Conformity evaluation and L_1-norm principal-component analysis of tensor data

Konstantinos Tountas[a], Dimitris A. Pados[a], and Michael J. Medley[b]

[a]Department of Electrical and Computer Engineering & Computer Science
Florida Atlantic University, Boca Raton, FL 33431
[b]US Air Force Research Laboratory, Rome, NY 13440

ABSTRACT

Multi-modal tensor data sets arise with increasing frequency in modern day scientific and engineering applications, for example in biomedical sciences and autonomous engineered systems. Over the past twenty years, tensor-domain data analysis has been attempted primarily in the context of standard (L_2-norm) eigenvector decompositions across tensor domains. The algorithms are not joint-tensor-domain optimal and exhibit the familiar sensitivity to faulty/corrupted/missing measurements that characterizes all L_2-norm principal-component analysis methods.

In this work, we present a robustified method to evaluate the conformity of tensor data entries with respect to the whole accessible data set. Conformity evaluation is based on a continuously refined sequence of calculated L_1-norm tensor subspaces. The theoretical developments are illustrated in the context of a multisensor localization application that indicates unprecedented estimation performance and resistance to intermittent disturbances. An electroencephalogram (EEG) data analysis experiment is also presented.

Keywords: L_1-norm, data conformity, conformity evaluation, tensors, tensor decomposition, anomaly detection, principal-component analysis.

1. INTRODUCTION

In many real-world applications such as social networking data mining,[1] video processing, biomedical signal processing, wireless communications,[2,3] and network traffic analysis,[4,5] the use of high-dimensional tensor datasets (multi-way arrays of order greater than two) has become prominent in order to capture relationships between multi-modal data. The dimensionality of such data sets is usually high. Fortunately, high dimensional data sets have frequently an intrinsic low-rank structure that can be captured through dimensionality reduction techniques. Subspace estimation methods such as principal-component analysis (PCA) has been widely applied for dimensionality reduction in matrix data sets due to their simplicity and effectiveness. Given a data set, PCA finds a set of projection vectors (called principal components) that maximize a norm of the projection of the data points. Conventional PCA uses the L_2-norm and is prone to misrepresentation of the data characteristics in the presence of outliers (the effect of outliers is exaggerated by the use of the L_2-norm.) To alleviate this problem, there has been a growing interest in robust PCA calculations by explicit L_1-norm projection maximization.[6] The exact calculation of L_1-norm principal components was presented in[7] for the first time in the literature. Later, suboptimal algorithms were developed[8] for fast computation of the L_1 principal components.

For tensor data sets, similar to matrix PCA, the TUCKER decomposition strives to jointly decompose the collected n-way tensors and unveil the low-rank multi-linear data structure. Higher-Order SVD (HOSVD)[9] and Higher-Order Orthogonal Iteration (HOOI)[10] algorithms are well-known solvers for TUCKER decompositions. A detailed presentation of TUCKER decompositions and the respective solvers is offered in.[10,11] Note that both types of solvers can generally only guarantee a locally optimal solution. Both HOSVD and HOOI base the calculation of the underlying subspaces on the L_2-norm.[12,13] Similar to,[6] the work in[14] reformulates the

Konstantinos Tountas: E-mail: ktountas2017@fau.edu
Dimitris A. Pados.: E-mail: dpados@fau.edu
Michael J. Medley: E-mail: michael.medley@us.af.mil

TUCKER2 problem ($(n-1)$-way analysis of the tensor, where one dimension of the tensor (usually the sample space) is not taken into account) and use the L_1-norm for alleviating the effects of the outliers.

In spite of their robustness, for a given data set with potential outliers, the existing L_1-TUCKER methods calculate "one-shot" L_1-norm subspaces, which can still be away from the true nominal subspaces of interest when the data sets are significantly contaminated. In this paper, we propose an iterative approach that generates an iteratively refined sequence of improved L_1-norm subspaces. In each iteration, for each mode of the tensor, the conformity of each sample is inferred by its distance relative to the L_1-norm subspaces calculated in the previous iteration. Highly conforming samples tend to be nominal samples; samples with lower conformity are more likely to be outliers. Then, all the samples of the original tensor data set are weighted by their conformity values and the L_1-norm principal components of each mode are re-calculated. This way the contribution of outlying samples in the tensor data set is further suppressed, resulting in improved L_1-norm subspaces.

The remaining of this paper is organized as follows. In Section 2, we provide a detailed presentation of the TUCKER decomposition, L_1-norm PCA, as well as the necessary terminology and operations involving tensors. In Section 3, we develop the iteratively refined L_1-norm PCA algorithm for tensor data. In Section 4, the effectiveness of the proposed algorithm is demonstrated through two experiments: (i) joint direction-of-arrival estimation and signature identification in an all-spectrum communication system, and (ii) supervised classification of smoking versus non-smoking patients from EEG data from the UCI machine learning data set. Finally, a few conclusions are drawn in Section 5.

2. BACKGROUND

Next we describe necessary terminology and operations regarding tensors. We also provide an overview of the Tucker model and L_1-PCA. In Table 1, we list the notation used throughout the paper.

2.1 Tensors and Tensor Operations

The *order* of a tensor is the number of its dimensions, also known as ways or modes. A *fiber* is a vector extracted from a tensor by fixing all modes but one. A *slice* of a tensor is a matrix extracted from a tensor by fixing all modes but two. In particular, the $[\mathbf{X}]_{:,:,n}$ slices of a third-order tensor $\mathcal{X}$ are called the frontal ones and we succinctly denote them as $\mathbf{X}_n$. *Matricization*, also known as *reshaping* or *unfolding*, logically reorganizes tensors into other forms without changing the tensor values themselves. The mode-n matricization of a N-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_N}$ is denoted by $\mathbf{X}_{(k)} \in \mathbb{R}^{I_n \times I_1 I_2 \ldots I_{K-1} I_{n+1} \ldots I_N}$ and arranges the mode-n fibers of the tensor as columns of the resulting matrix. The *tensor product* or *outer product* of vectors $\mathbf{a} \in \mathbb{R}^{D \times 1}$ and $\mathbf{b} \in \mathbb{R}^{L \times 1}$ is defined as the $D \times L$ matrix $\mathbf{a} \circledcirc \mathbf{b}$ with elements $[\mathbf{a} \circledcirc \mathbf{b}]_{i,j} = [\mathbf{a}]_i \, [\mathbf{b}]_j \,, \forall i, j$. Introducing a third vector $\mathbf{c} \in \mathbb{R}^{N \times 1}$ we can generalize to the outer product of three vectors, which is an $D \times L \times N$ third-order tensor $\mathbf{a} \circledcirc \mathbf{b} \circledcirc \mathbf{c}$ with elements $[\mathbf{a} \circledcirc \mathbf{b} \circledcirc \mathbf{c}]_{i,j,k} = [\mathbf{a}]_i \, [\mathbf{b}]_j \, [\mathbf{c}]_k \,, \forall i, j, k$.

Symbol	Definition
$\mathcal{X}, \mathbf{X}, \mathbf{x}, x$	Tensor, matrix, vector, scalar
$[\mathbf{X}]_{:,i}\,, [\mathbf{X}]_{i,:}$	i-th column and row of $\mathbf{X}$ (same for tensors)
$\operatorname{diag}(\mathbf{x})$	Diagonal matrix with vector $\mathbf{x}$ on the diagonal
$\operatorname{vec}(\mathbf{X})$	Stacking all input matrix columns in vector form
$\mathbf{X}_n$	Shorthand for $[\mathbf{X}]_{:,:,n}$ (n-th frontal slice of tensor $\mathcal{X}$)
$\mathbf{X}_{(n)}$	Mode-n matricization of tensor $\mathcal{X}$
$\circ$	Hadamard product (elementwise product of two tensors)
$\otimes$	Kronecker product
$\circledcirc$	Outer product

Table 1: Notation used throughout the paper.

2.2 TUCKER Model

The TUCKER decomposition of a third-order tensor $\mathcal{X} \in \mathbb{R}^{D \times L \times N}$ strives to jointly decompose the collected three-dimensional tensor and unveil the low-rank multilinear structure of the underlying data distribution. Formally, the decomposition approximates the tensor by a sum of three outer products

$$\mathcal{X} \approx \sum_{r=1}^{R} \mathbf{u}_r \circledcirc \mathbf{v}_r \circledcirc \mathbf{w}_r, \tag{1}$$

where $\mathbf{u}_r \in \mathbb{R}^{D \times 1}$, $\mathbf{v}_r \in \mathbb{R}^{L \times 1}$, and $\mathbf{w}_r \in \mathbb{R}^{N \times 1}$, and $\mathbf{u}_i^T \mathbf{u}_j = \mathbf{v}_i^T \mathbf{v}_j = \mathbf{w}_i^T \mathbf{w}_j = 0, \forall i, j, i \neq j$. If we assemble the column vectors $\mathbf{u}_r$, $\mathbf{v}_r$, $\mathbf{w}_r$ as: $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_R] \in \mathbb{R}^{D \times R}$, $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_R] \in \mathbb{R}^{K \times R}$, and $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_R] \in \mathbb{R}^{N \times R}$, then $\mathbf{U}$, $\mathbf{V}$, $\mathbf{W}$ are called the *factor matrices*. An equivalent, more intuitive formulation of (1) w.r.t the frontal slices $\mathbf{X}_n$ of the input tensor $\mathcal{X} \in \mathbb{R}^{D \times L \times N}$ is

$$\mathbf{X}_n \simeq \mathbf{U}\mathbf{S}_n\mathbf{V}^H, \quad k = 1, 2, \ldots, N, \tag{2}$$

where $\mathcal{S} \in \mathbb{R}^{R \times R \times N}$ is an auxiliary tensor. Each frontal slice $\mathbf{S}_n$ of $\mathcal{S}$ contains the row vector $[\mathbf{W}]_{n,:}$ along its diagonal, i.e., $\mathbf{S}_k = diag\left([\mathbf{W}]_{n,:}\right)$. Equation (2) provides an intuitive interpretation of the TUCKER decomposition, through its correspondence with L_2-norm PCA: each slice $\mathbf{X}_n$ is decomposed to a set of factor matrices $\mathbf{U}$, $\mathbf{V}$ (similar to the singular vectors) which are common for all the slices, and a diagonal middle matrix (similar to the singular values) which varies for each n-th slice. The low-rank approximation problem can be written as

$$\left(\widehat{\mathbf{U}}, \widehat{\mathbf{V}}, \widehat{\mathbf{W}}\right) = \underset{\substack{\mathbf{U} \in \mathbb{R}^{D \times R_1}, \mathbf{U}^T\mathbf{U} = \mathbf{I}_{R_1}, \\ \mathbf{V} \in \mathbb{R}^{L \times R_2}, \mathbf{V}^T\mathbf{V} = \mathbf{I}_{R_2}, \\ \mathbf{W} \in \mathbb{R}^{N \times R_3}, \mathbf{W}^T\mathbf{W} = \mathbf{I}_{R_3}}}{\operatorname{argmax}} \left\| \mathbf{U}^T \mathbf{X}_1 \left(\mathbf{V} \otimes \mathbf{W}\right) \right\|_F^2, \tag{3}$$

where $\|\cdot\|_F$ returns the summation of the absolute squares of the input matrix elements. The most popular algorithm for fitting the TUCKER model is the TUCKER-Alternating Least Squares (ALS); the main idea behind this method is to solve for one factor matrix at a time by fixing the rest. In that way, each subproblem is reduced to a linear least-squares problem.

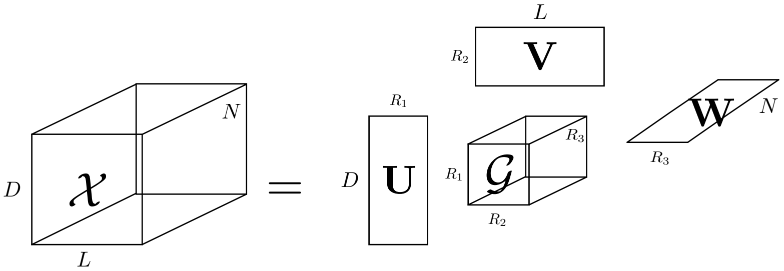

Figure 1: Compact (reduced) TUCKER model: R_1, R_2, R_3 are the mode (row, column, fiber, respectively) ranks of $\mathcal{X}$.

2.3 L_1-norm Principal Component Analysis

For $N = 1$, the third-order tensor $\mathcal{X} \in \mathbb{R}^{D \times L \times N}$ is converted to a $D \times L$ matrix $\mathbf{X}$. The low-rank approximation problem in (3) is then simplified to

$$(\mathbf{U}, \mathbf{V}) = \underset{\substack{\mathbf{U} \in \mathbb{R}^{D \times R_1}, \mathbf{U}^T\mathbf{U} = \mathbf{I}_{R_1}, \\ \mathbf{V} \in \mathbb{R}^{L \times R_2}, \mathbf{V}^T\mathbf{V} = \mathbf{I}_{R_2}}}{\operatorname{argmax}} \left\| \mathbf{U}^T \mathbf{X} \mathbf{V} \right\|_F^2, \tag{4}$$

for which the optimal $\mathbf{U}$ and $\mathbf{V}$ are built optimally by the R_1 left-hand and R_2 right-hand singular vectors of $\mathbf{X}$, respectively.

In the matrix case the columns of $\mathbf{X}$ are the real-valued sample vectors of the data matrix; it is of interest to calculate the principle components corresponding to the features of our data, not the samples. Hence, the rank-R_1 approximation problem can be written as

$$\mathbf{Q}_{L_2} = \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times R_1},\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_{R_1}}}{\operatorname{argmax}} \left\|\mathbf{X}^T\mathbf{Q}\right\|_2 . \tag{5}$$

Motivated by its resistance to anomalous data L_1-norm decomposition was proposed for robust low-rank subspace computation. Replacing the L_2-norm in (5) by the L_1-norm the so-called L_1-norm PCA calculates principal components in the form

$$\mathbf{Q}_{L_1} = \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times R_1},\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_{R_1}}}{\operatorname{argmax}} \left\|\mathbf{X}^T\mathbf{Q}\right\|_1 , \tag{6}$$

where $\|\cdot\|_1$ returns the sum of the absolute values of its input matrix elements. $\mathbf{Q}_{L_1}$ in (6) is likely to be closer to the true nominal rank-R_1 subspace than $\mathbf{Q}_{L_2}$. As shown in,[6] the exact calculation of the L_1-norm principal components in Eq. (6) can be written as a combinatorial problem

$$\begin{aligned}\mathbf{Q}_{L_1} &= \max_{\substack{\mathbf{Q}\in\mathbb{R}^{D\times R_1},\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_{R_1}}} \left\|\mathbf{X}^T\mathbf{Q}\right\|_1 = \max_{\substack{\mathbf{Q}\in\mathbb{R}^{D\times R_1},\\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_{R_1}}} \max_{\mathbf{B}\in\{\pm1\}^{L\times R_1}} \operatorname{tr}\left(\mathbf{Q}^T\mathbf{X}\mathbf{B}\right)\\ &= \max_{\mathbf{B}\in\{\pm1\}^{L\times R_1}} \operatorname{tr}\left(\mathbf{B}\mathbf{Q}^T\mathbf{X}\right) = \max_{\mathbf{B}\in\{\pm1\}^{L\times R_1}} \left\|\mathbf{X}\mathbf{B}\right\|_* ,\end{aligned} \tag{7}$$

The optimal solution for (7) can be obtained by exhaustive search in the space of the binary antipodal matrix $\mathbf{B}$ with complexity $\mathcal{O}\left(2^{LR_1}\right)$. By denoting $\mathbf{B}_{\text{OPT}}$ the optimal solution, we perform SVD on $\mathbf{X}\mathbf{B}_{\text{OPT}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$ and form the L_1-norm principal components as $\mathbf{Q}_{L_1} = [\mathbf{U}]_{:,R_1}\mathbf{V}^T$. Therefore, the overall complexity for finding R_1 L_1-norm principal components via exhaustive search is $\mathcal{O}\left(2^{LR_1}\min\left\{D^2R_1, DR_1^2\right\}\right)$.

In,[8] a fast greedy single-bit-flipping (SBF) algorithm was proposed for the computation of the rank-1 L_1-norm principal components with complexity $\mathcal{O}\left(L^3\right)$. The suboptimal algorithm is extended in[7] for the calculation of the $R_1 < \operatorname{rank}(\mathbf{X})$ L_1-norm principal components of $\mathbf{X}$ with complexity $\mathcal{O}\left(LD\min\{L, D\}\right.$ $\left.+L^2\left(R_1^4 + \operatorname{rank}(\mathbf{X})R_1^2\right) + \operatorname{rank}(\mathbf{X})LR_1^3\right)$. The SBF algorithm for rank $R_1 > 1$ starts with an initial binary matrix $\mathbf{B}^{\text{init}} \in \{\pm1\}^{L\times R_1}$ and iteratively produces a sequence of new binary matrices $\mathbf{B}^l, l = 1, 2, \ldots,$ where $\mathbf{B}^{l+1}$ at the $(l+1)$-th iteration differs from $\mathbf{B}^l$ only in a single bit position. At each iteration, just the bit that results in the highest increase of $\|\mathbf{X}\mathbf{B}\|_*$ is flipped to reach a (suboptimal) solution of (7), denoted as $\widehat{\mathbf{B}}$. The associated L_1-norm projection operator $\mathbf{Q}_{L_1}$ can be obtained by performing the following steps: 1) Perform SBF with input $\mathbf{B}^{\text{init}}$ to obtain $\widehat{\mathbf{B}}$, 2) Perform SVD on $\mathbf{X}\widehat{\mathbf{B}} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$, and 3) Return $\mathbf{Q}_{L_1} = [\mathbf{U}]_{:,R}\mathbf{V}^T$.

3. ITERATIVE $L1$-TENSOR DECOMPOSITION WITH DATA CONFORMITY CONSIDERATIONS

3.1 Methodology

Motivated by the challenges presented in Section 2, we propose an iterative algorithm for tensor decomposition, that exploits the outlier resistance properties of L_1-norm PCA. In the following, we describe the design of our algorithm for decomposing the tensor and calculating the subspaces. Without loss of generality, we present our algorithmic developments for a third order tensor $\mathcal{X} \in \mathbb{R}^{D\times L\times N}$. In the initialization step of the algorithm, with respect to the first mode of the tensor $\mathbf{X}_{(1)} \in \mathbb{R}^{D\times LN}$ (unfolding along the columns of the tensor) we calculate the R_1 principal components $\mathbf{Q}_1^{(0)} \in \mathbb{R}^{D\times R_1}$ by solving the maximization problem in (6). The resulting basis emphasizes the subspace spanned by the columns of the original tensor $\mathcal{X}$ that contain nominal data and de-emphasizes columns that are contaminated with anomalous data. Next, we calculate the orthogonal projection of each column $\left[\mathbf{X}_{(1)}\right]_{:,i_1}, i_1 = 1, \ldots, LN$ on the calculated subspace $\mathbf{Q}_1^{(0)}$ as

$$d_{1,i_1}^{(1)} = \left\|\mathbf{Q}_1^{(0)}\mathbf{Q}_1^{(0)^T}\left[\mathbf{X}_{(1)}\right]_{:,i_1}\right\|_2 \quad \forall i_1 = 1, 2, \ldots, LN. \tag{8}$$

We expect small $d_{1,i_1}^{(1)}$ values if $\left[\mathbf{X}_{(1)}\right]_{:,i_1}$ is an anomalous data point and large $d_{1,i_1}^{(1)}$ values if $\left[\mathbf{X}_{(1)}\right]_{:,i_1}$ is a nominal data point. After the calculation of the projection of each column on the subspace, the conformity values are brought back to tensor form

$$\mathcal{W}_1^{(1)} = \text{tensorization}\left(\begin{bmatrix} d_{1,1}^{(1)} & d_{1,2}^{(1)} & \dots & d_{1,LN}^{(1)} \end{bmatrix}_{LN\times 1}^{T} \circ \mathbf{1}_{D\times LN}\right) \in \mathbb{R}^{D\times L\times N}, \tag{9}$$

where $\mathbf{1}_{D\times LN}$ stands for an all-ones matrix of dimension $D \times LN$, and the *tensorization* operator converts the argument matricized tensor, to its original tensor form. The tensor $\mathcal{W}_1^{(1)}$ contains the conformity values corresponding to each column of the original tensor $\mathcal{X}$. The same procedure is repeated for the rest of the modes calculating the corresponding conformity tensors, $\mathcal{W}_2^{(1)}, \mathcal{W}_3^{(1)}$.

With all the mode conformity tensors calculated, we then weigh them according to the "importance" of the corresponding dimension and combine them in an additive fashion. The final Step-1 conformity tensor $\widetilde{\mathcal{W}}^{(1)}$ is then normalized so that each element is in the $[0, 1]$ range,

$$\widetilde{\mathcal{W}}^{(1)} = \frac{\sum_{k=1}^{3}\alpha_k \mathcal{W}_k^{(1)} - \min\left(\sum_{k=1}^{3}\alpha_k \mathcal{W}_k^{(1)}\right)}{\max\left(\sum_{k=1}^{3}\alpha_k \mathcal{W}_k^{(1)}\right) - \min\left(\sum_{k=1}^{3}\alpha_k \mathcal{W}_k^{(1)}\right)} \tag{10}$$

where $\alpha_1, \alpha_2, \alpha_3 \in \mathbb{R}^+, \sum_{k=1}^{3}\alpha_k = 1$ (for example, $\alpha_1 = \alpha_2 = \alpha_3 = \frac{1}{3}$), model the presumed importance of the corresponding dimension of the original tensor.

A new L_1-norm subspace of each tensor mode is then calculated on the conformity-adjusted data tensor by element-by-element multiplication by the conformity tensor. For mode one for example, we calculate

$$\mathbf{Q}_1^{(1)} = \underset{\substack{\mathbf{Q}\in\mathbb{R}^{D\times R_1}, \\ \mathbf{Q}^T\mathbf{Q}=\mathbf{I}_{R_1}}}{\operatorname{argmax}} \left\| \mathbf{Q}^T \left(\mathbf{X}_{(1)} \circ \widetilde{\mathbf{W}}_{(1)}^{(1)}\right)\right\|_1, \tag{11}$$

where $\circ$ is the element-wise (Hadamard) product. The same holds for rest of the modes. The subspaces are iteratively refined until numerical convergence of the data conformity tensor $\widetilde{\mathcal{W}}^{(l)}$ is observed. In Table 1, we present the pseudo-code corresponding to the proposed algorithm for a general tensor $\mathcal{X}^{I_1\times I_2\times\cdots\times I_K}$.

Table 2 Iterative L_1-norm tensor decomposition

```
Input: $\mathcal{X} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$, ranks $R_1, \ldots, R_K \in \mathbb{Z}^+$, and weights $\alpha_1, \ldots, \alpha_K, \in \mathbb{R}^+, \sum_{k=1}^{K} \alpha_k = 1$
Output: $\widetilde{\mathcal{W}} \in \mathbb{R}^{I_1 \times I_2 \times \cdots \times I_K}$
for k = 1, ..., K do
    $\mathbf{Q}_k^{(0)} = \underset{\substack{\mathbf{Q} \in \mathbb{R}^{I_k \times R_k}, \\ \mathbf{Q}^T \mathbf{Q} = \mathbf{I}_{R_k}}}{\text{argmax}} \left\| \mathbf{Q}^T \mathbf{X}_{(k)} \right\|_1$
    $M_k = \prod_{i=1, i \neq k}^{K} I_i, \quad l = 1$
end for
while convergence criterion is not met do
    for $k = 1, \ldots, K$ do
        $d_{k,i_k}^{(l)} = \left\| \mathbf{Q}_k^{(l-1)} \mathbf{Q}_k^{(l-1)^T} \left[ \mathbf{X}_{(k)} \right]_{:,i_k} \right\|_2, \quad \forall i_k = 1, 2, \ldots, M_k$
        $\mathcal{W}_k^{(l)} \leftarrow \text{tensorization}_k \left( \begin{bmatrix} d_{k,1}^{(l)} & d_{k,2}^{(l)} & \ldots & d_{k,M_k}^{(l)} \end{bmatrix}_{M_k \times 1}^T \circ \mathbf{1}_{M_k \times I_k} \right)$
    end for
    $\widetilde{\mathcal{W}}^{(l)} = \frac{\sum_{k=1}^{K} \alpha_k \mathcal{W}_k^{(l)} - \min\left(\sum_{k=1}^{K} \alpha_k \mathcal{W}_k^{(l)}\right)}{\max\left(\sum_{k=1}^{K} \alpha_k \mathcal{W}_k^{(l)}\right) - \min\left(\sum_{k=1}^{K} \alpha_k \mathcal{W}_k^{(l)}\right)}$
    for $k = 1, \ldots, K$ do
        $\mathbf{Q}_k^{(l)} = \underset{\substack{\mathbf{Q} \in \mathbb{R}^{I_k \times R_k}, \\ \mathbf{Q}^T \mathbf{Q} = \mathbf{I}_{R_k}}}{\text{argmax}} \left\| \mathbf{Q}^T \left( \mathbf{X}_{(k)} \circ \widetilde{\mathbf{W}}_{(k)}^{(l)} \right) \right\|_1$
    end for, $l = l + 1$
end while
```

4. SIMULATION STUDIES

4.1 Joint Direction-of-Arrival and Waveform Identification

We consider K single-antenna spread-spectrum beacons transmitting over a single-input single-output (SISO) additive white Gaussian noise (AWGN) channel. Particularly, the k-th beacon, $k = 1, \ldots, K$ transmits symbols $b_k[n], n = 1, 2, \ldots, N$ from a complex alphabet $\mathcal{C}$, at rate $1/T$, modulated by a spread-spectrum digital waveform $d_k(t)$ of duration T. The transmitted signal is represented by

$$x_k(t) \triangleq \sum_{n=1}^{N} \sqrt{P_k} b_k [n] \, s_k (t - nT) \, e^{2\pi f_c t} \tag{12}$$

where $P_k > 0$ denotes the transmitted energy per bit, and f_c is the carrier frequency. The spread-spectrum digital waveform is of the form

$$s_k(t) \triangleq \sum_{l=0}^{L-1} \mathbf{d}_k[l] g_{T_c} (t - lT_c) \tag{13}$$

where $\mathbf{d}_k$ is a length L normalized antipodal sequence, i.e. $\mathbf{d}_k \in \left\{ \pm 1/\sqrt{L} \right\}^L$, uniquely allocated to beacon k, and $g_{T_c}(\cdot)$ is a square-root raised cosine (SRRC) pulse-shaping waveform of duration T_c, so that $T = LT_c$.

The beacon signals are received by a D element antenna array from an azimuth angle ϕ and an elevation angle θ. The size $D \times 1$ array response vector $\mathbf{a}(\phi, \theta)$ to a far-field received signal that impinges on the array, is given by

$$\mathbf{a}(\phi, \theta) = \exp \left\{ j 2\pi \frac{1}{\lambda_c} \mathbf{P}^T \mathbf{k}(\phi, \theta) \right\} \in \mathbb{C}^{D \times 1}, \tag{14}$$

where λ_c is the carrier wavelength, the matrix $\mathbf{P}$ contains the antenna array element values

$$\mathbf{P} = \begin{bmatrix} x_1 & x_2 & \ldots & x_D \\ y_1 & y_2 & \ldots & y_D \\ z_1 & z_2 & \ldots & z_D \end{bmatrix} \in \mathbb{R}^{3\times D} \tag{15}$$

and the vector $\mathbf{k}(\phi, \theta)$ represents the projection of the received signal's steering vector on the antenna array coordinate system defined as

$$\mathbf{k}(\phi, \theta) = \begin{bmatrix} \cos(\theta)\sin(\phi) \\ \cos(\theta)\cos(\phi) \\ \sin(\theta) \end{bmatrix} \in \mathbb{R}^{3\times 1}. \tag{16}$$

Upon carrier demodulation at the carrier frequency f_c, pulse matched filtering, sampling at T_c, and buffering L samples, the n-th received space-code data snapshots is given by

$$\mathbf{Y}_n = \sum_{k=1}^{K} \sqrt{P_k} b_k\left[n\right] \mathbf{a}(\phi_k, \theta_k)\mathbf{s}_k^T + \mathbf{N}_n \in \mathbb{C}^{M\times L}, \quad n = 1, 2, \ldots, N, \tag{17}$$

where $[\mathbf{N}[n]]_{i,j}$ is an additive noise component with $\mathbb{E}\left\{\mathbf{N}_n\right\} = \mathbf{0}_{M\times L}$, $\mathbb{E}\left\{\mathbf{n}_i\mathbf{n}_j^H\right\} = \sigma^2\mathbf{I}_M, \forall i, j \in \left\{1, \ldots, L\right\}, i \neq j$, $\mathbb{E}\left\{\mathbf{n}_i^H\mathbf{n}_j\right\} = \sigma^2\mathbf{I}_L \forall i, j \in \left\{1, \ldots, M\right\}, i \neq j$. We define $\mathcal{A}$ as the set of active angle-of-arrival pairs $(\phi_k, \theta_k), k = 1, 2, \ldots, K$, and $\mathcal{D}$ as the set of active waveform sequences $\mathbf{s}_k, k = 1, 2, \ldots, K$.

By defining the power-scaled signature matrix $\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_K] \in \mathbb{R}^{L\times K}$, where $\mathbf{s}_k = \sqrt{P_k}\mathbf{d}_k$, the rank-$M$ steering matrix $\mathbf{A} = \left[\mathbf{a}\left(\phi_1, \theta_1\right), \ldots, \mathbf{a}\left(\phi_K, \theta_K\right)\right] \in \mathbb{C}^{M\times K}$, and a diagonal matrix $\mathbf{B}\left[n\right] =$
$diag\left(\left[b_1\left[n\right], b_2\left[n\right], \ldots, b_K\left[n\right]\right]\right)$, the n-th received data snapshot matrix in (17) can be expressed in the following matrix form

$$\mathbf{Y}_n = \mathbf{A}\mathbf{B}_n\mathbf{S}^T + \mathbf{N}_n, \quad \in \mathbb{C}^{D\times L} \quad n = 1, 2, \ldots, N. \tag{18}$$

We observe that the received matrices can be viewed as slices of an $M \times L \times N$ three-way tensor $\mathcal{Y} \in \mathbb{C}^{D\times L\times N}$. Our goal is to utilize the tensor structure of our data in order to accurately estimate the angles of arrival, as well as identify the waveforms the beacons are using.

In order to begin our algorithmic developments, we define the real-valued representation $\overline{\mathbf{A}} \in \mathbb{R}^{2m\times 2n}$ of any complex-valued matrix $\mathbf{A} \in \mathbb{C}^{m\times n}$ by concatenating real and imaginary parts as follows

$$\overline{\mathbf{A}} = \begin{bmatrix} \mathrm{Re}\left\{\mathbf{A}\right\}, & -\mathrm{Im}\left\{\mathbf{A}\right\} \\ \mathrm{Im}\left\{\mathbf{A}\right\}, & \mathrm{Re}\left\{\mathbf{A}\right\} \end{bmatrix} \in \mathbb{R}^{2m\times 2n}, \tag{19}$$

where $\mathrm{Re}\left\{\cdot\right\}$ and $\mathrm{Im}\left\{\cdot\right\}$ return the real and imaginary part of each matrix element, respectively. The transition from $\mathbf{A} \in \mathbb{C}^{m\times n}$ to $\overline{\mathbf{A}} \in \mathbb{R}^{2m\times 2n}$ is based on what is commonly referred to as complex-number *realification* in representation theory. Realification allows for any complex system of equations to be converted and solved through a real system. Before we proceed, we review briefly in the Lemma bellow some fundamental properties of matrix realification.

LEMMA 4.1. *For any* $\mathbf{A} \in \mathbb{C}^{m\times n}$ *and* $\mathbf{B} \in \mathbb{C}^{n\times q}$ *the following hold* $\overline{(\mathbf{A}+\mathbf{B})} = \overline{\mathbf{A}} + \overline{\mathbf{B}}$, $\overline{(\mathbf{A}\mathbf{B})} = \overline{\mathbf{A}}\,\overline{\mathbf{B}}$, *and* $\overline{(\mathbf{A}^H)} = \overline{\mathbf{A}}^T$.

By (19) and Lemma 1, we define the n-th *real-valued* snapshot $\overline{\mathbf{Y}}_n$ as

$$\overline{\mathbf{Y}}_n = \overline{\mathbf{A}}\,\overline{\mathbf{B}}_n\,\overline{\mathbf{S}}^T + \overline{\mathbf{N}}_n \in \mathbb{R}^{2D\times 2L}, \quad n = 1, 2, \ldots, N \tag{20}$$

With the realification operation, we convert our complex tensor $\mathcal{Y} \in \mathbb{C}^{D\times L\times N}$ to a real-valued tensor $\overline{\mathcal{Y}} \in \mathbb{R}^{2D\times 2L\times N}$. From (20), we notice that the angle components of interest lie in the $R_1 \geq 2K$ dimensional space $\mathcal{S}_A = \mathrm{span}\left(\overline{\mathbf{A}}\right)$, while the waveform components of interest lie in the $2K$ dimensional space $\mathcal{S}_D = \mathrm{span}\left(\overline{\mathbf{S}}\right)$. The following propositions highlight the utility of $\mathcal{S}_A$ and $\mathcal{S}_D$ for estimating the target directions of arrival (DoA's) and waveform sequences in $\mathcal{A}$ and $\mathcal{D}$, respectively.

Proposition 1. *For any pair* $(\phi, \theta) \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$, $\operatorname{span}\left(\bar{\mathbf{a}}\left(\phi, \theta\right)\right) \subseteq \mathcal{S}_A$ *if and only if* $(\phi, \theta) \in \mathcal{A}$. *Set equality holds only if* $K = 1$. By proposition 1, we can accurately decide whether some pair $(\phi, \theta) \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$ is a target DoA or not by means of any orthonormal basis $\mathbf{Q}_1 \in \mathbb{R}^{2D \times 2K}$ that spans $\mathcal{S}_A$ as

$$\left(\mathbf{I}_{2K} - \mathbf{Q}_1 \mathbf{Q}_1^T\right) \bar{\mathbf{a}}\left(\phi, \theta\right) = \mathbf{0}_{2K \times 2} \Leftrightarrow (\phi, \theta) \in \mathcal{A}. \tag{21}$$

Thus, in view of proposition 1, and in accordance to common practice, target DoAs in $\mathcal{A}$ can be approximated by the K highest peaks of the MUSIC-like pseudo-spectrum defined as

$$P_A\left(\phi, \theta\right) = \left\|\left(\mathbf{I}_{2D} - \mathbf{Q}_1 \mathbf{Q}_1^T\right) \bar{\mathbf{a}}\left(\phi, \theta\right)\right\|^{-2} \tag{22}$$

The same holds for the waveform estimation problem, as stated in the following proposition

Proposition 2. *For any* $\mathbf{s} \in \left\{\pm \frac{1}{\sqrt{L}}\right\}^L$, $\operatorname{span}\left(\bar{\mathbf{s}}\right) \subseteq \mathcal{S}_D$ *if and only if* $\mathbf{s} \in \mathcal{D}$. *Set equality holds only if* $K = 1$. By 2, the target sequences can be estimated by any $R_2 \geq 2K$ dimensional orthonormal basis $\mathbf{Q}_2 \in \mathbb{R}^{2L \times 2K}$ that spans $\mathcal{S}_D$ as

$$\left(\mathbf{I}_{2K} - \mathbf{Q}_2 \mathbf{Q}_2^T\right) \bar{\mathbf{s}} = \mathbf{0}_{2L \times 2} \Leftrightarrow \bar{\mathbf{s}} \in \mathcal{D}. \tag{23}$$

And the corresponding pseudo-spectrum is

$$P_D\left(\bar{\mathbf{s}}\right) = \left\|\left(\mathbf{I}_{2L} - \mathbf{Q}_2 \mathbf{Q}_2^T\right) \bar{\mathbf{s}}\right\|^{-2}. \tag{24}$$

Peaks in the two pseudo-spectra in (22) and (24) identify DoAs and waveform sequences without associating angles and sequences. In order to have this correspondence, we create all possible DoA pairs and waveform sequence candidates as

$$\mathbf{z}\left(\phi, \theta, \mathbf{s}\right) = \mathbf{a}\left(\phi, \theta\right) \otimes \mathbf{s} \in \mathbb{C}^{ML \times 1}. \tag{25}$$

Then, we calculate the K principle components, $\mathbf{Q}_{sc}$, of the space-code matrix $\mathbf{Y}_{sc} = \left[\operatorname{vec}\left(\mathbf{Y}_1\right), \operatorname{vec}\left(\mathbf{Y}_2\right), \ldots, \operatorname{vec}\left(\mathbf{Y}_N\right)\right]$ by means of SVD. We utilize the following pseudo-spectrum to find the peaks corresponding to the exact DoAs and waveform sequence associations

$$P_{SC}\left(\mathbf{z}\right) = \left\|\left(\mathbf{I}_{DL} - \mathbf{Q}_{sc} \mathbf{Q}_{sc}^H\right) \mathbf{z}\right\|^{-2}. \tag{26}$$

Algorithm 2 Joint direction-of-arrival estimation and waveform identification

```
Input: $\mathbf{Q}_1 \in \mathbb{R}^{2M \times R_1}$, $\mathbf{Q}_2 \in \mathbb{R}^{2L \times R_2}$, $\mathcal{Y} \in \mathbb{R}^{2M \times 2L \times N}$
Output: $\{(\phi_k, \theta_k)\} \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$, and $\{\mathbf{s}_k\} \in \left\{\pm 1/\sqrt{L}\right\}^L, k = 1, \ldots, K$
Calculate $P_A(\phi, \theta) = \left\|\left(\mathbf{I}_{2D} - \mathbf{Q}_1 \mathbf{Q}_1^T\right) \bar{\mathbf{a}}(\phi, \theta)\right\|^{-2}, \forall (\phi, \theta) \in \left(-\frac{\pi}{2}, \frac{\pi}{2}\right]$
Find the peaks of the $P_A(\mathbf{s})$ pseudo-spectrum
Calculate $P_D(\bar{\mathbf{s}}) = \left\|\left(\mathbf{I}_{2L} - \mathbf{Q}_2 \mathbf{Q}_2^T\right) \bar{\mathbf{s}}\right\|^{-2}, \forall \bar{\mathbf{s}} \in \left\{\pm 1/\sqrt{2L}\right\}^{2L}$
Find the peaks of the $P_c(\mathbf{s})$ pseudo-spectrum
Calculate $\mathbf{Q}_{sc} = \operatorname{SVD}\left(\left[\operatorname{vec}(\mathbf{Y}_1), \operatorname{vec}(\mathbf{Y}_2), \ldots, \operatorname{vec}(\mathbf{Y}_N)\right]\right)$
Calculate the space-code candidates $\mathbf{z}(\phi, \theta, \mathbf{s}) = \mathbf{a}(\phi, \theta) \otimes \mathbf{s} \in \mathbb{C}^{ML \times 1}$
Calculate $P_{sc}(\mathbf{z}) = \left\|\left(\mathbf{I}_{ML} - \mathbf{Q}_{sc} \mathbf{Q}_{sc}^H\right) \mathbf{z}\right\|^{-2}$
Find the peaks of the $P_{sc}(\mathbf{z})$ pseudo-spectrum
```

We test the effectiveness of our proposed joint direction-of-arrival estimation and signature detection method on a system with two single-antenna, all-spectrum transmitters, transmitting $N = 10$ BPSK symbols on signatures of length $L = 8$ from angles $(\phi_1 = -50, \theta_1 = 20), (\phi_2 = 15, \theta_2 = 60)$ to a receiver equipped with a triangular array. A single-antenna, spread-spectrum interferer is transmitting with probability $p = 0.3$ on the same signature

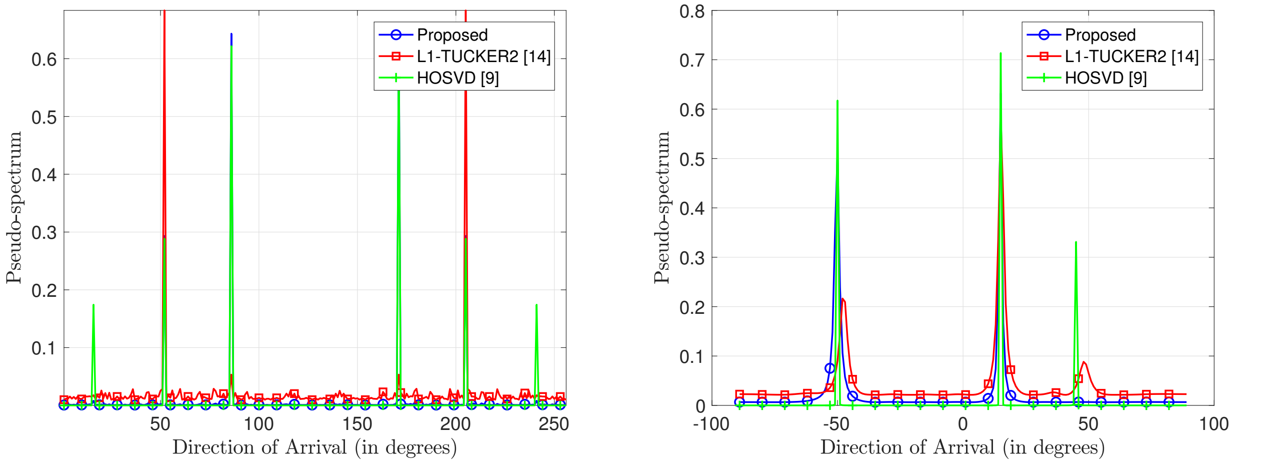

Figure 2: Left: Signature estimation spectra (two targets and one interferer), Right: DoA estimation spectra (two targets and one interferer).

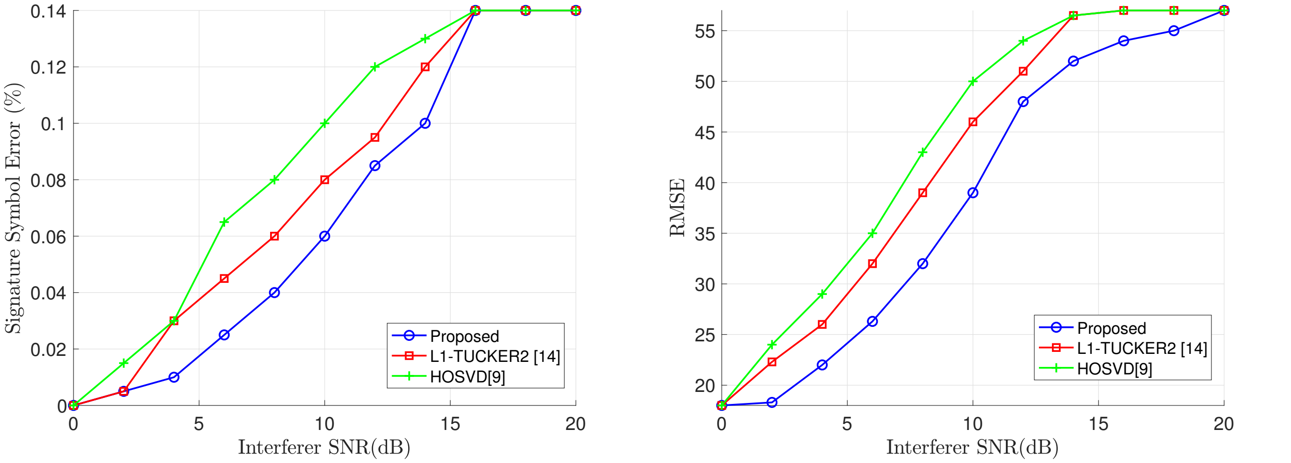

Figure 3: Left: Absolute error percentage of signature estimation, Right: Root-mean-squared-error (RMSE) of DoA estimation.

as transmitter 1, disrupting the joint DoA and Identification operation. Fig.2-Right depicts the pseudo-spectrum corresponding to the elevation angle, while 2-Left depicts the pseudo-spectrum corresponding to the signatures. It is clear that our proposed iteratively-refined L_1-norm tensor decomposition method produces an uncorrupted spectrum, in contrast to the HOSVD spectrum which is corrupted by the interferer. Fig. 3-Left depicts the root-mean-squared-error (RMSE), over 5000 independent realizations, of the proposed DoA estimation method, L_1-TUCKER and the standard L_2-norm PCA-based one (MUSIC), as a function of the signal-to-noise ratio (SNR) of the interferer transmitter. As shown, the proposed method offers much better RMSE performance compared against HOSVD, as well as L_1-TUCKER2. The absolute symbol error (percentage) of the signature identification is show in Fig. 3-Right. Again, the proposed method outperforms HOSVD and L_1-TUCKER.

4.2 Classification by Conformity L_1-norm PCA

We consider the problem of unsupervised classification of alcoholic versus non-alcoholic patients from electroencephalogram (EEG) data.[15] The data set contains measurements from 64 electrodes placed on subject's scalps which were sampled at 256 Hz (3.9-msec epoch) for 1 second, forming a series of real matrices with dimensions 64 × 256. Two groups of subjects exist: alcoholic and control. Each subject was exposed to either a single stimulus (S1) or to two stimuli (S1 and S2) which were pictures of objects chosen from the 1980 Snodgrass and

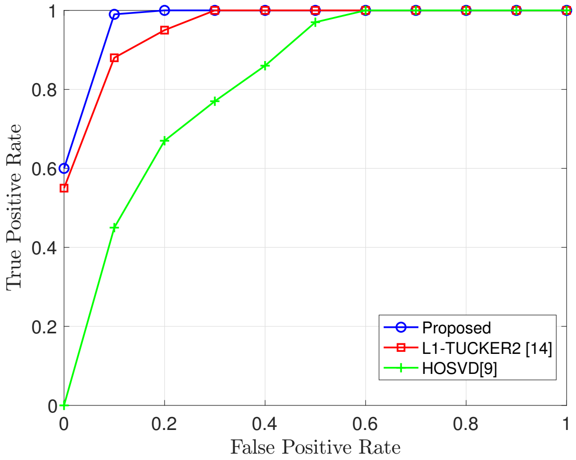

Figure 4: The receiver operating characteristic (ROC) curves of the EEG data set.

Vanderwart picture set. When two stimuli were shown, they were presented in either a matched condition where S1 was identical to S2 or in a non-matched condition where S1 differed from S2.

For our experiments we create a tensor data set $\mathcal{X} \in \mathbb{R}^{60\times 64\times 256}$, where 60 is the number of patients, 64 is the number of electrodes, and 256 is the number of samples per patient. The data set consists of 54 control patients and 6 alcoholic patients, which we consider the outliers class. We calculate the subspaces corresponding to each dimension of the tensor $\mathcal{X}$ with $R_1 = 20, R_2 = 20, R_3 = 40$ principal components per dimension. Subsequently, the L_2-norm projection of each slice $[\mathbf{X}]_{n,:,:} \in \mathbb{R}^{64\times 256}, n = 1, \ldots, 60$ corresponding to each patient on the subspaces $\left\|\mathbf{Q}_2^T [\mathbf{X}]_{n,:,:} \mathbf{Q}_3\right\|_F$ is calculated as the outlier score, and the samples with the lowest score are detected as outliers, i.e. alcoholic patients. We carried out 100 independent experiments, and plot the receiver operating characteristic curve (ROC) of the proposed algorithm compared agains HOSVD. The ROC plots the true positive rate versus the false positive rate. As shown in Fig. 4, the proposed iteratively-refined L_1 tensor decomposition method demonstrates the ability to capture the outliers significantly better than HOSVD.

5. CONCLUSION

We presented a novel, iteratively refined L_1-norm tensor decomposition algorithm, which measures the conformity of each tensor element with respect to the whole tensor data set. In each iteration, the L_1-norm subspace of each dimension computed in the previous iteration is utilized to calculate the conformity of each data sample. Non-conforming samples are weighted down, resulting in subspaces that better represent the nominal subspace of each tensor dimension. The proposed method significantly enhances performance in tasks such as DoA estimation and outlier detection, compared to traditional L_2-norm methods and L_1-TUCKER2 methods.

Acknowledgment

The work of D. A. Pados was supported by the Schmidt Family Foundation, Boca Raton, FL.

REFERENCES

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

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

[3] M. Haardt, F. Roemer, and G. Del Galdo, “Higher-order svd-based subspace estimation to improve the parameter estimation accuracy in multidimensional harmonic retrieval problems,” *IEEE Trans. Signal Processing*, vol. 56, no. 7, pp. 3198–3213, July 2008.

[4] 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 *Proceedings of the SIAM International Conference on Data Mining, SDM 2009, April 30 - May 2, 2009, Sparks, Nevada, USA*, 2009, pp. 1064–1075.

[5] T. G. Kolda and J. Sun, “Scalable tensor decompositions for multi-aspect data mining,” in *Eighth IEEE Int. Conf. on Data Mining*, Dec 2008, pp. 363–372.

[6] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, “Optimal algorithms for L_1-subspace signal processing,” *IEEE Trans. Signal Processing*, vol. 62, no. 19, pp. 5046–5058, Oct. 2014.

[7] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, “Efficient L_1-norm principal-component analysis via bit flipping,” *IEEE Trans. Signal Processing*, vol. 65, no. 16, pp. 4252–4264, Aug. 2017.

[8] S. Kundu, P. P. Markopoulos, and D. A. Pados, “Fast computation of the L_1-principal component of real-valued data,” in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP)*, May 2014, pp. 8028–8032.

[9] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, “A multilinear singular value decomposition,” *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253–1278, 2000.

[10] ——, “On the best rank-1 and rank-(r1, r2, , rn) approximation of higher-order tensors,” *SIAM J. Matrix Anal. Appl.*, vol. 21, pp. 1324–1342, 2000.

[11] T. G. Kolda and B. W. Bader, “Tensor decompositions and applications,” *SIAM Rev.*, vol. 51, pp. 455–500, 2009.

[12] J. Sun, D. Tao, S. Papadimitriou, P. S. Yu, and C. Faloutsos, “Incremental tensor analysis: Theory and applications,” *ACM Trans. Knowl. Discov. Data*, vol. 2, no. 3, pp. 11:1–11:37, Oct. 2008.

[13] X. Li, M. K. Ng, G. Cong, Y. Ye, and Q. Wu, “Mr-ntd: Manifold regularization nonnegative tucker decomposition for tensor data dimension reduction and representation,” *IEEE Trans. on Neural Networks and Learning Systems*, vol. 28, no. 8, Aug 2017.

[14] P. P. Markopoulos, D. G. Chachlakis, and E. E. Papalexakis, “The exact solution to rank-1 l1-norm tucker2 decomposition,” *IEEE Signal Processing Letters*, vol. 25, no. 4, pp. 511–515, April 2018.

[15] H. Begleiter, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml