A Computationally Efficient Tensor Completion Algorithm

Ioannis C. Tsaknakis, Paris V. Giampouras, Athanasios A. Rontogiannis, and Konstantinos D. Koutroumbas

***Abstract*—We introduce a tensor completion algorithm that uses a group-sparse regularizer with respect to the PARAFAC factors and is based on an optimization scheme that alternatingly minimizes a quadratic upper bound of the associated cost function. The proposed scheme allows matrixwise updates of the PARAFAC factors and, thus, leads to an efficient and scalable iterative algorithm, suitable for big-data applications. Experiments conducted on both synthetic and real data, corroborate the superior performance, in terms of runtime, of the proposed algorithm as compared with the other state-of-the-art approaches.**

***Index Terms*—BSUM framework, group-sparse regularization, PARAFAC decomposition, tensor completion.**

I. Introduction

RECENTLY, high-dimensional data generated by a wealth of machine-learning applications are naturally represented by a multidimensional array, commonly named as a tensor [1]. Such applications include recommendation systems [2], [3], radar-signal processing [4], video processing [5], [6] and topic models [7]. Furthermore, the context in which we are called to perform relevant tasks is often characterized by huge volumes of data suffering from corruptions and missing elements. The processing and extraction of information from highly incomplete and large-scale data sets can be achieved with the development of scalable and computationally efficient tensor completion algorithms that exploit the low-rank inherent in big tensor data.

The first approach for performing tensor completion entails an optimization task that uses rank as a regularizer. However, the computation of the tensor rank is an NP-hard problem [8] and, therefore, this approach does not offer a feasible path. Even so, there are other ways for recovering a low-rank tensor, such as matricizing the tensor and applying matrix-completion algorithms [9], [10] or defining a tensor nuclear norm and formulating the respective regularized optimization task [11]. Recently, there has been a line of work that uses suitable rank-penalization regularizers based on the widespread parallel factor analysis (PARAFAC) decomposition of tensors [12], [13].

Manuscript received April 2, 2018; revised June 12, 2018; accepted June 27, 2018. Date of publication July 2, 2018; date of current version July 13, 2018. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Joao Paulo Papa. *(Corresponding author: Ioannis C. Tsaknakis.)*

The authors are with the Institute for Astronomy, Astrophysics, Space Applications and Remote Sensing, National Observatory of Athens, Penteli 15236, Greece (e-mail: i.tsaknak@gmail.com; parisg@noa.gr; tronto@noa.gr; koutroum@noa.gr).

Digital Object Identifier 10.1109/LSP.2018.2852490

In this letter, we propose a novel tensor completion algorithm based on the PARAFAC decomposition. Specifically, inspired by [14], the formulation proposed in [13] is now suitably modified giving rise to an efficient alternating minimization algorithm. The proposed framework entails a suitably chosen quadratic approximation of the initial cost function and leads to an iterative scheme that enables *matrixwise updates* (in contrast to the computationally expensive column-/row-wise updates) of the PARAFAC factor matrices at each iteration. To the best of our knowledge, this possibility appears for the first time in the relevant literature and results in an efficient and scalable algorithm capable of handling instances that arise in modern big-data applications. Furthermore, the mechanism with which low rankness is imposed allows us to implement a *column pruning (CP) procedure* that dynamically erases factor matrix columns as they become approximately zero. We perform experiments that illustrate the superior performance of the new algorithm in terms of runtime, without sacrificing the accuracy, as compared with the state-of-the-art tensor completion algorithms.

Notation: We denote a vector, a matrix, and a tensor with the symbols $\mathbf{x}$, $\mathbf{X}$, and $\underline{\mathbf{X}}$, respectively. Also, we use the symbols $*$, $\circledcirc$, $\otimes$, $\odot$, and vec$\{\cdot\}$ for the Hadamard product, outer product, Kronecker product, Khatri–Rao product, and row-vectorization operation, respectively. We use the standard notation for vector and matrix norms, i.e., $\|\cdot\|_F$ for the Frobenius norm, and $\|\mathbf{a}\|_p = (\sum_{i=1}^{n} |a_i|^p)^{1/p}$ for the vector l_p norm ($p > 0$), where $\mathbf{a} = [a_1, a_2, \ldots, a_n]^T$. Moreover, we use the matrix $l_{1,2}$ group sparse norm, i.e., $\|\mathbf{A}\|_{1,2} = \sum_{i=1}^{n} \|\mathbf{a}_i\|_2$, where $\mathbf{a}_i, 1 \leq i \leq n$ are the columns of $\mathbf{A} \in \mathbb{R}^{m \times n}$.

Preliminaries: We can decompose a tensor $\underline{\mathbf{X}} \in \mathbb{R}^{I \times J \times K}$ into a sum $\underline{\mathbf{X}} = \sum_{r=1}^{R} \mathbf{a}_r \circledcirc \mathbf{b}_r \circledcirc \mathbf{c}_r$ or equivalently $\underline{\mathbf{X}}(i,j,k) = \sum_{r=1}^{R} \mathbf{a}_r(i)\mathbf{b}_r(j)\mathbf{c}_r(k)$, where $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, and $\mathbf{c}_r \in \mathbb{R}^K$. This is called the PARAFAC decomposition of $\underline{\mathbf{X}}$. The minimum number of terms in the PARAFAC decomposition with which we can express a given tensor is the PARAFAC rank or simply rank of that tensor. Moreover, we can organize the vectors $\{\mathbf{a}_r \in \mathbb{R}^I\}_{r=1}^{R}$, $\{\mathbf{b}_r \in \mathbb{R}^J\}_{r=1}^{R}$, and $\{\mathbf{c}_r \in \mathbb{R}^K\}_{r=1}^{R}$ as columns of the matrices $\mathbf{A} \in \mathbb{R}^{I \times R}$, $\mathbf{B} \in \mathbb{R}^{J \times R}$, and $\mathbf{C} \in \mathbb{R}^{K \times R}$, respectively and denote the PARAFAC decomposition, for convenience, as $\underline{\mathbf{X}} = [[\mathbf{A}, \mathbf{B}, \mathbf{C}]]$.

One notion that will be proven to be very useful is matricization or tensor unfolding. There are three standard ways to unfold a 3-way tensor $\underline{\mathbf{X}}$ depending on the way we position the slabs of the tensor. We can express these matricizations in terms of the PARAFAC factor matrices as $\mathbf{X}_{(1)} = (\mathbf{C} \odot \mathbf{B})\mathbf{A}^T$, $\mathbf{X}_{(2)} = (\mathbf{C} \odot \mathbf{A})\mathbf{B}^T$, and $\mathbf{X}_{(3)} = (\mathbf{B} \odot \mathbf{A})\mathbf{C}^T$.

II. Efficient Tensor Completion Algorithm

We tackle the tensor completion problem by solving an optimization task of the following form:

$$\min_{\substack{\mathbf{A}\in\mathbb{R}^{I\times F},\mathbf{B}\in\mathbb{R}^{J\times F},\\ \mathbf{C}\in\mathbb{R}^{K\times F}}} \frac{1}{2}\|\left(\underline{\mathbf{X}}-[[\mathbf{A},\mathbf{B},\mathbf{C}]]\right)*\underline{\mathbf{\Omega}}\|_F^2+\lambda f(\mathbf{A},\mathbf{B},\mathbf{C}) \tag{1}$$

where $f(\mathbf{A},\mathbf{B},\mathbf{C})$ is an appropriate low-rank inducing regularizer defined with respect to the PARAFAC factor matrices $\mathbf{A}\in\mathbb{R}^{I\times F}$, $\mathbf{B}\in\mathbb{R}^{J\times F}$, and $\mathbf{C}\in\mathbb{R}^{K\times F}$, $\lambda>0$ is a parameter that controls the tradeoff between the fitting term and the low-rank regularization term and $\underline{\mathbf{\Omega}}$ specifies the observed elements of the tensor, i.e.,

$$\underline{\mathbf{\Omega}}(i,j,k)=\begin{cases}1, & \text{if } \underline{\mathbf{X}}(i,j,k) \text{ is observed}\\ 0, & \text{if } \underline{\mathbf{X}}(i,j,k) \text{ is missing}\end{cases}.$$

In order to solve (1), we need to make a safe overestimate F of the true rank and leave the minimization solver to unveil its actual value R.

In [12], the following low-rank-promoting regularizer is used:

$$\begin{aligned}f(\mathbf{A},\mathbf{B},\mathbf{C})&=\frac{1}{2}\left(\|\mathbf{A}\|_F^2+\|\mathbf{B}\|_F^2+\|\mathbf{C}\|_F^2\right)\\&=\frac{1}{2}\left(\sum_{i=1}^{F}\|\mathbf{a}_i\|_2^2+\sum_{i=1}^{F}\|\mathbf{b}_i\|_2^2+\sum_{i=1}^{F}\|\mathbf{c}_i\|_2^2\right)\end{aligned} \tag{2}$$

while in [13], an alternative regularization term is employed based on the group-sparse $l_{1,2}$ norm[1], i.e.,

$$\begin{aligned}f(\mathbf{A},\mathbf{B},\mathbf{C})&=\|\mathbf{A}\|_{1,2}+\|\mathbf{B}\|_{1,2}+\|\mathbf{C}\|_{1,2}\\&=\sum_{i=1}^{F}\|\mathbf{a}_i\|_2+\sum_{i=1}^{F}\|\mathbf{b}_i\|_2+\sum_{i=1}^{F}\|\mathbf{c}_i\|_2.\end{aligned} \tag{3}$$

In this letter, we solve the optimization task in (1) with the regularizer in (3) using a new method. It should be noted that the formulated cost function is nonseparable with respect to $\mathbf{A},\mathbf{B}$, and $\mathbf{C}$, and is also nonsmooth[2]. In order to deal with the former issue, we employ the *block successive upper bound minimization (BSUM)* approach [16], which alternatingly updates the block variables $\mathbf{A}$, $\mathbf{B}$, and $\mathbf{C}$ by solving optimization tasks whose objective functions are upper bounds of the original one. Moreover, the utilization of suitable upper bounds (as the ones proposed in [14] for low-rank matrix recovery problems) will provide us with closed-form expressions for the explicit updates of the factor matrices at each iteration.

In this respect, let us first fix matrices $\mathbf{B}$ and $\mathbf{C}$ and consider the cost function in (1) [with the regularizer (3)] with respect to matrix $\mathbf{A}$, i.e.,

$$\begin{aligned}g(\mathbf{A},\mathbf{B},\mathbf{C})&=\frac{1}{2}\left\|\left\{\mathbf{X}_{(1)}-(\mathbf{C}\odot\mathbf{B})\mathbf{A}^T\right\}*\mathbf{\Omega}_{(1)}\right\|_F^2\\&\quad+\sum_{i=1}^{F}\sqrt{\|\mathbf{a}_i\|_2^2+\epsilon^2}.\end{aligned} \tag{4}$$

[1]Note that the matrix $l_{1,2}$ norm was first used as a low-rank promoting regularizer in [15] in the context of *online* matrix completion, albeit defined in a slightly different way.

[2]This is in contrast to the case where the cost function is formulated using (2).

TABLE I
Complexity per Iteration of Three Tensor Completion Algorithms

Algorithm	CPI
TC-FNR-ALS [12]	$\mathcal{O}\left(\lvert\underline{\mathbf{\Omega}}\rvert F^2+(I+J+K)F^3\right)$
TC-GS-IRLS [13]	$\mathcal{O}\left(\lvert\underline{\mathbf{\Omega}}\rvert F^2+(I+J+K)F^3\right)$
TC-GS-BSUM	$\mathcal{O}\left(\lvert\underline{\mathbf{\Omega}}\rvert F+(I+J+K)F^2+F^3\right)$

$|\underline{\mathbf{\Omega}}|$ denotes the number of observed elements.

In (4), we have used the suitable tensor unfolding $\mathbf{X}_{(1)}$ and we introduced a small positive scalar ϵ^2 in order to mitigate the possible complications of nonsmoothness. Next, we approximate the objective function in (4) using an appropriate quadratic upper-bound function [16]. Assuming that we have at our disposal the estimates of the factor matrices at iteration k, i.e., $\mathbf{A}^{(k)},\mathbf{B}^{(k)}$, and $\mathbf{C}^{(k)}$ at the $(k+1)$th iteration we minimize the following quadratic function:

$$\begin{aligned}&u_{\mathbf{A}}\left(\mathbf{A}|\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)}\right)=g\left(\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)}\right)\\&+\operatorname{tr}\left\{\left(\mathbf{A}-\mathbf{A}^{(k)}\right)^T\nabla_{\mathbf{A}}g\left(\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)}\right)\right\}\\&+\frac{1}{2}\operatorname{vec}\left\{\mathbf{A}-\mathbf{A}^{(k)}\right\}^T\overline{\mathbf{H}}_{\mathbf{A}}\left(\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)}\right)\operatorname{vec}\left\{\mathbf{A}-\mathbf{A}^{(k)}\right\}.\end{aligned} \tag{5}$$

In (5), as a surrogate of the true Hessian, we introduce $\overline{\mathbf{H}}_{\mathbf{A}}(\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)})=\mathbf{I}_I\otimes\widetilde{\mathbf{H}}_{\{\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)}\}}$ with

$$\widetilde{\mathbf{H}}_{\{\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)}\}}=\left(\mathbf{C}^{(k)}\odot\mathbf{B}^{(k)}\right)^T\left(\mathbf{C}^{(k)}\odot\mathbf{B}^{(k)}\right)+\lambda\mathbf{D}_{\mathbf{A}^{(k)}} \tag{6}$$

and $\mathbf{D}_{\mathbf{A}^{(k)}}=\operatorname{diag}\left(\frac{1}{\sqrt{\|\mathbf{a}_1^{(k)}\|_2^2+\epsilon^2}},\ldots,\frac{1}{\sqrt{\|\mathbf{a}_F^{(k)}\|_2^2+\epsilon^2}}\right)$.

Essentially, as shown in the Appendix, (5) is an upper bound for the second-order Taylor approximation of the cost function $g(\mathbf{A},\mathbf{B}^{(k)},\mathbf{C}^{(k)})$ around $\mathbf{A}^{(k)}$. The minimization of the function in (5) leads to a closed-form expression for updating $\mathbf{A}$ at the iteration $k+1$. Also, in (6) we may use the identity $(\mathbf{C}\odot\mathbf{B})^T(\mathbf{C}\odot\mathbf{B})=\left(\mathbf{C}^T\mathbf{C}\right)*\left(\mathbf{B}^T\mathbf{B}\right)$ in order to further reduce the complexity of the algorithm. We can follow the same procedure for the other two factors and obtain analytical expressions for updating matrices $\mathbf{B}$ and $\mathbf{C}$, as well. These expressions are the main steps of the proposed tensor completion group sparse BSUM (TC-GS-BSUM) algorithm given in Algorithm 1.

Note that by adopting the upper-bound surrogate cost functions as above, the resulting tensor completion scheme incorporates solely the matrixwise updates that can be accomplished with reduced computational complexity. This is also verified with the results presented in Table I, where the complexity per iteration (CPI) of the proposed algorithm is compared with that of two state-of-the-art relevant schemes that rely on row-wise factor updates. Moreover, we may establish the convergence of the iterative procedure to a stationary point of the objective function with the following proposition.

Proposition 1: Every limit point of the sequence $\{(\mathbf{A}^{(k)},\mathbf{B}^{(k)},\mathbf{C}^{(k)})\}$ generated by Algorithm 1 is a stationary point of the cost function in (1) with regularization in (3).

Algorithm 1: Tensor completion-group sparse regularization-BSUM.

Input: $\underline{\mathbf{X}}, \underline{\mathbf{\Omega}}, \lambda$, toler

Initialize: $\mathbf{A}^{(0)}, \mathbf{B}^{(0)}, \mathbf{C}^{(0)}$

repeat

$$\mathbf{A}^{(k+1)} \leftarrow \mathbf{A}^{(k)} + \left\{ \left[\left(\mathbf{X}_{(1)}^T - \mathbf{A}^{(k)} \left(\mathbf{C}^{(k)} \odot \mathbf{B}^{(k)} \right)^T \right) * \mathbf{\Omega}_{(1)}^T \right] \left(\mathbf{C}^{(k)} \odot \mathbf{B}^{(k)} \right) - \lambda \mathbf{A}^{(k)} \mathbf{D}_{\mathbf{A}^{(k)}} \right\} \left[\left(\mathbf{C}^{(k)T} \mathbf{C}^{(k)} \right) * \left(\mathbf{B}^{(k)T} \mathbf{B}^{(k)} \right) + \lambda \mathbf{D}_{\mathbf{A}^{(k)}} \right]^{-1}$$

$$\mathbf{B}^{(k+1)} \leftarrow \mathbf{B}^{(k)} + \left\{ \left[\left(\mathbf{X}_{(2)}^T - \mathbf{B}^{(k)} \left(\mathbf{C}^{(k)} \odot \mathbf{A}^{(k+1)} \right)^T \right) * \mathbf{\Omega}_{(2)}^T \right] \left(\mathbf{C}^{(k)} \odot \mathbf{A}^{(k+1)} \right) - \lambda \mathbf{B}^{(k)} \mathbf{D}_{\mathbf{B}^{(k)}} \right\} \left[\left(\mathbf{C}^{(k)T} \mathbf{C}^{(k)} \right) * \left(\mathbf{A}^{(k+1)T} \mathbf{A}^{(k+1)} \right) + \lambda \mathbf{D}_{\mathbf{B}^{(k)}} \right]^{-1}$$

$$\mathbf{C}^{(k+1)} \leftarrow \mathbf{C}^{(k)} + \left\{ \left[\left(\mathbf{X}_{(3)}^T - \mathbf{C}^{(k)} \left(\mathbf{B}^{(k+1)} \odot \mathbf{A}^{(k+1)} \right)^T \right) * \mathbf{\Omega}_{(3)}^T \right] \left(\mathbf{B}^{(k+1)} \odot \mathbf{A}^{(k+1)} \right) - \lambda \mathbf{C}^{(k)} \mathbf{D}_{\mathbf{C}^{(k)}} \right\} \left[\left(\mathbf{B}^{(k+1)T} \mathbf{B}^{(k+1)} \right) * \left(\mathbf{A}^{(k+1)T} \mathbf{A}^{(k+1)} \right) + \lambda \mathbf{D}_{\mathbf{C}^{(k)}} \right]^{-1}$$

$$\widehat{\underline{\mathbf{X}}}^{(k+1)} \leftarrow \left[\left[\mathbf{A}^{(k+1)}, \mathbf{B}^{(k+1)}, \mathbf{C}^{(k+1)} \right] \right]$$

until $\|\widehat{\underline{\mathbf{X}}}^{(k+1)} - \widehat{\underline{\mathbf{X}}}^{(k)}\|_F / \|\widehat{\underline{\mathbf{X}}}^{(k)}\|_F < \text{toler}$

Output: $\widehat{\underline{\mathbf{X}}} = [[\mathbf{A}, \mathbf{B}, \mathbf{C}]]$

Proof: Appendix A. ■

III. Experimental Evaluation

In order to evaluate the performance of the new algorithm, we test it for both synthetic and real data and compare it with the following recently reported algorithms.

1) *TC-FNR-ALS:* Tensor Completion with the Frobenius norm regularizer in (2) using an alternating least squares algorithm. [12].
2) *TC-GS-IRLS:* Tensor Completion with the group-sparse regularizer (3) using an iteratively re-weighted least squares algorithm. [13].

The experiments were implemented on MATLAB and were executed on a computer using an i5 2.5 GHz dual-core CPU with 4-GB RAM.

As it is claimed in [12], the mechanism with which low rankness is promoted with regularizer in (2) is through erasing rank-1 terms $\mathbf{a}_r \circledcirc \mathbf{b}_r \circledcirc \mathbf{c}_r$ or equivalently zeroing the columns of the factor matrices $\mathbf{A}, \mathbf{B}$, and $\mathbf{C}$. In [13], it is implied that the same mechanism also applies for the regularizer in (3). This property may enable us to apply a CP procedure that removes at each iteration the columns of the factor matrices, whose energy falls under a predefined threshold.

TABLE II
Results of the Experiments on Synthetic Data

Exp.	Algorithm	Without CP		With CP	
		# Iter.	Time(s)	# Iter.	Time(s)
1	*FNR-ALS*	68	99	68	20
	GS-IRLS	39	57	24	10
	GS-BSUM	134	16	71	6
2	*FNR-ALS*	62	160	67	58
	GS-IRLS	42	110	31	29
	GS-BSUM	100	21	84	13
3	*FNR-ALS*	64	297	58	69
	GS-IRLS	46	212	38	57
	GS-BSUM	74	27	84	21

A. Synthetic Data

On synthetic data we evaluate the performance using the following three criteria:

1) normalized reconstruction error (NRE) $= \frac{\|\underline{\mathbf{X}} - \widehat{\underline{\mathbf{X}}}\|_F}{\|\underline{\mathbf{X}}\|_F}$, where $\underline{\mathbf{X}}$ is the original complete tensor and $\widehat{\underline{\mathbf{X}}}$ is the algorithm's output;
2) rank of the recovered tensor;
3) runtime.

In this letter, we are mainly interested in runtime.

We consider the following three experiments : 1) $80 \times 80 \times 80$, rank $= 15$; 2) $100 \times 100 \times 100$, rank $= 20$; and 3) $120 \times 120 \times 120$, rank $= 20$. In all experiments we uniformly remove at random 80% of the tensors' entries and we add independent identically distributed Gaussian noise to achieve SNR $= 18$ dB. Also, we choose the algorithm parameters such that given that the correct rank is approximately achieved, the error is the minimum one (actually we allowed a small deterioration in the error in return of a significant improvement in the runtime). Finally, the results are averaged over 50 independent runs and presented in Table II.

First of all, we should note that all the algorithms approximately recover the correct rank and achieve errors that lie in the range $[2.5, 2.7] \cdot 10^{-2}$, $[2.2, 2.5] \cdot 10^{-2}$ and $[1.9, 2.2] \cdot 10^{-2}$ in the first, second, and third experiments, respectively. There are no significant differences in the NREs of the three algorithms in the same experiment as well as when we consider the same algorithm with and without CP.

However, the runtime of TC-GS-BSUM is lower than the runtime of the other two algorithms in all the scenarios examined. Especially, when the experiments are performed without CP, the difference is greater. Furthermore, there is a considerable reduction in the runtime of the other two algorithms when we apply the CP mechanism. That highlights the power of the latter to reduce the runtime of the existing tensor completion algorithms that work by annihilating columns of the PARAFAC factor matrices. In conclusion, the results on synthetic data show that TC-GS-BSUM offers reduced runtime compared with the other schemes without suffering from losses in accuracy.

B. Video Completion

Here, we compare the TC-GS-IRLS and TC-GS-BSUM algorithms with CP in recovering the missing pixels of an incomplete gray-scale video (in literature video-completion experiments have been performed in [5] and [6]). For our experiments, we

TABLE III
RESULTS IN VIDEO COMPLETION

Missing(%)	Algorithm	NRE($\cdot 10^{-2}$)	# Iter.	Time(s)
70	*TC-GS-IRLS*	5.32	141	874
	TC-GS-BSUM	5.23	205	289
90	*TC-GS-IRLS*	5.25	152	794
	TC-GS-BSUM	5.43	297	414

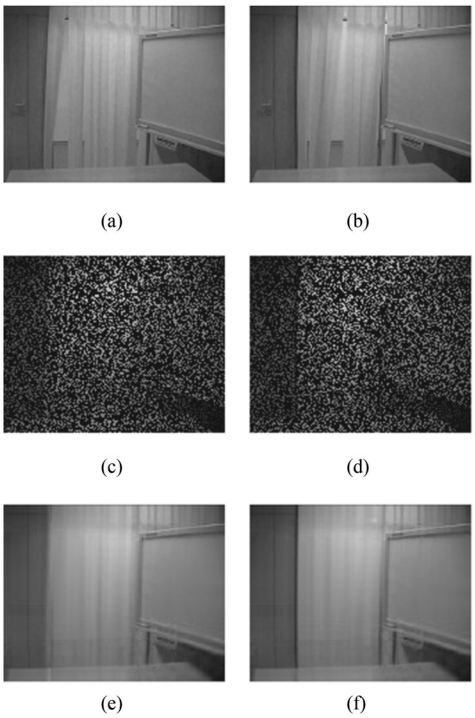

Fig. 1. Original (a) and (b); incomplete (c) and (d) and reconstructed by TC-GS-BSUM (e) and (f) images for two different frames and 70% missing data.

use the "Curtain" video [17]. Specifically, we consider a tensor, where every frontal slab corresponds to a frame of the video and randomly erase a specific percentage of the tensor entries. Moreover, we start the algorithms with initial rank $F = 100$ and choose λ so that the algorithms reach a rank approximately equal to 20. In Table III, we present the results and in Fig. 1 we show the output of TC-GS-BSUM for two specific frames.

Notice that both algorithms recover the corrupted video with almost the same accuracy, but the time needed by TC-GS-BSUM is considerably less than the respective time of TC-GS-IRLS, in both experiments.

IV. CONCLUSION

In this letter, we presented an algorithm for tensor completion that offers enhanced computational efficiency compared with other algorithms that belong to the same category. This is achieved mainly with the utilization of the appropriate optimization approach and is further facilitated with the usage of a CP mechanism. It should be noted that we can employ the same approach to other regularizers and problems, such as tensor decomposition, thus, expecting comparable improvements in runtime.

APPENDIX A
PROOF OF THE PROPOSITION 1

Proof: For a quadratic upper-bound function that satisfies the assumptions of the BSUM theory ([16, Table 3]), Theorem 1 in [16] states that every limit point of the sequence generated by the respective minimization solver is a stationary point of the original objective function. Therefore, it suffices to show that the quadratic function u_A defined in (5) is indeed an upper bound. For this, we need to prove that $\overline{\mathbf{H}}_{\mathbf{A}}$ and $\overline{\mathbf{H}}_{\mathbf{A}} - \mathbf{H}_{\mathbf{A}}$ are positive semidefinite matrices, where $\mathbf{H}_{\mathbf{A}}$ is the true Hessian with respect to $\mathbf{A}$ of the original cost function. The proof follows the same reasoning used in [14]. It is easy to see that $\overline{\mathbf{H}}_{\mathbf{A}} \succeq 0$. Moreover, it can be easily shown that the difference between the exact Hessian and the surrogate is

$$\begin{aligned}\overline{\mathbf{H}}_{\mathbf{A}}(\mathbf{A},\mathbf{B},\mathbf{C}) - \mathbf{H}_{\mathbf{A}}(\mathbf{A},\mathbf{B},\mathbf{C}) &= \mathbf{I}_I \otimes \left(\mathbf{U}^T\mathbf{U} + \lambda\mathbf{D}_{\mathbf{A}}\right) \\ &- \left\{\text{diag}\left(\mathbf{U}^T\mathbf{P}_1\mathbf{U},\ldots,\mathbf{U}^T\mathbf{P}_I\mathbf{U}\right) + \mathbf{M}\right\} \\ &= \text{diag}\left(\mathbf{U}^T\left(\mathbf{I}-\mathbf{P}_1\right)\mathbf{U},\ldots,\mathbf{U}^T\left(\mathbf{I}-\mathbf{P}_I\right)\mathbf{U}\right) \qquad (7)\end{aligned}$$

$$+\begin{bmatrix} \lambda\mathbf{D}_{\mathbf{A}} - \mathbf{M}_{11} & -\mathbf{M}_{12} & \ldots & -\mathbf{M}_{1I} \\ -\mathbf{M}_{21} & \lambda\mathbf{D}_{\mathbf{A}} - \mathbf{M}_{22} & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ -\mathbf{M}_{I1} & \ldots & -\mathbf{M}_{I(I-1)} & \lambda\mathbf{D}_{\mathbf{A}} - \mathbf{M}_{II} \end{bmatrix} \qquad (8)$$

where $\mathbf{U} = (\mathbf{C}\odot\mathbf{B})^T(\mathbf{C}\odot\mathbf{B})$, $\mathbf{M} = [\mathbf{M}_{ij}]$, $\mathbf{P}_i$ is a diagonal matrix with

$$\mathbf{P}_i(j,j) = \begin{cases} 1, & \text{if } \mathbf{\Omega}_{(1)}(j,i) = 1 \\ 0, & \text{if } \mathbf{\Omega}_{(1)}(j,i) = 0 \end{cases}$$

$$\mathbf{M}_{ii} = \lambda\cdot\text{diag}\left(\frac{\|\mathbf{a}_1\|_2^2 - (a_{i1})^2 + \epsilon^2}{\left(\|\mathbf{a}_1\|_2^2+\epsilon^2\right)^{3/2}},\ldots,\frac{\|\mathbf{a}_F\|_2^2 - (a_{iF})^2 + \epsilon^2}{\left(\|\mathbf{a}_F\|_2^2+\epsilon^2\right)^{3/2}}\right)$$

and for $i \neq j$

$$\mathbf{M}_{ij} = \lambda\cdot\text{diag}\left(\frac{-a_{i1}a_{j1}}{\left(\|\mathbf{a}_1\|_2^2+\epsilon^2\right)^{3/2}},\ldots,\frac{-a_{iF}a_{jF}}{\left(\|\mathbf{a}_F\|_2^2+\epsilon^2\right)^{3/2}}\right).$$

To ease the notation, we denote the matrices in (7) and (8) of the above sum with $\mathbf{V}$ and $\mathbf{N}$, respectively. We can easily see that $\mathbf{V} \succeq 0$. Also, the $F\times F$ ijth block submatrix of $\mathbf{N}$ can be written as

$$\begin{aligned}\mathbf{N}_{ij} &= \lambda\cdot\text{diag}\left(\frac{a_{i1}}{\left(\|\mathbf{a}_1\|_2^2+\epsilon^2\right)^{3/4}},\ldots,\frac{a_{iF}}{\left(\|\mathbf{a}_F\|_2^2+\epsilon^2\right)^{3/4}}\right) \\ &\quad\cdot\text{diag}\left(\frac{a_{j1}}{\left(\|\mathbf{a}_1\|_2^2+\epsilon^2\right)^{3/4}},\ldots,\frac{a_{jF}}{\left(\|\mathbf{a}_F\|_2^2+\epsilon^2\right)^{3/4}}\right) \\ &= \lambda\cdot\mathbf{L}_i\cdot\mathbf{L}_j^T,\end{aligned}$$

where $\mathbf{L}_i = \text{diag}(\frac{a_{i1}}{\left(\|\mathbf{a}_1\|_2^2+\epsilon^2\right)^{3/4}},\ldots,\frac{a_{iF}}{\left(\|\mathbf{a}_F\|_2^2+\epsilon^2\right)^{3/4}})\cdot$

Therefore, if we define the matrix $\mathbf{L} = [\mathbf{L}_1^T,\mathbf{L}_2^T,\ldots,\mathbf{L}_I^T]^T$ it holds that $\mathbf{N} = \lambda\cdot\mathbf{L}\mathbf{L}^T$ and as a result $\mathbf{N}\succeq 0$. ■

REFERENCES

[1] 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 Trans. Signal Process.*, vol. 65, no. 13, pp. 3551–3582, Jul. 2017.

[2] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver, "Multiverse recommendation: N-dimensional tensor factorization for context-aware collaborative filtering," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010.

[3] P. Symeonidis and A. Zioupos, *Matrix and Tensor Factorization Techniques for Recommender Systems*. New York, NY, USA: Springer-Verlag, 2016.

[4] D. Nion and N. D. Sidiropoulos, "Tensor algebra and multidimensional harmonic retrieval in signal processing for MIMO radar," *IEEE Trans. Signal Process.*, vol. 58, no. 11, pp. 5693–5705, Nov. 2010.

[5] Z. Zhang and S. Aeron, "Exact tensor completion using t-SVD," *IEEE Trans. Signal Process.*, vol. 65, no. 6, pp. 1511–1526, Mar. 2017.

[6] J. A. Bengua, H. N. Phien, H. D. Tuan, and M. N. Do, "Efficient tensor completion for color image and video recovery: Low-rank tensor train," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2466–2479, May 2017.

[7] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, and M. Telgarsky, "Tensor decompositions for learning latent variable models," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 2773–2832, 2014.

[8] C. J. Hillar and L.-H. Lim, "Most tensor problems are NP-hard," *J. ACM*, vol. 60, no. 6, 2013, Art. no. 45.

[9] S. Gandy, B. Recht, and I. Yamada, "Tensor completion and low-n-rank tensor recovery via convex optimization," *Inverse Probl.*, vol. 27, no. 2, 2011, Art. no. 025010.

[10] C. Mu, B. Huang, J. Wright, and D. Goldfarb, "Square deal: Lower bounds and improved relaxations for tensor recovery," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 73–81.

[11] M. Yuan and C.-H. Zhang, "On tensor completion via nuclear norm minimization," *Found. Comput. Math.*, vol. 16, no. 4, pp. 1031–1068, 2016.

[12] J. A. Bazerque, G. Mateos, and G. B. Giannakis, "Rank regularization and Bayesian inference for tensor completion and extrapolation," *IEEE Trans. Signal Process.*, vol. 61, no. 22, pp. 5689–5703, Nov. 2013.

[13] B. Yang, G. Wang, and N. D. Sidiropoulos, "Tensor completion via group-sparse regularization," in *Proc. IEEE 50th Asilomar Conf. Signals, Syst. Comput.*, 2016, pp. 1750–1754.

[14] P. V. Giampouras, A. A. Rontogiannis, and K. D. Koutroumbas, "Alternating iteratively reweighted minimization algorithms for low-rank matrix factorization," *arXiv:1710.02004*.

[15] P. V. Giampouras, A. A. Rontogiannis, and K. D. Koutroumbas, "Online low-rank subspace learning from incomplete data using rank revealing l2/l1 regularization," in *Proc. IEEE Statist. Signal Process. Workshop,*, 2016, pp. 1–5.

[16] M. Hong, M. Razaviyayn, Z.-Q. Luo, and J.-S. Pang, "A unified algorithmic framework for block-structured optimization involving big data: with applications in machine learning and signal processing," *IEEE Signal Process. Mag.*, vol. 33, no. 1, pp. 57–77, Jan. 2016.

[17] "Curtain video." [Online]. Available: http://pages.cs.wisc.edu/jiaxu/projects/gosus/supplement/