^{(\ell_1)}$, $\mathcal{X}_2^{(\ell_2)}$ and $\mathcal{X}_3^{(\ell_3)}$ along mode-1, mode-2 and mode-3, respectively. Note that these matrices collect different rearrangements of the same data contained in the sub-tensor $\mathcal{X}^{(\ell_1,\ell_2,\ell_3)}$. With these definitions, we can recast each one of the LS problems (7)-(9) as summations over smaller LS subproblems, as follows:

$$\mathbb{J}_{\mathbf{A}^{(\ell_1)}} = \min_{\mathbf{A}^{(\ell_1)}} \sum_{\ell_2=1}^{L_2} \sum_{\ell_3=1}^{L_3} \underbrace{\left\| \mathbf{X}_{(1)}^{(\ell_1,\ell_2,\ell_3)} - (\mathbf{B}^{(\ell_2)} \odot \mathbf{C}^{(\ell_3)})\mathbf{A}^{(\ell_1)T} \right\|_F^2}_{\mathbb{J}_{\mathbf{A}^{(\ell_1)}}^{(\ell_2,\ell_3)}},$$

$$\mathbb{J}_{\mathbf{B}^{(\ell_2)}} = \min_{\mathbf{B}^{(\ell_2)}} \sum_{\ell_1=1}^{L_1} \sum_{\ell_3=1}^{L_3} \underbrace{\left\| \mathbf{X}_{(2)}^{(\ell_1,\ell_2,\ell_3)} - (\mathbf{C}^{(\ell_3)} \odot \mathbf{A}^{(\ell_1)})\mathbf{B}^{(\ell_2)T} \right\|_F^2}_{\mathbb{J}_{\mathbf{B}^{(\ell_2)}}^{(\ell_1,\ell_3)}},$$

$$\mathbb{J}_{\mathbf{C}^{(\ell_3)}} = \min_{\mathbf{C}^{(\ell_3)}} \sum_{\ell_1=1}^{L_1} \sum_{\ell_2=1}^{L_2} \underbrace{\left\| \mathbf{X}_{(3)}^{(\ell_1,\ell_2,\ell_3)} - (\mathbf{A}^{(\ell_1)} \odot \mathbf{B}^{(\ell_2)})\mathbf{C}^{(\ell_3)T} \right\|_F^2}_{\mathbb{J}_{\mathbf{C}^{(\ell_3)}}^{(\ell_1,\ell_2)}}.$$

The solution that minimizes $\mathbb{J}_{\mathbf{A}^{(\ell_1)}}$ (resp. $\mathbb{J}_{\mathbf{B}^{(\ell_2)}}$ and $\mathbb{J}_{\mathbf{A}^{(\ell_3)}}$) can then be written as:

$$\mathbf{A}^{(\ell_1)T} = \left(\frac{1}{L_2L_3} \sum_{\ell_2,\ell_3} \mathbf{\Gamma}_{\mathbf{A}^{(\ell_1)}}^{(\ell_2,\ell_3)} \right)^{-1} \left(\frac{1}{L_2L_3} \sum_{\ell_2,\ell_3} \mathbf{\Psi}_{\mathbf{A}^{(\ell_1)}}^{(\ell_2,\ell_3)} \right) = \left(\overline{\mathbf{\Gamma}}_{\mathbf{A}^{(\ell_1)}} \right)^{-1} \overline{\mathbf{\Psi}}_{\mathbf{A}^{(\ell_1)}} \quad (10)$$

$$\mathbf{B}^{(\ell_2)T} = \left(\frac{1}{L_1L_3} \sum_{\ell_1,\ell_3} \mathbf{\Gamma}_{\mathbf{B}^{(\ell_2)}}^{(\ell_1,\ell_3)} \right)^{-1} \left(\frac{1}{L_1L_3} \sum_{\ell_1,\ell_3} \mathbf{\Psi}_{\mathbf{B}^{(\ell_2)}}^{(\ell_1,\ell_3)} \right) = \left(\overline{\mathbf{\Gamma}}_{\mathbf{B}^{(\ell_2)}} \right)^{-1} \overline{\mathbf{\Psi}}_{\mathbf{B}^{(\ell_2)}} \quad (11)$$

$$\mathbf{C}^{(\ell_3)T} = \left(\frac{1}{L_1L_2} \sum_{\ell_1,\ell_2} \mathbf{\Gamma}_{\mathbf{C}^{(\ell_3)}}^{(\ell_1,\ell_2)} \right)^{-1} \left(\frac{1}{L_1L_2} \sum_{\ell_1,\ell_2} \mathbf{\Psi}_{\mathbf{C}^{(\ell_3)}}^{(\ell_1,\ell_2)} \right) = \left(\overline{\mathbf{\Gamma}}_{\mathbf{C}^{(\ell_3)}} \right)^{-1} \overline{\mathbf{\Psi}}_{\mathbf{C}^{(\ell_3)}} \quad (12)$$

where $\overline{\mathbf{\Gamma}}_{\mathbf{A}^{(\ell_1)}}^{(\ell_2,\ell_3)} = \mathbf{B}^{(\ell_2)H}\mathbf{B}^{(\ell_2)} * \mathbf{C}^{(\ell_3)H}\mathbf{C}^{(\ell_3)} \in \mathbb{C}^{R \times R}$, $\overline{\mathbf{\Gamma}}_{\mathbf{B}^{(\ell_2)}}^{(\ell_1,\ell_3)} = \mathbf{C}^{(\ell_3)H}\mathbf{C}^{(\ell_3)H} * \mathbf{A}^{(\ell_1)H}\mathbf{A}^{(\ell_1)} \in \mathbb{C}^{R \times R}$, $\overline{\mathbf{\Gamma}}_{\mathbf{C}^{(\ell_3)}}^{(\ell_1,\ell_2)} = \mathbf{A}^{(\ell_1)H}\mathbf{A}^{(\ell_1)H} * \mathbf{B}^{(\ell_2)H}\mathbf{B}^{(\ell_2)} \in \mathbb{C}^{R \times R}$, $\overline{\mathbf{\Psi}}_{\mathbf{A}^{(\ell_1)}}^{(\ell_2,\ell_3)} = (\mathbf{B}^{(\ell_2)} \odot \mathbf{C}^{(\ell_3)})^H \mathbf{X}_{(1)}^{(\ell_1,\ell_2,\ell_3)} \in \mathbb{C}^{R \times I_{\ell_1}}$, $\overline{\mathbf{\Psi}}_{\mathbf{B}^{(\ell_2)}}^{(\ell_1,\ell_3)} = (\mathbf{C}^{(\ell_3)} \odot \mathbf{A}^{(\ell_1)})^H \mathbf{X}_{(2)}^{(\ell_1,\ell_2,\ell_3)} \in \mathbb{C}^{R \times J_{\ell_2}}$, and $\overline{\mathbf{\Psi}}_{\mathbf{C}^{(\ell_3)}}^{(\ell_1,\ell_2)} = (\mathbf{A}^{(\ell_1)} \odot \mathbf{B}^{(\ell_2)})^H \mathbf{X}_{(3)}^{(\ell_1,\ell_2,\ell_3)} \in \mathbb{C}^{R \times K_{\ell_3}}$.

The estimation of $\{\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(L_1)}\}$, $\{\mathbf{B}^{(1)}, \ldots, \mathbf{B}^{(L_2)}\}$, and $\{\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(L_3)}\}$ is carried out in three steps. First, the L_1 connected components of $\mathbb{G}^{(1)}$ operate in parallel to estimate $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(L_1)}$. Second, the L_2 connected components of $\mathbb{G}^{(2)}$ operate in parallel to estimate $\mathbf{B}^{(1)}, \ldots, \mathbf{B}^{(L_2)}$. Finally, the L_3 connected components of $\mathbb{G}^{(3)}$ operate in parallel to estimate $\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(L_3)}$. Thanks to the star topology adopted herein, the averaging operations involved in (10), (11), and (12) can be carried

[1]The choice of a star topology is not restrictive. According to the available resource any other kind of topology can be adopted.

out in two steps[2]. First, the leaves of the star sub-network send their data to the internal node of the corresponding sub-network and then the averaged quantities are sent back to the leaves.

Identifiability issues: Strictly local computations of a CPD of the (ℓ_1, ℓ_2, ℓ_3)-th node requires satisfying the following necessary condition for the identifiability of the triplet $\{\mathbf{A}^{(\ell_1)}, \mathbf{B}^{(\ell_2)}, \mathbf{C}^{(\ell_3)}\}$ at each node [23]:

$$\min(J_{\ell_2}K_{\ell_3}, I_{\ell_1}K_{\ell_3}, I_{\ell_1}J_{\ell_2}) \geq R. \tag{13}$$

This condition may not always hold, especially for small sub-tensors where the dimensions are too small compared to the number R of hidden factors to be identified from the global data tensor. By allowing collaboration along the three modes, identifiability conditions are improved by relaxing the necessary constraints on the sub-tensor dimensions at each node. In this case, the necessary condition turns out to be

$$\min\Big(\sum_{\ell_2,\ell_3} J_{\ell_2}K_{\ell_3}, \sum_{\ell_1,\ell_3} I_{\ell_1}K_{\ell_3}, \sum_{\ell_1,\ell_2} I_{\ell_1}J_{\ell_2}\Big) \geq R, \tag{14}$$

which is clearly less restrictive than the previous one. Moreover, thanks to the multi-mode cooperation between the machines in the proposed scheme, permutation and scaling indeterminacies are similar to those occurring in the standard CPD. Thus, the reconstruction of the global factor matrices at the central server can be done without resorting to additional structural constraints such as anchor [14] or overlapping [15] sub-tensors.

Communication and computation costs: The overall computation cost of the proposed scheme is similar to that of the centralized scheme (standard ALS applied to the large-scale tensor). However, the cost per machine can be significantly lowered. For instance, for a large cubic tensor ($I = J = K, I > 10^2$) and considering a uniform sampling of the tensor so that $I_l = J_l = K_l = mI$, with $m < 1$, the communication and computation costs are given in Table 1:

Table 1. Complexity evaluation per machine and per iteration for an $I \times I \times I$ tensor of rank R.

Method	Computation cost	Communication cost
Centralized	$6RI^3$	-
Distributed	$6mRI^3$	$3R^2 + 3mRI$

Now let us define by q the ratio between the power required for achieving one floating operation and that for transmitting one real value. Assuming that $q << I^2$, the ratio between the per machine consumptions E_d and E_c in the distributed and the centralized schemes, respectively, is given by: $\frac{E_d}{E_c} \approx (1 + \frac{q}{2I^2})m \approx m < 1$. This means that the reduction of power consumption per machine follows the reduction of the tensor dimensions.

From Table 1, we can note that the additional power consumption is due to communications. Indeed at each ALS iteration, each machine (ℓ_1, ℓ_2, ℓ_3) has to transmit three $R \times R$ matrices ($\overline{\mathbf{\Gamma}}_{\mathbf{A}^{(\ell_1)}}$, $\overline{\mathbf{\Gamma}}_{\mathbf{B}^{(\ell_2)}}$, $\overline{\mathbf{\Gamma}}_{\mathbf{C}^{(\ell_3)}}$) and three rectangular matrices ($\overline{\mathbf{\Psi}}_{\mathbf{A}^{(\ell_1)}} \in \mathbb{C}^{R\times I_{\ell_1}}$,$\overline{\mathbf{\Psi}}_{\mathbf{B}^{(\ell_2)}} \in \mathbb{C}^{R\times J_{\ell_2}}$, $\overline{\mathbf{\Psi}}_{\mathbf{C}^{(\ell_3)}} \in \mathbb{C}^{R\times K_{\ell_3}}$). The overall communication cost per iteration is then given by:

$$\mathcal{C} = 3R^2L + R\sum_{\ell_1,\ell_2,\ell_3}(I_{\ell_1} + J_{\ell_2} + K_{\ell_3}).$$

[2]For arbitrary topologies, exact averaged quantities can be obtained using the finite-time average consensus protocol designed in [21]. In particular, it has been shown that consensus can be achieved in two steps when the topology is restricted to be strongly regular [22].

We can conclude that the communication cost grows linearly with the number of machines, meaning that the proposed scheme is scalable in terms of communication cost. Moreover, by appropriately selecting the dimensions of the sub-tensors, one can significantly reduce the communication cost per node.

Optimizing the sub-tensors' dimensions: Given a number L of available machines, one have to minimize the communication cost per machine so that the necessary identifiability conditions are fulfilled. The dimensions of the sub-tensors and the parameters L_i are obtained by solving the following integer programming problem:

$$\begin{aligned}
&\text{minimize } \tfrac{1}{L_1L_2L_3}\sum_{\ell_1,\ell_2,\ell_3}(I_{\ell_1} + J_{\ell_2} + K_{\ell_3})\\
&\text{s.t.} \sum_{\ell_1} I_{\ell_1} \geq I,\ \sum_{\ell_2} J_{\ell_2} \geq J,\ \sum_{\ell_3} K_{\ell_3} \geq K,\ L_1L_2L_3 \leq L\\
&\sum_{\ell_2,\ell_3} J_{\ell_2}K_{\ell_3} \geq R,\ \sum_{\ell_1,\ell_3} I_{\ell_1}K_{\ell_3} \geq R,\ \sum_{\ell_1,\ell_2} I_{\ell_1}J_{\ell_2} \geq R.
\end{aligned}$$

This problem is NP-hard. A more tractable solution can be obtained if we adopt a uniform sampling for each mode, i.e. $I_{\ell_1} = m_1I$, $J_{\ell_2} = m_2J$, and $K_{\ell_3} = m_3K$. Table 2 gives the optimal configuration for a cubic tensor obtained by solving the above problem relaxed to the case of uniform sampling for each mode.

Table 2. Optimal number and dimensions of sub-tensors for different numbers of available machines for a $10^3 \times 10^3 \times 10^3$ tensor ($\mathcal{C}_m$= communication cost per machine; $\mathcal{R}$= rate of machines used).

♯ machines	Sub-tensor size	(L_1, L_2, L_3)	$\mathcal{C}_m$	$\mathcal{R}$ (%)
20	$500 \times 334 \times 334$	$(2, 3, 3)$	1468	90
40	$334 \times 334 \times 250$	$(3, 3, 4)$	1218	90
60	$334 \times 250 \times 200$	$(3, 4, 5)$	1084	100
80	$250 \times 250 \times 200$	$(4, 4, 5)$	1000	100
100	$250 \times 200 \times 200$	$(4, 5, 5)$	950	100

Note that as the number of available machines increases a more efficient use of the resources is obtained and the communication cost per node is reduced, corroborating the relevance of optimizing the sampling of the data tensor across the three modes.

4. CONCLUSION

Performing CPD for large-scale tensors is particularly challenging. In this paper, we have proposed a fully distributed method that allows computing CPD across a network of machines with limited computation resources. In the proposed scheme, the central server does not interfere in the computation process and can achieve other tasks at the same time. The machines in the network collaborate with each other according to the three modes of the tensor. Such a multi-modal collaboration is captured by a multi-graph whose layers have several connected components. Due to this full collaboration, improved necessary identifiability conditions are achieved and the global factor matrices can be reconstructed in an efficient way. However, a communication overhead is introduced. Hopefully, the communication cost grows linearly with the number of machines and the dimensions of the sub-tensors. Additionally, optimizing the sampling of the tensor across the three modes yields a significant reduction of this overhead while maximizing the use of the available computation resources. Extension to tensors of higher orders is straightforward although the proposed optimization problem becomes more challenging. Future works include resources optimization for heterogeneous networks of machines having different computation capabilities, extension to other tensor factorization, and exploitation of sparsity and structural properties of the factor matrices.

5. REFERENCES

[1] A. H. Phan and A. Cichocki, "PARAFAC algorithms for large-scale problems," *Neurocomputing*, vol. 74, no. 11, pp. 1970 – 1984, 2011.

[2] R. Gemulla, P.J. Haas, E. Nijkamp, and Y. Sismanis, "Large-Scale Matrix Factorization with Distributed Stochastic Gradient Descent", *Proc. KDD* 2011, August 21-24, 2011, San Diego, CA.

[3] G. Tomasi and R. Bro, "A comparison of algorithms for fitting the PARAFAC model," *Comp. Stat. Data Anal.*, vol. 50, no. 7, pp. 1700–1734, 2006.

[4] P. Comon, X. Luciani, and A. L. F. de Almeida, "Tensor decompositions, alternating least squares and other tales," *J. Chemom.*, vol. 23, pp. 393–405, 2009.

[5] P. Tichavsky and Z. Koldovsky, "Simultaneous Search for all Modes in Multilinear Models", *Proc. ICASSP* 2010, Dallas, TX, March 14-19, pp.4114–4117.

[6] D. Nion and N. Sidiropoulos, "Adaptive algorithms to track the PARAFAC decomposition of a third-order tensor," *IEEE Trans. on Signal Proc.*, vol. 57, no. 6, pp. 2299–2310, June 2009.

[7] A. Y. Kibangou and A. L F. de Almeida, "Distributed PARAFAC based DS-CDMA blind receiver for wireless sensor networks," *Proc. SPAWC 2010*, Marrakech, Morocco, Jun. 2010.

[8] A. Y. Kibangou and A. L F. de Almeida, "Distributed Khatri-Rao space-time coding and decoding for cooperative networks," *Proc. EUSIPCO 2011*, Barcelona, Spain, Aug. 2011.

[9] L. Xiao, S. Boyd, and S.-J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel Distrib. Comput.*, vol. 67, pp. 33–46, 2007.

[10] U. Kang, E. Papalexakis, A. Harpale, and C. Faloutsos, "GigaTensor: scaling tensor analysis up by 100 times-Algorithms and discoveries," *Proc. KDD 2012*, Beijing, China, Aug. 2012.

[11] B. Bader and T. Kolda, "Efficient MATLAB computations with sparse and factored tensors," *SIAM Journal on Scientific Computing*, vol. 30, no. 1, pp. 205-231, December, 2007.

[12] E. Acar, T. G. Kolda, D. M. Dunlavy, M. Morup, "Scalable tensor factorizations for incomplete data," *Chemometrics and Intelligent Laboratory Systems*, vol. 106, no. 1, pp. 41-56, Mar. 2011.

[13] N. Sidiropoulos and A. Kyrillidis, "Multi-way compressed sensing for sparse low-rank tensors," *IEEE Signal Proc. Letters*, vol. 19, no. 11, pp. 757–760, November, 2012.

[14] E. Papalexakis, C. Faloutsos, and N. Sidiropoulos, "ParCube: Sparse parallelizable tensor decompositions," *Proc. European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases 2012 (ECML-PKDD)*, Bristol, United Kingdom.

[15] A. de Almeida and A.Y. Kibangou, "Distributed Computation of Tensor Decompositions in Collaborative Networks," *Proc. IEEE CAMSAP 2013*, Saint-Martin, Dec. 2013.

[16] R. Harshman, "Foundation of the PARAFAC procedure: models and conditions for an "explanatory" multimodal factor analysis," *UCLA working papers in phonetics*, vol. 16, pp. 1–84, 1970.

[17] J. Caroll and J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, pp. 283–319, 1970.

[18] J. Kruskal, "Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics," *Linear Algebra Applicat.*, vol. 18, pp. 95–138, 1977.

[19] L. Tang, X. Wang, and H. Liu, "Community detection via heterogeneous interaction analysis," *Data Min. Knowl. Discov.*, vol. 25, no. 1, pp. 1-33, 2012.

[20] X. Dong, P. Frossard, P. Vandergheynst, and N. Nefedov, "Clustering with Multi-Layer Graphs: A Spectral Perspective," *IEEE Transactions on Signal Processing*, vol. 60, no. 11, pp. 5820-5831, Nov 2012.

[21] A.Y. Kibangou, "Graph Laplacian based Matrix Design for Finite-time Distributed Average Consensus," *Proc. of American Control Conference (ACC)*, Montreal, Canada, June 2012.

[22] A.Y. Kibangou, "Step-Size Sequence Design for finite-time Average Consensus in Secure Wireless Sensor Networks," *Systems and Control Letters*, To appear.

[23] X. Liu, N. D. Sidiropoulos, "Cramér-Rao lower bounds for low-rank decomposition of multidimensional arrays," *IEEE Transactions on Signal Processing*, vol. 49, no. 9, pp. 2074–2086, Sept 2011.