NOVEL ALGORITHMS FOR EXACT AND EFFICIENT L1-NORM-BASED TUCKER2 DECOMPOSITION

Dimitris G. Chachlakis and Panos P. Markopoulos[*]

Dept. of Electrical and Microelectronic Engineering, Rochester Institute of Technology
Rochester, NY, 14623, USA
E-mail: dimitris@mail.rit.edu,panos@rit.edu

ABSTRACT

We consider corruption-resistant L1-norm-based TUCKER2 (L1-TUCKER2) decomposition of a $D \times M \times N$ 3-way tensor, treated (with no loss of generality) as a collection of N $D \times M$ matrices. Our contributions are as follows. First, we show that rank-1 L1-TUCKER2 can be cast as a combinatorial problem over N antipodal-binary variables; accordingly, we provide the first exact algorithm for its solution. Then, we develop an efficient (quadratic-cost/near-exact) algorithm that approximates the solution to rank-1 L1-TUCKER2 by means of a converging sequence of optimal single-bit flips; the algorithm is accompanied by formal convergence proof and complexity analysis. Finally, by means of the standard deflation technique, we generalize the developed bit-flipping algorithm for solving L1-TUCKER2 decomposition problems of general rank. Our extensive numerical studies show that the bit-flipping algorithm returns the exact L1-TUCKER2 solution with very high frequency. Moreover, the developed exact and efficient algorithms exhibit remarkable outlier resistance, outperforming some of the most popular L2-norm-based and L1-norm-based counterparts.

***Index Terms*—** Data analysis, L1-norm, outliers, robust, tensors, Tucker decomposition.

1. INTRODUCTION

TUCKER decomposition [1, 2] is a fundamental method for n-way tensor analysis, with applications in a wide range of fields, including computer vision [3, 4], wireless communications [5], biomedical signal processing [6], deep neural networks [7], and social-network data analysis [8, 9] to name a few. If the n-way tensor under processing has been formed by the concatenation (say, across the n-th mode) of a number of coherent (same class, or distribution) $(n-1)$-way coherent tensor measurements, then TUCKER2 decomposition is commonly employed. TUCKER2 seeks to *jointly* decompose the collected $(n-1)$-way tensors and unveil the low-rank multi-linear structure of their class/distribution. Popular solvers for TUCKER/TUCKER2 are the Higher-Order Singular-Valued Decomposition (HOSVD) and the Higher-Order Orthogonal Iteration (HOOI) algorithms [2, 4]. A detailed presentation of TUCKER, TUCKER2, and their algorithmic solvers is offered in [2, 10, 11]. For $n = 2$, TUCKER/TUCKER2 coincide with standard matrix Principal-Component Analysis (PCA). For $n = 3$, TUCKER2 has been also studied/solved under the name Generalized Low-Rank Approximation of Matrices (GLRAM) [12] and Multilinear PCA [13]. In this work, we focus on TUCKER2 and, more specifically, on its L1-norm reformulation discussed below.

Similar to PCA, TUCKER and TUCKER2 have been shown to be sensitive against outliers within the processed tensor [14–16], due to their Frobenius/L2-norm formulation (i.e., L2-residual-error minimization, or, equivalently, L2-projection-variance maximization [2]). On the other hand, L1-norm-based PCA (L1-PCA) [17–20], formulated simply by substitution of the L2-norm in PCA by the L1-norm, has exhibited in the past few years remarkable outlier resistance in many applications. Extending this formulation to tensor processing, one can similarly endow robustness/outlier-resistance to TUCKER2 by substituting the L2-norm in its formulation by the L1-norm. We call this new tensor decomposition method *L1-TUCKER2*.

For $n = 3$, an approximate algorithm for L1-TUCKER2 was proposed in [14] under the title L1-norm Tensor PCA (TPCA-L1). This algorithm is iterative and guarantees convergence, but not exact solution to the L1-TUCKER2 problem. In fact, the exact solution to L1-TUCKER2 remains to date unknown. In this work, we present for the first time the exact solution to L1-TUCKER2, for the special case of rank-1 decomposition. Specifically, in Section 3, we show that rank-1 L1-TUCKER2 can be solved by an exhaustive search over antipodal-binary (bit) variables. Next, in Section 4, we develop an efficient (quadratic-cost/near-exact) algorithm that approximates the solution to rank-1 L1-TUCKER2 by means of converging single-bit-flipping iterations. Finally, we employ rank-deflation to generalize the developed bit-flipping algorithm for solving L1-TUCKER2 decomposition of general rank. Our numerical studies in Section 5 show that the bit-flipping algorithm returns the exact solution to L1-TUCKER2 with very high probability. Moreover, our numerical studies on low-rank tensor approximation in the presence of outliers show that both the exact and the efficient algorithms presented in this work exhibit remarkable outlier resistance, outperforming some of the most popular L2-norm-based and L1-norm-based TUCKER, such as HOSVD, HOOI, GLRAM, and TPCA-L1.

2. PROBLEM STATEMENT

We consider a collection of N real-valued matrices of equal size, $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_N \in \mathbb{R}^{D \times M}$, collated to form the 3-way tensor $\underline{\mathbf{X}} \in \mathbb{R}^{D \times M \times N}$, such that $\underline{\mathbf{X}}_{:,:,i} = \mathbf{X}_i$. For any rank $d \leq \min\{D, M\}$, TUCKER2 decomposition of $\underline{\mathbf{X}}$ seeks to jointly analyze its matrix "slabs" $\{\mathbf{X}_i\}_{i=1}^N$, by maximizing $\sum_{i=1}^N \|\mathbf{U}^\top \mathbf{X}_i \mathbf{V}\|_F^2$ over $\mathbf{U} \in \mathbb{R}^{D \times d}$ and $\mathbf{V} \in \mathbb{R}^{M \times d}$, such that $\mathbf{U}^\top \mathbf{U} = \mathbf{V}^\top \mathbf{V} = \mathbf{I}_d$. Then, for every i, $\mathbf{X}_i$ is low-rank approximated as $\mathbf{U}\mathbf{U}^\top \mathbf{X}_i \mathbf{V}\mathbf{V}^\top$. The squared L2/Frobenius norm $\| \cdot \|_F^2$ in the metric of TUCKER2 returns the summation of the squared entries of its matrix argument. Clearly, for $N = 1$, TUCKER2 simplifies to the rank-d approximation of the single matrix $\mathbf{X}_1 \in \mathbb{R}^{D \times M}$, solved by means of standard Singular Value Decomposition (SVD) [21]; i.e., the optimal arguments $\mathbf{U}$ and $\mathbf{V}$ are built by the d left-hand and right-hand singular

[*]Corresponding author.

vectors of $\mathbf{X}_1$, respectively.

To counteract against the impact of any outliers that possibly exist among the entries of $\{\mathbf{X}_i\}_{i=1}^N$, in this work we consider the L1-norm-based TUCKER2 reformulation

$$\text{L1-TUCKER2:} \quad \underset{\substack{\mathbf{U}\in\mathbb{R}^{D\times d};\ \mathbf{U}^\top\mathbf{U}=\mathbf{I}_d \\ \mathbf{V}\in\mathbb{R}^{M\times d};\ \mathbf{V}^\top\mathbf{V}=\mathbf{I}_d}}{\text{maximize}} \sum_{i=1}^N \|\mathbf{U}^\top\mathbf{X}_i\mathbf{V}\|_1, \quad (1)$$

where the L1-norm $\|\cdot\|_1$ returns the summation of the absolute values of its matrix argument. The formulation in (1) was studied in [14] under the title TPCA-L1, where authors presented an approximate iterative algorithm for its solution (with no guarantee for global optimality). The exact solution to (1) remains to date unknown. In this work, we deliver for the first time the exact solution to L1-TUCKER2 for the special case of $d = 1$. In addition, we offer efficient (quadratic-cost/near-exact) algorithms for L1-TUCKER2, for decomposition of rank $d \geq 1$.

3. EXACT SOLUTION TO L1-TUCKER2 FOR $d = 1$

We first show that, for $d = 1$, L1-TUCKER2 in (1) can be cast as (and solved through) a combinatorial problem over antipodal-binary variables. We start with the observation that, for $d = 1$, L1-TUCKER2 takes the simpler form

$$\underset{\substack{\mathbf{u}\in\mathbb{R}^{D\times 1},\mathbf{v}\in\mathbb{R}^{M\times 1} \\ \|\mathbf{u}\|_2=\|\mathbf{v}\|_2=1}}{\text{maximize}} \sum_{i=1}^N |\mathbf{u}^\top\mathbf{X}_i\mathbf{v}|. \quad (2)$$

Then, we notice that, for any real vector $\mathbf{a} \in \mathbb{R}^m$, it holds $\|\mathbf{a}\|_1 = \sum_{i=1}^m |a_i| = \sum_{i=1}^m \text{sgn}(a_i)\, a_i = \text{sgn}(\mathbf{a})^\top \mathbf{a} = \max_{\mathbf{b}\in\{\pm1\}^m} \mathbf{b}^\top\mathbf{a}$, where $\text{sgn}(\cdot)$ returns the $\{\pm1\}$-sign of its argument. Therefore, for any given $\mathbf{u} \in \mathbb{R}^D$ and $\mathbf{v} \in \mathbb{R}^M$, it holds that

$$\sum_{i=1}^N |\mathbf{u}^\top\mathbf{X}_i\mathbf{v}| = \max_{\mathbf{b}\in\{\pm1\}^N} \mathbf{u}^\top\mathbf{X}(\mathbf{b})\mathbf{v}, \quad (3)$$

where, for any $\mathbf{b} \in \{\pm1\}^N$, we define $\mathbf{X}(\mathbf{b}) \doteq \sum_{i=1}^N b_n\mathbf{X}_n$, for ease in notation. Accordingly, the maximum in (3) is attained for $\mathbf{b} = [\text{sgn}\left(\mathbf{u}^\top\mathbf{X}_1\mathbf{v}\right), \text{sgn}\left(\mathbf{u}^\top\mathbf{X}_2\mathbf{v}\right), \ldots, \text{sgn}\left(\mathbf{u}^\top\mathbf{X}_N\mathbf{v}\right)]^\top$. Next, we observe that, for any $\mathbf{b} \in \{\pm1\}^N$ and corresponding $D \times M$ matrix $\mathbf{X}(\mathbf{b})$, it holds that

$$\max_{\mathbf{u},\mathbf{v};\|\mathbf{u}\|_2=\|\mathbf{v}\|_2=1} \mathbf{u}^\top\mathbf{X}(\mathbf{b})\mathbf{v} = \sigma_{\max}\left(\mathbf{X}(\mathbf{b})\right) \quad (4)$$

where $\sigma_{\max}(\cdot)$ returns the highest singular value of its matrix argument [21]. The maximum in (4) is attained if $\mathbf{u}$ and $\mathbf{v}$ are the left-hand and right-hand dominant singular vectors of $\mathbf{X}(\mathbf{b})$, respectively. Combining (3) and (4), we obtain

$$\max_{\mathbf{u},\mathbf{v};\|\mathbf{u}\|_2=\|\mathbf{v}\|_2=1} \sum_{i=1}^N |\mathbf{u}^\top\mathbf{X}_i\mathbf{v}| = \max_{\substack{\mathbf{b}\in\{\pm1\}^N \\ \mathbf{u},\mathbf{v};\|\mathbf{u}\|_2=\|\mathbf{v}\|_2=1}} \mathbf{u}^\top\left(\mathbf{X}(\mathbf{b})\right)\mathbf{v} \quad (5)$$

$$= \max_{\mathbf{b}\in\{\pm1\}^N} \sigma_{\max}\left(\mathbf{X}(\mathbf{b})\right). \quad (6)$$

The following Proposition 1 derives straightforwardly from (5)-(6) and unveils the combinatorial nature of L1-TUCKER2 in (2).

Proposition 1. *Let $\mathbf{b}_{\text{opt}}$ be a solution to*

$$\underset{\mathbf{b}\in\{\pm1\}^N}{\text{maximize}}\ \sigma_{\max}(\mathbf{X}(\mathbf{b})). \quad (7)$$

Also, let $\mathbf{u}_{\text{opt}} \in \mathbb{R}^D$ and $\mathbf{v}_{\text{opt}} \in \mathbb{R}^M$ be the left-hand and right-hand dominant singular vectors of $\mathbf{X}(\mathbf{b}_{\text{opt}}) \in \mathbb{R}^{D\times M}$, respectively. Then, $(\mathbf{u}_{\text{opt}}, \mathbf{v}_{\text{opt}})$ is an optimal solution to (2). Also, $\mathbf{b}_{\text{opt}} = [\text{sgn}\left(\mathbf{u}_{\text{opt}}^\top\mathbf{X}_1\mathbf{v}_{\text{opt}}\right), \ldots, \text{sgn}\left(\mathbf{u}_{\text{opt}}^\top\mathbf{X}_N\mathbf{v}_{\text{opt}}\right)]^\top$ and $\sum_{i=1}^N |\mathbf{u}_{\text{opt}}^\top\mathbf{X}_i\mathbf{v}_{\text{opt}}| = \mathbf{u}_{\text{opt}}^\top\mathbf{X}(\mathbf{b}_{\text{opt}})\mathbf{v}_{\text{opt}} = \sigma_{\max}\left(\mathbf{X}(\mathbf{b}_{\text{opt}})\right)$. In the special case that $\mathbf{u}_{\text{opt}}^\top\mathbf{X}_i\mathbf{v}_{\text{opt}} = 0$, for some $i \in \{1, 2, \ldots, N\}$, $[\mathbf{b}_{\text{opt}}]_i$ can be set to $+1$, having no effect to the metric of (7). □

Algorithm 1: Exact L1-TUCKER2 for $d = 1$

Input: $\underline{\mathbf{X}}$
0: Set $m' \leftarrow 0$
1: For every $\mathbf{b} \in \{\pm1\}^N$, do
2: Calculate $m \leftarrow \sigma_{\max}(\mathbf{X}(\mathbf{b}))$
3: If $m > m'$, $m' \leftarrow m$ and $\mathbf{b}_{\text{opt}} \leftarrow \mathbf{b}$
4: $(\mathbf{u}_{\text{opt}}, \mathbf{v}_{\text{opt}}) \leftarrow \texttt{dsv}(\mathbf{X}(\mathbf{b}_{\text{opt}}))$
Output: $\mathbf{u}_{\text{opt}}, \mathbf{v}_{\text{opt}}$

Fig. 1. Exact algorithms for L1-TUCKER2 decomposition of $\underline{\mathbf{X}}$, for $d = 1$. $\texttt{dsv}(\cdot)$ returns the dominant singular vectors of its matrix argument.

Proposition 1 establishes that, given the solution to (7), $\mathbf{b}_{\text{opt}}$, the solution to L1-TUCKER2 ($d = 1$) is obtained simply by the SVD of the $D \times M$ matrix $\mathbf{X}(\mathbf{b}_{\text{opt}})$. The solution to (7) can be obtained, e.g., by exhaustive search in its size-2^N feasibility set $\{\pm1\}^N$. This exhaustive search, summarized in Fig. 1, constitutes the first exact algorithm in the literature for L1-TUCKER2 rank-1 decomposition.

Certainly, exhaustive search over $\{\pm1\}^N$ becomes quickly intractable as N increases. For instance, for $N = 100$, the exact algorithm of Fig. 1 would require 2^{100} singular-value calculations, which is of course impractical. In the sequel we present an efficient solver for L1-TUCKER2 with cost at most quadratic in (D, N, M).

4. L1-TUCKER2 VIA BIT-FLIPPING

Similar to L1-TUCKER2, L1-PCA has also been shown to be solvable through combinatorial optimization over a binary feasibility set [17, 18]. Recent works on L1-PCA [22, 23] propose to evade a costly exhaustive search and, instead, approach the solution by means of *bit-flipping* iterations. Motivated by this idea, here we develop *(BF-TUCKER2)*: a quadratic-cost algorithm that approximates the solution to L1-TUCKER2, by means of converging bit-flipping iterations. In the sequel, we present BF-TUCKER2 for $d = 1$. Then, we generalize it to $d > 1$.

4.1. BF-TUCKER2 for $d = 1$

We initialize at an arbitrary binary vector $\mathbf{b}^{(1)} \in \{\pm1\}^N$, e.g., as $\mathbf{b}^{(1)} = \text{sgn}(\mathbf{y})$, for some Gaussian $\mathbf{y} \sim \mathcal{N}(\mathbf{0}_N, \mathbf{I}_N)$. Then, we conduct single-bit-flipping iterations as follows. At the q-th iteration step, $q \geq 1$, we find the entry of vector $\mathbf{b}^{(q)}$ the negation/flipping of which offers the maximum increase in the metric of (7). That is, we find the index k that solves

$$k = \underset{i\in\{1,2,\ldots,N\}}{\text{argmax}}\ \sigma_{\max}\left(\mathbf{X}\left(\mathbf{b}^{(q)} - 2b_i^{(q)}\mathbf{e}_{i,N}\right)\right), \quad (8)$$

where $\mathbf{e}_{i,N}$ is the i-th column of $\mathbf{I}_N$ and, thus, $\mathbf{b}^{(q)} - 2b_i^{(q)}\mathbf{e}_{i,N}$ is the binary vector that results by flipping the i-th entry of $\mathbf{b}^{(q)}$. If $\sigma_{\max}(\mathbf{X}(\mathbf{b}^{(q)} - 2b_k^{(q)}\mathbf{e}_{k,N})) > \sigma_{\max}(\mathbf{X}(\mathbf{b}^{(q)}))$ (i.e., the best possible flipping offers an increase to the target metric), then we flip the k-th entry of $\mathbf{b}^{(q)}$ to obtain the update

$$\mathbf{b}^{(q+1)} = \mathbf{b}^{(q)} - 2b_k^{(q)}\mathbf{e}_{k,N}. \quad (9)$$

Algorithm 2: BF-TUCKER2 for $d = 1$

Input: $\underline{\mathbf{X}}$

$q \leftarrow 1, \mathbf{b}^{(q)} \leftarrow \mathrm{sgn}(\mathbf{y}); \mathbf{y} \sim \mathcal{N}(\mathbf{0}_N, \mathbf{I}_N)$
  $m' \leftarrow \sigma_{\max}(\mathbf{X}(\mathbf{b}^{(q)}))$
  While $q < Q$, do
    For every $i \in \{1, 2, \ldots, N\}$, do
      $\omega_i \leftarrow \sigma_{\max}(\mathbf{X}(\mathbf{b}^{(q)} - 2b_i^{(q)}\mathbf{e}_{i,N}))$
    $k \leftarrow \underset{i \in \{1,2,\ldots,N\}}{\mathrm{argmax}} \; \omega_i$
    If $\omega_k > m'$, $m' \leftarrow \omega_k$
      $\mathbf{b}^{(q+1)} \leftarrow \mathbf{b}^{(q)} - 2b_k^{(q)}\mathbf{e}_{k,N}, q \leftarrow q + 1$
      $(\mathbf{u}_{bf}, \mathbf{v}_{bf}) \leftarrow \mathtt{dsv}(\mathbf{X}(\mathbf{b}^{(q)}))$
    else, break

Output: $(\mathbf{u}_{\text{bf}}, \mathbf{v}_{\text{bf}})$

Fig. 2. BF-TUCKER2 algorithm for rank-1 L1-TUCKER2 decomposition of $\underline{\mathbf{X}}$. $\mathtt{dsv}(\cdot)$ returns the dominant singular vectors of its matrix argument.

If there is no k such that flipping the k-th entry of $\mathbf{b}^{(q)}$ increases the metric, then the iterations terminate. Let us denote by Q the terminating iteration index. Upon termination, the algorithm conducts SVD to $\mathbf{X}(\mathbf{b}_{\text{opt}}^{(Q)})$ and returns the dominant singular vector pair $(\mathbf{u}_{\text{bf}}, \mathbf{v}_{\text{bf}})$ as the (approximate) solution to L1-TUCKER2 ($d = 1$).

Convergence/Termination: By definition, the above procedure increases the metric of (7) at each bit-flipping iteration. Since the metric is upper-bounded by $\sigma_{\max}(\mathbf{X}(\mathbf{b}_{\text{opt}}))$ and takes a finite number of values (no more than $2^N = |\{\pm 1\}^N|$), the presented bit-flipping iterations will converge in finite steps. In fact, our numerical studies show that the iterations usually converge at $Q < N$ (see Fig. 4 in Section 5). Therefore, for computational stability, we limit the number of iterations to $Q = 2N$ (i.e., we terminate iterations at $q = 2N$). BF-TUCKER2 for $d = 1$ is summarized in Fig. 2.

Complexity: At each iteration, the algorithm calculates N times the singular value of a $D \times M$ matrix, in order to find the best bit to flip. Thus, the cost of each iteration is $\mathcal{O}(DMN \min\{D, M\})$ [24]. Limiting $Q \leq 2N$, as discussed above, yields a total maximum cost of $\mathcal{O}(DMN^2 \min\{D, M\})$ for BF-TUCKER2 with $d = 1$ –i.e., cost quadratic in N and $\min\{D, M\}$ and linear in $\max\{D, M\}$.

4.2. BF-TUCKER2 for $d \geq 1$

Here, we generalize BF-TUCKER2 for rank-d L1-TUCKER2 decomposition. For this generalization, we follow the standard sequential rank-deflation approach. Before we introduce the generalized algorithm, we adjust our notation by defining $\mathbf{X}_i^{(1)} \doteq \mathbf{X}_i$, for every $i \in \{1, 2, \ldots, N\}$. This new superscript will be useful in presenting the deflation process; superscript '(1)' means that the matrix has not been deflated. BF-TUCKER2 for $d \geq 1$ will be presented below as a sequence of d steps.

Step 1: First, we execute the rank-1 BF-TUCKER2 algorithm of Fig. 2 on $\{\mathbf{X}_i^{(1)}\}_{i=1}^N$ to obtain the solution pair $(\mathbf{u}_{bf\,1}, \mathbf{v}_{bf\,1})$.

Step $j \in \{2, 3, \ldots, d\}$: At the beginning of this step, we first rank-deflate $\mathbf{X}_i$, for every $i \in \{1, 2, \ldots, N\}$, to form $\mathbf{X}_i^{(j)} \doteq \mathbf{P}_u^{(p)}\mathbf{X}_i\mathbf{P}_v^{(p)}$, where $\mathbf{P}_u^{(p)} \doteq \mathbf{I}_D - \sum_{p=1}^{j-1} \mathbf{u}_{bf\,p}\mathbf{u}_{bf\,p}^\top$ and $\mathbf{P}_v^{(p)} \doteq \mathbf{I}_D - \sum_{p=1}^{j-1} \mathbf{v}_{bf\,p}\mathbf{v}_{bf\,p}^\top$. Then, similar to step 1, we run again rank-1 BF-TUCKER2, this time on the deflated $\{\mathbf{X}_i^{(j)}\}_{i=1}^N$, to obtain a new rank-1 solution pair $(\mathbf{u}_{bf\,j}, \mathbf{v}_{bf\,j})$.

At the end of the d-th step, the algorithm returns $\mathbf{U}_{bf} = [\mathbf{u}_{bf\,1}, \mathbf{u}_{bf\,2}, \ldots, \mathbf{u}_{bf\,d}]$ and $\mathbf{V}_{bf} = [\mathbf{v}_{bf\,1}, \mathbf{v}_{bf\,2}, \ldots, \mathbf{v}_{bf\,d}]$ as approximate solutions to rank-d L1-TUCKER2. A pseudocode of BF-TUCKER2 for $d \geq 1$ is offered in Fig. 3.

Algorithm 3: BF-TUCKER2 for $d > 1$

Input: $\underline{\mathbf{X}}$, rank d

For every $j \in \{1, 2, \ldots, d\}$, do
  For every $i \in \{1, 2, \ldots, N\}$, do
    $\mathbf{X}_i^{(j)} \leftarrow (\mathbf{I}_D - \sum_{p=1}^{j-1} \mathbf{u}_p\mathbf{u}_p^\top)\mathbf{X}_i(\mathbf{I}_M - \sum_{p=1}^{j-1} \mathbf{v}_p\mathbf{v}_p^\top)$
  $(\mathbf{u}_j, \mathbf{v}_j) \leftarrow \mathtt{Algorithm2}(\{\mathbf{X}_i^{(j)}\}_{i=1}^N)$

Output: $\mathbf{U}_{bf} \leftarrow [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d], \; \mathbf{V}_{bf} \leftarrow [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_d]$

Fig. 3. BF-TUCKER2 algorithm for rank-d L1-TUCKER2 decomposition of $\underline{\mathbf{X}}$. $\mathtt{Algorithm2}(\cdot)$ runs the algorithm of Fig. 2 on its argument.

5. NUMERICAL STUDIES

We commence our studies with an empirical evaluation of the average number of iterations that are required for the bit-flipping iterations in BF-TUCKER2 of Fig. 2 to converge. We generate size-N collection of arbitrary matrices, such that, each entry of $\mathbf{X}_i$ is drawn from the standard-Normal distribution $\mathcal{N}(0, 1)$, for every i. We let N vary in $\{2, 5, 10, 20, 30, 40, 60, 80, 100\}$ and (D, M) vary in $\{(10, 10), (30, 30), (50, 50)\}$, and we run on $\underline{\mathbf{X}}$ the BF-TUCKER2 algorithm of Fig. 2. We compute the mean number of bit-flipping iterations until convergence, over 1000 independent matrix/tensor realizations for each value of the triplet (D, M, N). In Fig. 4, we illustrate the average convergence iteration index versus the number of data matrices, N. We observe that, on average, fewer than N iterations are run before convergence, for every value of (D, M). In addition, for a fixed value of N we see that the mean number of bit-flipping iterations decreases as D and M increase.

We continue our studies by measuring the Performance Degradation Ratio (PDR) attained in the metric of (2) for $d = 1$ (i.e., with respect to the exact solution provided by the new algorithm of Fig. 1) by BF-TUCKER2 of Fig. 2 and TPCA-L1 of [14]. For any approximate solution pair $(\mathbf{u}, \mathbf{v})$, PDR is defined as

$$\Delta(\mathbf{u}, \mathbf{v}) \doteq \frac{\sum_{i=1}^N |\mathbf{u}_{\text{opt}}^\top \mathbf{X}_i \mathbf{v}_{\text{opt}}| - |\mathbf{u}^\top \mathbf{X}_i \mathbf{v}|}{\sum_{i=1}^N |\mathbf{u}_{\text{opt}}^\top \mathbf{X}_i \mathbf{v}_{\text{opt}}|}. \tag{10}$$

To that end, we consider a collection of $N = 10$ arbitrary matrices $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{10} \in \mathbb{R}^{20 \times 20}$. For every $i \in \{1, 2, \ldots, 10\}$, $\mathbf{X}_i$ is of the form $\mathbf{X}_i = \mathbf{Y}_i + \mathbf{N}_i$, where $\mathbf{Y}_i = \alpha_i \mathbf{u}\mathbf{v}^\top$ is the rank-1 signal content of $\mathbf{X}_i$ (i.e., what we would want to reconstruct) and $\mathbf{N}_i$ is additive zero-mean white Gaussian noise (AWGN) with per-entry variance 1. We set $\|\mathbf{u}\|_2 = \|\mathbf{v}\|_2 = 1$ and draw scaling factor α_i from $\mathcal{N}(0, 16)$ independently across i. In Fig. 5, we plot the empirical cumulative distribution function (CDF) of PDR $\Delta(\mathbf{u}, \mathbf{v})$ for BF-TUCKER2 and TPCA-L1 [14]. We observe that the proposed BF-TUCKER2 algorithm attains the exact solution (zero PDR) approximately 85% of the time. On the other hand, TPCA-L1 is optimal about 76% of the time. We also notice that BF-TUCKER2 exhibits PDR of no more than 0.34, with probability 1. On the other hand, TPCA-L1 may reach PDR 0.46 with non-zero probability.

Next, we proceed with a study on the Mean-Squared reconstruction Error (MSE) for $d = 1$ attained by the the proposed exact L1-TUCKER2 (Fig. 1) and BF-TUCKER2 (Fig. 2). We consider a collection of 12 arbitrary matrices, $\mathbf{X}_1, \mathbf{X}_2, \ldots, \mathbf{X}_{12} \in \mathbb{R}^{20 \times 20}$. Similar to the previous study, for every $i \in \{1, 2, \ldots, 12\}$, $\mathbf{X}_i = \mathbf{Y}_i + \mathbf{N}_i$ where $\mathbf{Y}_i = \alpha_i \mathbf{u}_{\text{nom}}\mathbf{v}_{\text{nom}}^\top$, with $\|\mathbf{u}_{\text{nom}}\|_2 = \|\mathbf{v}_{\text{nom}}\|_2 = 1$, and $\mathbf{N}_i$ is zero-mean white Gaussian noise (WGN) with per-entry variance 1. Here, $\alpha_i \sim \mathcal{N}(0, 36)$. $\mathbf{u}_{\text{nom}}$ and $\mathbf{v}_{\text{nom}}$ capture the common nominal rank-1 structure of all matrices $\{\mathbf{X}_i\}_{i=1}^N$. In this study, we consider *additional irregular corruption* (sparse outliers) in the form of additive WGN with variance σ_c^2, added to 34 entries in 2

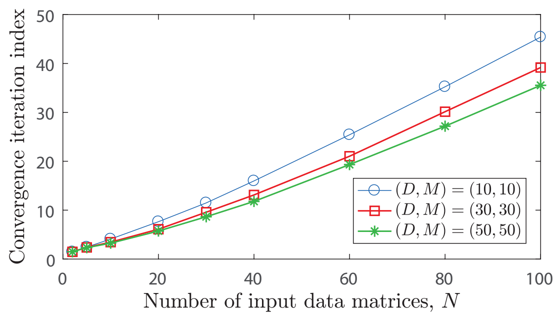

Fig. 4. Number of iterations to convergence vs. number of data matrices N.

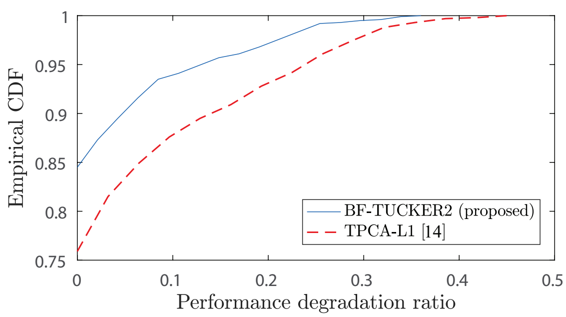

Fig. 5. CDF of PDR with respect to exact rank-1 L1-TUCKER2.

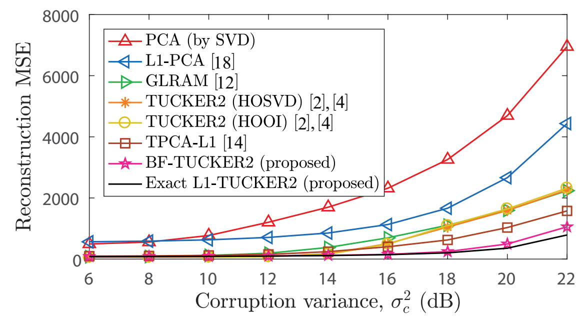

Fig. 6. Rank-1 reconstruction MSE vs. corruption variance σ_c^2.

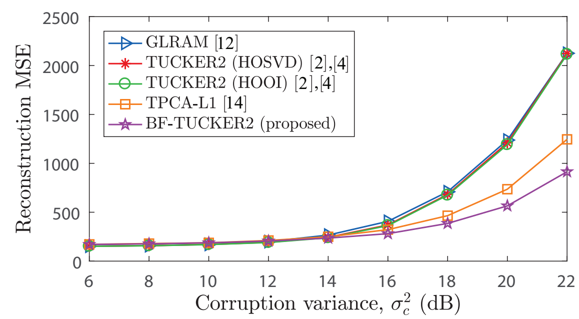

Fig. 7. Rank-3 reconstruction MSE vs. corruption variance σ_c^2.

out of the 12 matrices (i.e., 68 entries out of the total 4800 entries in $\underline{\mathbf{X}}$). Noticing that the rank-1 structure ($\mathbf{u}_{\text{nom}}\mathbf{v}_{\text{nom}}^\top$) is the same for every $\mathbf{X}_i$, we seek to reconstruct the signal content $\mathbf{Y}_i$ by means of joint decomposition of $\{\mathbf{X}_i\}_{i=1}^N$. To that end, we follow one of the two common approaches below. By the first approach, we vectorize the matrix samples and obtain the first ($d = 1$) principal component (PC) of $[\text{vec}(\mathbf{X}_1), \text{vec}(\mathbf{X}_2), \ldots, \text{vec}(\mathbf{X}_{12})]$, $\mathbf{q}$. Then, for every i, we approximate $\mathbf{Y}_i$ by $\hat{\mathbf{Y}}_i = \text{mat}(\mathbf{q}\mathbf{q}^\top \text{vec}(\mathbf{X}_i))$, where $\text{mat}(\cdot)$ reshapes its vector argument into a 20×20 matrix, reversing the operation of $\text{vec}(\cdot)$. In this approach we calculate $\mathbf{q}$ by both PCA (i.e., SVD) and L1-PCA [17]. In the second approach, we process the samples in their natural form, as matrices, analyzing them by rank-1 TUCKER2-type processing. If $(\mathbf{u}, \mathbf{v})$ is the TUCKER2 solution pair, then we approximate $\mathbf{Y}_i$ by $\hat{\mathbf{Y}}_i = \mathbf{u}\mathbf{u}^\top \mathbf{X}_i \mathbf{v}\mathbf{v}^\top$. In this second approach, we calculate $(\mathbf{u}, \mathbf{v})$ by HOSVD [4] , HOOI [2], GLRAM [12], TPCA-L1 [14], the proposed BF-TUCKER2, and the proposed exact L1-TUCKER2. Then, for each reconstruction method and bursty-corruption variance $\sigma_c^2 \in \{6, 8, \ldots, 22\}$dB, we measure the mean of the squared error $\sum_{i=1}^{12} \left\| \mathbf{Y}_i - \hat{\mathbf{Y}}_i \right\|_F^2$ by averaging over 1000 independent realizations of noise and bursty-corruption. In Fig. 6, we plot the reconstruction MSE for every method, versus σ_c^2. We notice that both methods of the first approach (vectorization and PCA-type reconstruction) start from a higher MSE compared to the methods of the second approach (TUCKER2-type reconstruction), arguably, due to the vectorization operation. GLRAM [12], HOSVD [4], and HOOI [2] exhibit almost identical reconstruction MSE, higher than that of both proposed methods. Exact L1-TUCKER2 outperforms every counterpart across the board, exhibiting the strongest outlier resistance. Also, interestingly, despite its low computational cost, BF-TUCKER2 attains outlier resistance similar to that of exact L1-TUCKER2.

We conclude our studies with reconstruction MSE evaluation for $d > 1$. Specifically, we consider $d = 3$ and collection of $N = 10$ matrices $\{\mathbf{X}_i\}_{i=1}^{10}$. The i-th matrix is again of form $\mathbf{X}_i = \mathbf{Y}_i + \mathbf{N}_i$ where $\mathbf{Y}_i = \mathbf{U}_{\text{nom}} \mathbf{\Sigma}_{3\times 3}^{(i)} \mathbf{V}_{\text{nom}}^\top$ is the rank-($d = 3$) signal content of $\mathbf{X}_i$. Extending our $d = 1$ study above, it now holds that $\mathbf{U}_{\text{nom}}^\top \mathbf{U}_{\text{nom}} = \mathbf{V}_{\text{nom}}^\top \mathbf{V}_{\text{nom}} = \mathbf{I}_3$ and $\mathbf{\Sigma}_{3\times 3}^{(i)}$ is a diagonal matrix with diagonal entries drawn from $\mathcal{N}(0, 36)$. We apply HOSVD [4], HOOI [2], GLRAM [12], TPCA-L1 [14], and BF-TUCKER2 (Fig. 3) on $\underline{\mathbf{X}}$ and by each method we obtain a solution pair $(\mathbf{U}, \mathbf{V})$ – wishing to approximate the nominal basis pair $(\mathbf{U}_{\text{nom}}, \mathbf{V}_{\text{nom}})$. Then we approximate $\mathbf{Y}_i$ by $\hat{\mathbf{Y}}_i = \mathbf{U}\mathbf{U}^\top \mathbf{X}_i \mathbf{V}\mathbf{V}^\top$. Similar to our previous study, irregular bursty corruption in the form of sparse additive WGN is drawn from $\mathcal{N}(0, \sigma_c^2)$ and added to 8 entries in 2 out of the 10 matrices (16 out of the total 1000 entries in $\{\mathbf{X}_i\}_{i=1}^{10}$). We let $\sigma_c^2 \in \{6, 8, \ldots, 22\}$dB and, for each method, we plot in Fig. 7 the reconstruction MSE, averaged over 1000 independent noise and bursty corruption realizations. Interestingly, all methods exhibit identical (or very similar) reconstruction performance for values of corruption variance below 12dB. However, as σ_c^2 increases, all L2-norm based methods (HOSVD [4], HOOI [2], and GLRAM [12]) become quickly misled by outliers, exhibiting high reconstruction MSE. TPCA-L1 of [14] outperforms the L2-norm based methods, exhibiting some robustness. The proposed BF-TUCKER2 algorithm of Fig. 3 offers even higher robustness, yielding lower MSE than any counterpart, for σ_c^2 above 14dB.

6. CONCLUSIONS

We showed for the first time that rank-1 L1-TUCKER2 tensor decomposition can be solved as a combinatorial optimization over antipodal-binary variables and provided the first exact solver. Then, we presented BF-TUCKER2, an efficient bit-flipping algorithm that approximates the exact solution to rank-1 L1-TUCKER2, attaining optimality with very high probability. Then, by means of standard rank-deflation, we generalized BF-TUCKER for rank-d tensor decomposition. Our numerical studies show that the proposed exact and efficient algorithms exhibit strong outlier resistance, outperforming all tested counterparts.

7. REFERENCES

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

[2] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, pp. 455-500, 2009.

[3] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *Proc. 7th European Conf. on Comput. Vision (ECCV 2002)*, Copenhagen, Denmark, May 2002, pp. 447-460.

[4] L.D. Lathauwer, B.D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. App.,* vol. 21, pp. 1253-1278, 2000.

[5] P. R. B. Gomes, A. L. F. de Almeida, and J. P. C. L. de Costal, "Fourth-order tensor method for blind spatial signature estimation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (IEEE ICASSP 2014),* Florence, Italy, May 2014, pp. 2992-2996.

[6] M. Mørup, L. Hansen, K. Lars, and S. M. Arnfred, "Algorithms for sparse nonnegative TUCKER decompositions," *Neural computation (MIT Press)*, vol. 20, pp. 2112-2131, 2008.

[7] Y.-D. Kim, E. Parl, S. Yoo, T. Choi, L. Yang, and D. Shin, "Compression of deep convolutional neural networks for fast and low power mobile applications," arXiv:1511.06530 [cs], Feb., 2016.

[8] J. Sun, S. Papadimitriou, C. Lin, N. Cao, S. Liu, and W. Qian, "Multivis: Content-based social network exploration through multi-way visual analysis," in *Proc. SIAM Int. Conf. Data Mining (SDM)*, Sparks, NV, May 2009, pp. 1064-1075.

[9] T. G. Kolda and J. Sun, "Scalable tensor decompositions for multi-aspect data mining," in *Proc. IEEE Int. Conf. Data Mining,* Pisa, Italy, Dec. 2008, pp. 363-372.

[10] J. Sun, D. Tao, S. Papadimitriou, P. Yu, and C. Faloutsos, "Incremental tensor analysis: Theory and applications," *ACM Trans. Knowl. Disc. Data*, vol. 2, pp. 1-37, Oct. 2008.

[11] N. Sidiropoulos, L. De Lathauwer, X. Fu, K. Huang, E. Papalexakis, and C. Faloutsos, "Tensor decomposition for signal processing and machine learning," KU Leuven, ESAT-STADIUS, Leuven, Belgium, Tech. Rep. 16-34, 2016.

[12] J. Ye, "Generalized low rank approximations of matrices," *Mach. Learn.,* vol. 61, pp. 167-191, Nov. 2005.

[13] H. Lu, K. N. Plataniotis and A. N. Venetsanopoulos, "MPCA: Multilinear principal component analysis of tensor objects," *IEEE Trans. Neural Net.*, vol. 19, no. 1, pp. 18-39, Jan. 2008.

[14] Y. Pang, X. Li, and Y. Yuan, "Robust tensor analysis with L1-norm," *IEEE Trans. Circuits Syst. Video Technol.,* vol. 20, pp. 172-178, Feb. 2010.

[15] X. Fu, K. Huang, W. K. Ma, N. D. Sidiropoulos, and R. Bro, "Joint tensor factorization and outlying slab suppression with applications," *IEEE Trans. Signal Process.,* vol. 63, pp. 6315-6328, Dec. 2015.

[16] X. Cao, X. Wei, Y. Han, and D. Lin, "Robust face clustering via tensor decomposition," *IEEE Trans. Cybern.,* vol. 45, pp. 2546-2557, Nov. 2015.

[17] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Some options for L1-subspace signal processing," in *Proc. 10th Intern. Sym. Wireless Commun. Sys. (IEEE ISWCS 2013)*, Ilmenau, Germany, Aug. 2013, pp. 1-5.

[18] P. P. Markopoulos, G. N. Karystinos and D. A. Pados, "Optimal algorithms for L1-subspace signal processing," *IEEE Trans. Signal Process.*, vol. 62, pp. 5046-5058, Oct. 2014.

[19] N. Tsagkarakis, P. P. Markopoulos, and D. A. Pados, "Direction finding by complex L1-principal component analysis," in *Proc. IEEE Int. Workshop on Signal Proc. Adv. Wireless Comm. (IEEE SPAWC 2015)*, Stockholm, Sweden, Jun. 2015, pp. 475-479.

[20] P. P. Markopoulos, S. Kundu, and D. A. Pados, "L1-fusion: Robust linear-time image recovery from few severely corrupted copies," in *Proc. IEEE Int. Conf. on Image Process. (IEEE ICIP 2015)*, Quebec City, Canada, Sep. 2015, pp. 1225-1229.

[21] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: Johns Hopkins University Press, 1996.

[22] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, "Efficient L1-norm Principal-Component Analysis via bit flipping," *IEEE Trans. Signal Process.*, vol. 65, pp. 4252-4264, Aug. 2017.

[23] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, "L1-norm principal-component analysis via bit flipping," in *Proc. IEEE Int. Conf. Mach. Learn. Appl. (IEEE ICMLA 2016)*, Anaheim, CA, USA, Dec. 2016, pp. 326-332.

[24] A. K. Cline and I. S. Dhillon. "Computation of the Singular Value Decomposition," in *Handbook of Linear Algebra*, 2nd ed., L. Hogben Ed. Boca Raton, FL: CRC Press, pp 58-1–58-13, 2017.