PROCEEDINGS OF SPIE

SPIEDigitalLibrary.org/conference-proceedings-of-spie

Options for multimodal classification based on L1-Tucker decomposition

Dimitris G. Chachlakis, Mayur Dhanaraj, Ashley Prater-Bennette, Panos P. Markopoulos

Dimitris G. Chachlakis, Mayur Dhanaraj, Ashley Prater-Bennette, Panos P. Markopoulos, "Options for multimodal classification based on L1-Tucker decomposition," Proc. SPIE 10989, Big Data: Learning, Analytics, and Applications, 109890O (13 May 2019); doi: 10.1117/12.2520140

SPIE.

Event: SPIE Defense + Commercial Sensing, 2019, Baltimore, Maryland, United States

Options for Multimodal Classification Based on L1-Tucker Decomposition

Dimitris G. Chachlakis[a], Mayur Dhanaraj[a], Ashley Prater-Bennette[b], and Panos P. Markopoulos[a]

[a]Dept. of Electrical and Microelectronic Engineering, Rochester Institute Of Technology, Rochester NY, USA
[b]Air Force Research Laboratory, Rome NY, USA

ABSTRACT

Most commonly used classification algorithms process data in the form of vectors. At the same time, modern datasets often comprise multimodal measurements that are naturally modeled as multi-way arrays, also known as tensors. Processing multi-way data in their tensor form can enable enhanced inference and classification accuracy. Tucker decomposition is a standard method for tensor data processing, which however has demonstrated severe sensitivity to corrupted measurements due to its L2-norm formulation. In this work, we present a selection of classification methods that employ an L1-norm-based, corruption-resistant reformulation of Tucker (L1-Tucker). Our experimental studies on multiple real datasets corroborate the corruption-resistance and classification accuracy afforded by L1-Tucker.

Keywords: Classification, L1-norm Tucker, multi-modal data, tensor processing, Tucker decomposition.

1. INTRODUCTION

Classification is the task of detecting the class which an unknown data sample belongs to. In its supervised and semi-supervised versions, classification relies on an available collection of training samples from the classes of interest. Commonly, data samples are organized and processed in the form of vectors. Over the past decades, an array of successful vector-based classification algorithms have been presented in the literature, including, for example, Support Vector Machines (SVM) [1], k-Nearest Neighbors ($k-$NN) [2], Naive Bayes [3], Random Forest [4], Nearest Subspace (NS) [5, 6], and Artificial Neural Network (ANN) classifiers following Deep Learning (DL) paradigm [7, 8].

Modern datasets comprise large volumes of measurements, collected across diverse sensing modalities, and naturally organized in higher-order arrays (matrices, tensors). Over the past few years, it has been documented that analyzing multi-way data in their natural tensor form can enable the discovery of patterns and underlying structures that can significantly enhance inference and learning [9]. Accordingly, taking into account inter-modality dependencies of the data, tensor processing has also been shown to be beneficial in terms of classification accuracy [10]. Over the past decade, multiple works in the literature have presented classification algorithms that employ tensor processing, commonly carried out by Tucker decomposition [11] –implemented by means of Higher Order Singular Value Decomposition (HOSVD)– or Canonical Polyadic Decomposition (CPD) [12, 13].

At the same time, rigorous studies have shown that Tucker and CPD are significantly sensitive against outliers (samples that significantly deviate from the nominal ones) [14, 15], due to their L2-norm-based formulations. In this work, we present a selection of successful tensor-based methods for classification of multi-way data

Further author information: (Send correspondence to P.P.M.)
D.G.C. : E-mail: dimitris@mail.rit.edu; Telephone: +1 585 733 9355
M.D. : E-mail: mxd6023@rit.edu; Telephone: +1 585 743 0245
A.P. : E-mail: ashley.prater.3@us.af.mil; Telephone: +1 315 330 2804
P.P.M. : E-mail: panos@rit.edu; Telephone: +1 585 475 7917

Funding: This research is supported in part by the Air Force Office of Scientific Research, under grant 18RICOR029, and the National Science Foundation, under grant 1808582.

Big Data: Learning, Analytics, and Applications, edited by Fauzia Ahmad,
Proc. of SPIE Vol. 10989, 109890O · © 2019 SPIE · CCC code:
0277-786X/19/$18 · doi: 10.1117/12.2520140

and combine them with corruption-resistant L1-Tucker. Then, we conduct experimental studies on real-world datasets, which corroborate that L1-Tucker decomposition exhibits significant robustness against outliers among the processed data.

2. PROBLEM STATEMENT

We consider $L \geq 2$ classes of N-way measurements of size $D_1 \times D_2 \times \ldots \times D_N$. For supervised classification, we consider availability of T_i training data samples $\boldsymbol{\mathcal{X}}_1^{(i)}, \ldots, \boldsymbol{\mathcal{X}}_{T_i}^{(i)} \in \mathbb{R}^{D_1 \times \ldots \times D_N}$ from class i, collated across the $(N+1)$-th (sample) mode of $\boldsymbol{\mathcal{X}}^{(i)} \in \mathbb{R}^{D_1 \times \ldots \times D_N \times T_i}$. A supervised classifier will be trained on $\{\boldsymbol{\mathcal{X}}^{(i)}\}_{i=1}^L$ so as to be able to detect the source-class of any new (testing) sample $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{D_1 \times \ldots \times D_N}$.

In standard vector processing, for any i, t, tensor sample $\boldsymbol{\mathcal{X}}_t^{(i)}$ would be vectorized into $\mathbf{x}_t^{(i)} \in \mathbb{R}^W$, where $W = \prod_{n=1}^N D_n$. Accordingly, $\boldsymbol{\mathcal{X}}^{(i)}$ would be mode-$(N+1)$ unfolded/flattened* into

$$\mathbf{X}^{(i)} = [\boldsymbol{\mathcal{X}}^{(i)}]_{(N+1)}^\top = [\mathbf{x}_1^{(i)}, \ldots, \mathbf{x}_{T_i}^{(i)}] \in \mathbb{R}^{W \times T_i}. \tag{1}$$

Then, a standard classifier (e.g., SVM) would be trained on $\{\mathbf{X}^{(i)}\}_{i=1}^L$ and tested on new vector $\mathbf{y} = \text{vec}(\boldsymbol{\mathcal{Y}}) \in \mathbb{R}^W$.

In this work, we focus on classifiers that rely on tensor analysis of the original tensor training and testing measurements, $\{\boldsymbol{\mathcal{X}}^{(i)}\}_{i=1}^L$ and $\boldsymbol{\mathcal{Y}}$, respectively. In the next Section, we briefly review Tucker and L1-Tucker tensor decompositions.

3. L1-TUCKER ANALYSIS OF TENSOR DATA

3.1 Review of Tucker Formulation

Consider N-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1 \times D_2 \times \ldots \times D_N}$ and integers $d_1, d_2, \ldots, d_N$ such that $d_n \leq D_n \ \forall n$. Standard (L2-norm based) *Tucker* decomposition seeks to solve

$$\underset{\substack{\boldsymbol{\mathcal{G}} \in \mathbb{R}^{d_1 \times d_2 \times \ldots \times d_N} \\ \{\mathbf{Q}_n \in \mathbb{R}^{D_n \times d_n};\ \mathbf{Q}_n^\top \mathbf{Q}_n = \mathbf{I}_{d_n}\}_{n=1}^N}}{\text{minimize}} \left\| \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_{n \in [N]} \mathbf{Q}_n \right\|_F^2, \tag{2}$$

where the L2-norm $\|\cdot\|_F^2$ returns the summation of squared entries of its input tensor argument and, for brevity, $[N] = \{1, 2, \ldots, N\}$ and $\times_{n \in [N]} \mathbf{Q}_n = \times_1 \mathbf{Q}_1 \times_2 \mathbf{Q}_2 \times_3 \ldots \times_N \mathbf{Q}_N$. Denoting by $\boldsymbol{\mathcal{G}}^{\text{L2}}$ and $\{\mathbf{Q}_n^{\text{L2}}\}_{n=1}^N$ the optimal core and factors, respectively, it can be shown that $\boldsymbol{\mathcal{G}}^{\text{L2}} = \boldsymbol{\mathcal{X}} \times_{n \in [N]} \mathbf{Q}_n^{\text{L2}\top}$. Accordingly, Tucker decomposition in (2) can be simplified to a pursuit of $\{\mathbf{Q}_n^{\text{L2}}\}_{n=1}^N$ as

$$\underset{\{\mathbf{Q}_n \in \mathbb{R}^{D_n \times d_n};\ \mathbf{Q}_n^\top \mathbf{Q}_n = \mathbf{I}_{d_n}\}_{n=1}^N}{\text{maximize}} \left\| \boldsymbol{\mathcal{X}} \times_{n \in [N]} \mathbf{Q}_n{}^\top \right\|_F^2. \tag{3}$$

Finally, given Tucker-optimal core and factors, $\boldsymbol{\mathcal{X}}$ can be approximated by the *reduced-Tucker-rank* tensor $\widehat{\boldsymbol{\mathcal{X}}}^{\text{L2}} = \boldsymbol{\mathcal{G}}^{\text{L2}} \times_{n \in [N]} \mathbf{Q}_n^{\text{L2}}$, or, equivalently, $\widehat{\boldsymbol{\mathcal{X}}}^{\text{L2}} = \boldsymbol{\mathcal{X}} \times_{n \in [N]} \mathbf{Q}_n^{\text{L2}} \mathbf{Q}_n^{\text{L2}\top}$.

A popular solver for Tucker decomposition is the HOSVD algorithm which approximates $\mathbf{Q}_n^{\text{L2}}$ by means of singular value decomposition (SVD) of the mode-n flattening of $\boldsymbol{\mathcal{X}}$ [17]. That is, HOSVD approximates $\mathbf{Q}_n^{\text{L2}}$ by

$$\mathbf{Q}_n^{\text{HOSVD}} = \left[\Phi([\boldsymbol{\mathcal{X}}]_{(n)})\right]_{:,1:d_n}, \tag{4}$$

where, for any $D \times M$ matrix $\mathbf{A}$ with SVD $\mathbf{U}\boldsymbol{\Sigma}_{D \times M}\mathbf{V}^\top$, $\Phi(\mathbf{A}) = \mathbf{U}$ contains its left-hand singular vectors, in order of decreasing corresponding singular value.

*Consider tensor $\boldsymbol{\mathcal{A}} \in \mathbb{R}^{D_1 \times \ldots \times D_N}$. For any $n \in \{1, 2, \ldots, N\}$ and any set of indices $\{i_m\}_{m \in \{1,2,\ldots,N\} \setminus n}$, vector $[\boldsymbol{\mathcal{A}}]_{i_1,\ldots,i_{n-1},:,i_{n+1},\ldots,i_N} \in \mathbb{R}^{D_n}$ is one out of the $W_n = \prod_{i \neq n} D_i$ mode-n fibers of $\boldsymbol{\mathcal{A}}$. A matrix $[\mathbf{A}]_{(n)} \in \mathbb{R}^{D_n \times W_n}$ that contains as columns all mode-n fibers of $\boldsymbol{\mathcal{A}}$ is known as the mode-n flattening (or mode-n matrix unfolding) of $\boldsymbol{\mathcal{A}}$ [16].

In many applications of interest, the processed tensor $\boldsymbol{\mathcal{X}}$ is formed by, say, mode-N concatenation of D_N coherent $(N-1)$-way samples. In such cases, it may be of interest to jointly decompose all D_N samples as

$$\underset{\{\mathbf{Q}_n \in \mathbb{R}^{D_n \times d_n};\ \mathbf{Q}_n^\top \mathbf{Q}_n = \mathbf{I}_{d_n}\}_{n=1}^{N-1}}{\text{maximize}} \sum_{j=1}^{D_N} \left\| [\boldsymbol{\mathcal{X}}]_{:,\ldots,:,j} \times_{n \in [N-1]} \mathbf{Q}_n{}^\top \right\|_F^2 \tag{5}$$

which is known as *Tucker2* decomposition and derives from (3) by simply fixing $\mathbf{Q}_N = \mathbf{I}_{D_N}$. Similarly, standard Tucker decomposition in (3) can be viewed as a Tucker2 decomposition of a $(N+1)$-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1 \times D_2 \times \ldots \times D_N \times 1}$. Thus, any Tucker solver (e.g., HOSVD) can be used for Tucker2 decomposition and vice versa.

3.2 L1-Tucker Formulation

Standard Tucker decomposition in (3) relies on the L2-norm, which places squared emphasis on each entry of the processed tensor, benefiting high-magnitude/peripheral entries (outliers). Such entries are typically unexpected and undesired and result due to corruption of the dataset at hand. To remedy the impact of outliers, an L1-norm-based formulation of Tucker (*L1-Tucker*) was recently proposed in [18] as

$$\underset{\{\mathbf{Q}_n \in \mathbb{R}^{D_n \times d_n};\ \mathbf{Q}_n^\top \mathbf{Q}_n = \mathbf{I}_{d_n}\}_{n=1}^{N}}{\text{maximize}} \left\| \boldsymbol{\mathcal{X}} \times_{n \in [N]} \mathbf{Q}_n^\top \right\|_1, \tag{6}$$

where $\|\cdot\|_1$ returns the sum of the absolute values of its argument. Similar to standard Tucker, having obtained a set of orthonormal factors $\{\mathbf{Q}_n^{\text{L1}} \in \mathbb{R}^{D_n \times d_n};\ \mathbf{Q}_n^{\text{L1}}{}^\top \mathbf{Q}_n^{\text{L1}} = \mathbf{I}_{d_n}\}_{n=1}^N$ by solving (6), the core tensor of L1-Tucker is given by $\boldsymbol{\mathcal{G}}^{\text{L1}} = \boldsymbol{\mathcal{X}} \times_{n \in [N]} \mathbf{Q}_n^{\text{L1}}{}^\top$. Accordingly, $\boldsymbol{\mathcal{X}}$ is low rank approximated by $\widehat{\boldsymbol{\mathcal{X}}}^{\text{L1}} = \boldsymbol{\mathcal{G}}^{\text{L1}} \times_{n \in [N]} \mathbf{Q}_n^{\text{L1}}$. L1-Tucker2 can also be formulated by fixing $\mathbf{Q}_N = \mathbf{I}_{D_N}$ in (6). An approximate solution to L1-Tucker and L1-Tucker2 can be obtained by L1-HOSVD, the L1-norm based analogous solver of the standard HOSVD algorithm [18]. L1-HOSVD is presented in Section 3.3.

For the special case of $N = 3$, L1-Tucker2 was also studied in [19]. For $N = 3$ and $d_1 = d_2 = 1$, it was recently shown that L1-Tucker2 can be cast as a combinatorial problem over variables in $\{\pm 1\}$ [15] and the first ever exact solver was offered. In view of the theoretical findings in [15], a novel approximate solver for L1-Tucker2 was offered in [20] for $N = 3$ and any d_1 and d_2. Finally, a state-of-the-art solver for L1-Tucker2 of 3-way tensors was offered in [21].

3.3 L1-HOSVD Algorithm

Motivated by HOSVD, authors in [18] proposed to approximate the n-th orthonormal factor of L1-Tucker in (6) by L1-PCA the mode-n flattening of $\boldsymbol{\mathcal{X}}$. That is, for every n, L1-HOSVD seeks

$$\mathbf{Q}_n^{\text{L1}} = \underset{\mathbf{Q} \in \mathbb{R}^{D_n \times d_n};\ \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_{d_n}}{\text{argmax}} \left\| \mathbf{Q}^\top [\boldsymbol{\mathcal{X}}]_{(n)} \right\|_1. \tag{7}$$

The L1-PCA formulation of (7) has been extensively studied over the past few years. In [22], it was shown that L1-PCA can be cast as a combinatorial problem over antipodal binary variables and the first ever exact algorithms for its solution were offered. Prior to the exact algorithms of [22], a conceptually simple approximate solver based on alternating maximization was presented in [23]. More recently, a state-of-the-art solver was offered in [24] and an adaptive solver for L1-PCA was offered in [25]. Finally, the first ever solvers for L1-PCA of complex-valued matrices were offered in [26]. In this work, we employ the solver of [23] for computing a solution to (7), as shown below.

The optimization problem in (7) is equivalent to [22]

$$\underset{\substack{\mathbf{Q} \in \mathbb{R}^{D_n \times d_n};\ \mathbf{Q}^\top \mathbf{Q} = \mathbf{I}_{d_n} \\ \mathbf{B} \in \{\pm 1\}^{W_n \times d_n}}}{\text{max.}} \text{Tr}\left(\mathbf{Q}^\top [\boldsymbol{\mathcal{X}}]_{(n)} \mathbf{B}\right). \tag{8}$$

L1-HOSVD solver for L1-Tucker (or L1-Tucker2) [18]

Input: $\mathcal{X} \in \mathbb{R}^{D_1 \times \ldots \times D_N}, \{d_n\}_{n=1}^N$

1: $\{\mathbf{Q}_n^{\text{L1-HOSVD}} \leftarrow \mathbf{Q}_n^{\text{HOSVD}}\}_{n=1}^N$

2: for $n = 1, 2, \ldots, N$ % `L1-Tucker2: Substitute` N `by` $N-1$ `and fix` $\mathbf{Q}_N^{\text{L1-HOSVD}} = \mathbf{I}_{D_N}$

3: Until termination/convergence, do

4: $\mathbf{Q}_n^{\text{L1-HOSVD}} \leftarrow \Psi\left([\mathcal{X}]_{(n)} \text{sgn}\left([\mathcal{X}]_{(n)}^\top \mathbf{Q}_n^{\text{L1-HOSVD}}\right)\right)$

Return: $\{\mathbf{Q}_n^{\text{L1-HOSVD}}\}_{n=1}^N$

Figure 1: L1-HOSVD algorithm [18] for solving L1-Tucker (or L1-Tucker2) in (6).

In accordance with the Procrustes Theorem [27], for any fixed $\mathbf{B}$ and SVD $[\mathcal{X}]_{(n)}\mathbf{B} \overset{\text{svd}}{=} \mathbf{U}\mathbf{\Sigma}_{d_n \times d_n}\mathbf{V}^\top$, maximization in (8) is attained for $\mathbf{Q} = \Psi\left([\mathcal{X}]_{(n)}\mathbf{B}\right) = \mathbf{U}\mathbf{V}^\top$. Similarly, for any fixed orthonormal basis $\mathbf{Q}$, maximization in (8) is attained for $\mathbf{B} = \text{sgn}([\mathcal{X}]_{(n)}^\top \mathbf{Q})$. By these observations, and similar to [23], one may initialize to an arbitrary orthonormal basis $\mathbf{Q}^{(0)}$ and perform alternating updates as

$$\mathbf{B}^{(t)} = \text{sgn}\left([\mathcal{X}]_{(n)}^\top \mathbf{Q}^{(t)}\right), \tag{9}$$

$$\mathbf{Q}^{(t+1)} = \Psi\left([\mathcal{X}]_{(n)}\mathbf{B}^{(t)}\right), \tag{10}$$

for $t = 0, 1, \ldots,$. Conveniently, the update of the antipodal binary matrix $\mathbf{B}$ can be absorbed in the update of the orthonormal basis matrix, leading to the compact-form update

$$\mathbf{Q}^{(t+1)} = \Psi\left([\mathcal{X}]_{(n)}\text{sgn}\left([\mathcal{X}]_{(n)}^\top \mathbf{Q}^{(t)}\right)\right). \tag{11}$$

The update in (11) provably increases the L1-PCA metric in (7) and, since the metric is upper bounded by the exact solution of [22], convergence is guaranteed [18]. Steering again our focus to (7), one can initialize $\mathbf{Q}_n^{(0)} = \mathbf{Q}_n^{\text{HOSVD}}$ $\forall n \in [N]$, and perform updates similar to the update in (11), until the L1-PCA metric of (7) ceases to increase. A pseudocode of the presented L1-HOSVD algorithm is offered in Fig. 1.

4. OPTIONS FOR CLASSIFICATION BASED ON L1-TUCKER

There is an array of works in the literature that study tensor-based classification (e.g., see [28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39]). In this section, we present four selected approaches, reformulated to employ L1-Tucker decomposition.

4.1 Approach 1: Low-Rank Samples Approximation by Per-Class L1-Tucker2

Per the notation of Section 2, we consider $(N+1)$-way tensor $\mathcal{X}^{(i)}$ that contains T_i N-way training samples form class i. This approach applies L1-Tucker2 decomposition on $\mathcal{X}^{(i)}$ to obtain class-specific orthonormal factors $\mathbf{Q}_1^{(i)}, \ldots, \mathbf{Q}_N^{(i)}$. The core tensor and lower rank approximation of $\mathcal{X}^{(i)}$ are accordingly computed as $\mathcal{G}^{(i)} = \mathcal{X}^{(i)} \times_{n\in[N]} \mathbf{Q}_n^{(i)\top}$ and $\widehat{\mathcal{X}}^{(i)} = \mathcal{G}^{(i)} \times_{n\in[N]} \mathbf{Q}_n^{(i)}$, respectively. In practice, $\widehat{\mathcal{X}}^{(i)}$ contains reduced-Tucker-rank representations of the original tensor samples, such that similarity among distinct classes (due to noise, or low-variance components) is restrained. Then, for every i, the processed tensor samples are vectorized and arranged as columns of $\widehat{\mathbf{X}}^{(i)} = [\widehat{\mathcal{X}}^{(i)}]_{(N+1)}^\top \in \mathbb{R}^{W \times T_i}$. Finally, $\{\widehat{\mathbf{X}}^{(i)}\}_{i=1}^L$ are given as input to any standard vector-based classifier.

As an example, given testing sample $\mathcal{Y}$ and its vectorized representation $\mathbf{y} = \text{vec}(\mathcal{Y})$, authors in [28] (where Tucker was used instead of L1-Tucker) proposed to classify it to class $i^\star$, if

$$i^\star = \underset{i\in\{1,2,\ldots,L\}}{\text{argmax}} \sum_{j=1}^{T_i} \frac{\mathbf{y}^\top [\widehat{\mathbf{X}}^{(i)}]_{:,j}}{T_i \|\mathbf{y}\|_F \|[\widehat{\mathbf{X}}^{(i)}]_{:,j}\|_F}, \quad i = 1, 2, \ldots, L. \tag{12}$$

The above classification metric captures the average angular proximity of $\text{vec}(\boldsymbol{\mathcal{Y}})$ to the training samples of the i-th class. Under the same approach, other classifiers can also be used, such as SVM, k-NN, or NS. For example, by NS, the classifier assigns $\boldsymbol{\mathcal{Y}}$ to class i^*, if

$$i^\star = \underset{i\in\{1,2,\ldots,L\}}{\text{argmin}} \left\| \mathbf{y} - \mathbf{U}^{(i)}\mathbf{U}^{(i)\top}\mathbf{y} \right\|_F, \tag{13}$$

where $\mathbf{U}^{(i)} = [\Phi(\widehat{\mathbf{X}}^{(i)})]_{:,1:K}$, for some $K \leq \min\{W, T_i\}$.

4.2 Approach 2: Feature Extraction by Joint-Class Tucker2

Originally presented in [10] for standard Tucker decomposition, this approach compresses the training data and extracts their most significant features by joint tensor analysis across all classes. Same as above, we consider $(N+1)$-way tensor $\boldsymbol{\mathcal{X}}^{(i)} \in \mathbb{R}^{D_1\times\ldots\times D_N\times T_i}$ containing the T_i training samples that are available from class i. Then, we concatenate $\boldsymbol{\mathcal{X}}^{(1)}, \ldots, \boldsymbol{\mathcal{X}}^{(L)}$ across their $(N+1)$-th mode to define the all-class training tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1\times\ldots\times D_N\times T}$, where $T = \sum_{i=1}^{L} T_i$. Next, we carry out L1-Tucker2 decomposition of $\boldsymbol{\mathcal{X}}$ to obtain orthonormal factors $\{\mathbf{Q}_n\}_{n=1}^{N}$ and core tensor $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{d_1\times\ldots\times d_N\times T}$. The mode-$(N+1)$ slabs of $\boldsymbol{\mathcal{G}}$ (i.e., N-way tensors that derive by fixing the mode-$(N+1)$ index of $\boldsymbol{\mathcal{G}}$) are the reduced-size features extracted from the original training samples. These slabs are then vectorized and collated to form $\mathbf{G} = [\boldsymbol{\mathcal{G}}]^{\top}_{(N+1)} \in \mathbb{R}^{w\times T}$, where $w = \prod_{n=1}^{N} d_n$. Finally, $\mathbf{G}$ is given as input to any standard classifier (e.g., k-NN, SVM, NS, ANN, or other). During testing, the corresponding w features are first extracted from testing sample $\boldsymbol{\mathcal{Y}}$ as $\mathbf{y} = \text{vec}(\boldsymbol{\mathcal{Y}} \times_{n\in[N]} \mathbf{Q}_n^{\top}) \in \mathbb{R}^{w}$. Then, the class of $\mathbf{y}$ is detected based on the above trained classifier.

4.3 Approach 3: Feature Extraction of Vectorized Samples by Joint-Class L1-Tucker2

In this approach, originally presented in [28] for Tucker, all tensor samples are first vectorized and organized in a 3-way tensor $\boldsymbol{\mathcal{Z}} \in \mathbb{R}^{W\times T\times L}$, the three modes of which are feature index, sample index, and class index. Then, L1-Tucker2 decomposition is applied on $\boldsymbol{\mathcal{Z}}$ and returns orthonormal-basis factors $\mathbf{Q}_1 \in \mathbb{R}^{W\times\bar{w}}$ and $\mathbf{Q}_2 \in \mathbb{R}^{T\times t}$, where $\bar{w} \leq W$ and $t \leq T$, respectively. Accordingly, the reduced-size core tensor $\boldsymbol{\mathcal{G}}$ is extracted $\boldsymbol{\mathcal{G}} = \boldsymbol{\mathcal{Z}} \times_1 \mathbf{Q}_1^{\top} \times_2 \mathbf{Q}_2^{\top}$. Intuitively, $\mathbf{G}_l = [\boldsymbol{\mathcal{G}}]_{:,:,l} \in \mathbb{R}^{\bar{w}\times t}$ can be viewed as a reduced-size representation of the training data ensemble from class l. Across mode 1, the number of features is reduced. Across mode 2, the number of samples is reduced by high-variance linear combinations. Finally, any standard vector-based classifier (SVM, k-NN, ANN, NS, or other) can be trained on $\{\mathbf{G}_l\}_{l=1}^{L}$, and applied/tested on the pertinent features $\widehat{\mathbf{y}} = \mathbf{Q}_1^{\top}\text{vec}(\boldsymbol{\mathcal{Y}})$ of new/unlabeled sample $\boldsymbol{\mathcal{Y}}$. In the specific version of this method that was proposed in [28], an NS classifier was employed.

4.4 Approach 4: L1-Tucker-Based Unsupervised Event Detection

In many applications, a measurement comprises a low-rank component that is present in all samples and, possibly, an additional "event" component drawn from a different subspace. This event could be, for example, a subpixel target in hyperspectral images [40], a signal presence in the sensed radio-frequency spectrum [41], a network anomaly [42, 43], or a foreground object in video analytics [44, 45, 46, 47]. Often, one would be willing to identify whether a particular sample carries this additional component or not –which in essence constitutes a binary classification problem. A generic model in vector processing would be $\mathbf{y} = \mathbf{b} + \alpha\mathbf{s} + \mathbf{n} \in \mathbb{R}^{D}$, where $\mathbf{b}$ is the component drawn from coherent low-rank subspace $\mathcal{Q} = \text{span}(\mathbf{Q})$, for orthonormal basis $\mathbf{Q} \in \mathbb{R}^{D\times d}$, $\mathbf{s}$ is the foreground signal (e.g., target of foreground object in a video frame), with significant presence in the complement of $\mathcal{Q}$, $\alpha \in \{0, 1\}$ is the coefficient that determines signal absence (when 0) and signal presence (when 1), and $\mathbf{n}$ typically constitutes benign zero-mean additive white (Gaussian) noise. Indeed, if $\mathbf{Q}$ is available, one can determine whether $\mathbf{y}$ carries $\mathbf{s}$ ($\alpha = 1$) or not ($\alpha = 0$), based on the value of $f(\mathbf{y}; \mathbf{Q}) = \|(\mathbf{I}_D - \mathbf{Q}\mathbf{Q}^{\top})\mathbf{y}\|_2^2$ compared to an ad-hoc threshold τ –the binary classification would yield $\alpha = 1$ when $f(\mathbf{y}; \mathbf{Q}) \geq \tau$ and $\alpha = 0$, otherwise. In an unsupervised way (i.e., with no training data with labeled α), one commonly estimates $\mathbf{Q}$ by SVD of a collection of unlabled measurements, or by some low-rank-plus-sparse analysis [48, 49]. Similar processing applies to N-way tensor samples [50, 51, 52]. Specifically, we consider unlabeled collected samples

$$\boldsymbol{\mathcal{Y}}_t = \boldsymbol{\mathcal{B}}_t + \alpha_t\boldsymbol{\mathcal{S}}_t + \boldsymbol{\mathcal{N}}_t, \ t = 1, 2, \ldots, T, \tag{14}$$

where we assume that $\boldsymbol{\mathcal{B}}_t$ has a low-Tucker-rank structure $\boldsymbol{\mathcal{B}}_t = \boldsymbol{\mathcal{Z}}_t \times_{n\in[N]} \mathbf{Q}_n^\top$, for some $\boldsymbol{\mathcal{Z}}_t \in \mathbb{R}^{d_1\times d_2\times\ldots\times d_N}$ and $\mathbf{Q}_n \in \mathbb{R}^{D_n\times d_n}$ with $\mathbf{Q}_n^\top\mathbf{Q}_n = \mathbf{I}_{d_n}$, and that $\boldsymbol{\mathcal{S}}_t$ deviates significantly from that multilinear subspace. Also, we assume that only few of the T samples carry $\boldsymbol{\mathcal{S}}_t$. Then, one approximates $\{\widehat{\mathbf{Q}}_n\}_{n=1}^N$ by L1-Tucker2 on the $(N+1)$-way tensor $\boldsymbol{\mathcal{Y}}$, which is constructed by collating all samples $\{\boldsymbol{\mathcal{Y}}_t\}_{t=1}^T$ across its $(N+1)$-th mode. Finally, similar to the vector-processing scheme, for every t, the classifier decides that α_t is 1 if and only if

$$\left\|\boldsymbol{\mathcal{Y}}_t \times_{n\in[N]} (\mathbf{I}_{D_n} - \widehat{\mathbf{Q}}_n\widehat{\mathbf{Q}}_n^\top)\right\|_F^2 \geq \tau, \tag{15}$$

for some ad-hoc threshold $\tau > 0$.

5. NUMERICAL STUDIES

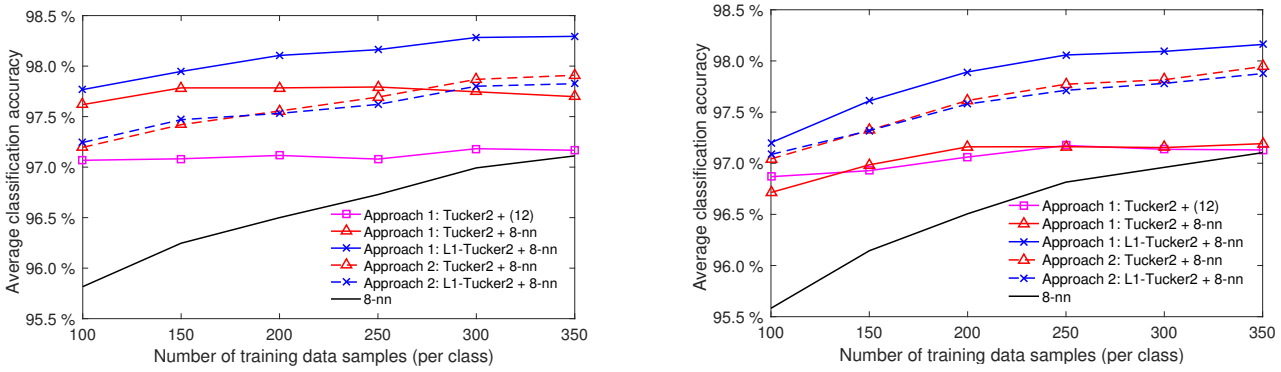
5.1 Numerical Study 1: Classification of Handwritten Digits with Approach 1 and Approach 2

Similar to [53, 54], we consider a binary classification problem of handwritten digits. That is, we consider 28 by 28 images of handwritten digits '1' and '7' of the popular MNIST dataset [55]. We consider availability of $T_1 = T_2 = T$ training and 200 testing samples per class. Data samples are chosen arbitrarily from the available training/testing data samples of the MNIST dataset with no overlap between training and testing data. We examine the classification accuracy performance of the approaches presented in Section 4.1 and Section 4.2. In this study, tensor analysis (Tucker2 and L1-Tucker2) is followed a by $(k = 8)$-NN classifier. We let $T = \{100, 150, \ldots, 350\}$ and compute the classification accuracy of the k-NN classifier and the angular-proximity classifier of [28] (see Section 4.1). As a benchmark, we also compute the accuracy attained by a plain $(k = 8)-$NN classifier applied on the vectorized samples with no tensor analysis. We repeat the experiment over 400 independent selections of training/testing samples and, in Fig. 2a, we plot the average classification accuracy versus T. We observe that L1-Tucker2 of Approach 1 (per class processing) followed by k-NN attains the best performance across the board. Tucker2 of Approach 1 attains similar but lower performance. The plain k-NN classifier starts from a lower accuracy value and its performance improves as the number of training samples increases. L1-Tucker2 and Tucker2 of Approach 2 (joint-class processing) attain almost equal performance across the board.

Next, we repeat the above experiment while considering that 8 data samples of digit '1' class are mistakenly labeled as digit '7' samples and 8 digit '7' samples are mistakenly labeled as digit '1' samples. We plot the classification accuracy versus number of training samples in presence of mislabeling corruption in Fig. 2b. L1-Tucker2 of Approach 1 exhibits remarkable resistance compared to Tucker2 of Approach 1, which has clearly suffered performance loss due to the corrupted (mislabeled) training samples. Interestingly, the performances of L1-Tucker2 and Tucker2 of Approach 2, appear to have remained the same. The plain k-NN classifier exhibits the lowest performance across the board.

5.2 Numerical Study 2: Image-based Object Classification with Approach 2

In this experiment, we operate on the Columbia University Image Library (COIL-20) dataset [56] which consists of 1440 images of objects from 20 classes (72 images per class). First, we select and operate on 5 out of the 20 objects in COIL-20, i.e., 'object 1', 'object 2', 'object 6', 'object 11', and 'object 19'. We down-sample each image to a size of 45 by 45 pixels and consider availability of 40 training samples, and 32 unknown/testing samples per class. In order to simulate outliers in the data, we arbitrarily choose 4 samples from class 'object 2' and 4 samples from class 'object 6' and replace them by arbitrarily chosen images from class 'object 5' (which does not participate in the classification experiment at hand). Moreover, we consider that ρ% of the training image-samples of each class are corrupted by salt-&-pepper noise with density 0.4 (40% of the total number of pixels in an image are set to a pixel value of either 255–'salt' or 0–'pepper'). We let the sample corruption ratio ρ vary in $\{10, 15, \ldots, 50\}$% and compute the classification accuracy attained by the classifier presented in 4.2 (joint-class L1-Tucker2 (or Tucker2) feature extraction followed by SVM) with L1-Tucker2 parameters $d_1 = d_2 = 5$. We repeat this experiment over 1000 statistically independent realizations of training/testing samples selection and salt-&-pepper corruption, and plot the average classification accuracy attained by both



(a) (b)

Figure 2: Numerical Study 1: Average classification of handwritten digits versus number of data samples when (a) operating on nominal data and (b) when data are corrupted by mislabeling.

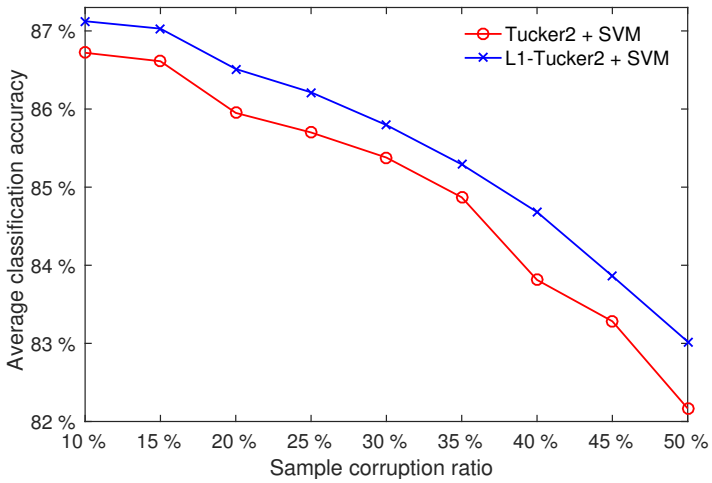
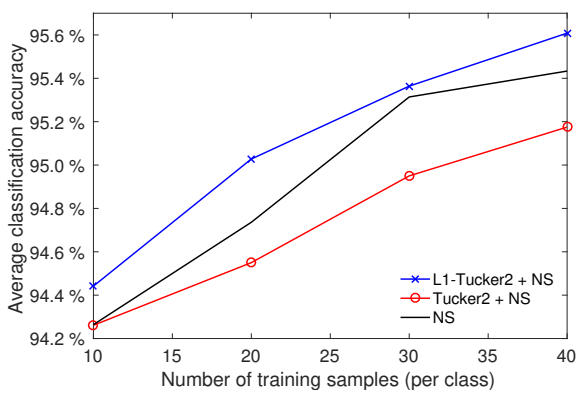


Figure 3: Numerical study 2: Average classification accuracy versus sample corruption ratio. Tucker2 parameters: $d_1 = d_2 = 5$, 40 training and 32 testing samples per class.

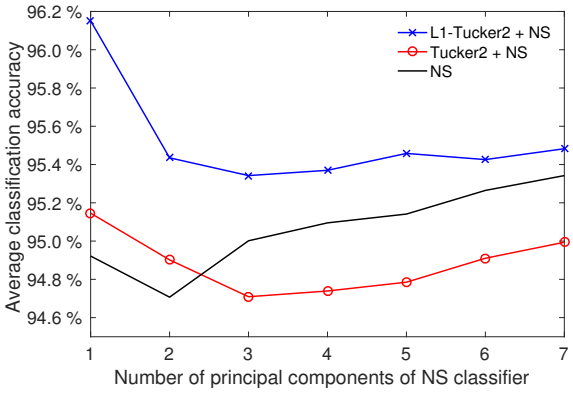
L1-Tucker2 and Tucker2 in Fig. 3. We observe that L1-Tucker2 attains higher average classification accuracy across increasing sample corruption ratio illustrating the robustness of L1-Tucker2 compared to Tucker2. Next, for the same study, we compute the average classification accuracy attained by a plain SVM classifier (SVM on the vectorized raw images) which exhibits average classification accuracies 83.64% and 82.93% for $\rho = 10\%$ and $\rho = 50\%$, respectively. Finally, for comparison purposes, we compute the classification accuracy of plain SVM, joint-class L1-Tucker2 feature extraction followed by SVM, and joint-class Tucker2 feature extraction followed by SVM when all training samples are nominal (i.e., $\rho = 0$ and no training samples are replaced by samples from classes that do not participate in this experiment). The respective average classification accuracies attained are 99.33%, 99.56%, and 99.68%.

5.3 Numerical Study 3: Classification of Text Documents with Approach 3

We continue our studies with a classification task of text documents. Specifically, we work with the top 30 categories TDT2 corpus (Nist Topic Detection and Tracking corpus) dataset which is available online [57] and contains 9394 text documents from 30 classes. Each document comes in the form of a length-36771 numerical vector resulting from text-to-vector transformation approaches of [58, 59, 60]. Our experimental setup follows. For every class we randomly pick T samples for training and T_{test} samples for testing. The classification approach



(a) L1/-Tucker2 parameters: $\bar{w} = 200$, $t = 8$. $k = 6$ components for NS classifier.



(b) L1/-Tucker2 parameters: $\bar{w} = 200$, $t = 8$. $T = 30$ training samples per class.

Figure 4: Numerical Study 2: Classification of text documents: average classification accuracy versus (a) number of training samples (per class) and (b) number of principal components of NS classifier.

followed in this study is the approach described in Section 4.3; i.e., the training data tensor is a 3-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{36771 \times T \times 30}$. The parameters for Tucker2 are $\bar{w} < 36771$ and $t < T$. Tucker2 decomposition returns tensor $\widehat{\boldsymbol{\mathcal{X}}} \in \mathbb{R}^{\bar{w} \times t \times 30}$ and a NS classifier is employed. That is, we compute 30 SVDs –one for each class– and retain the most dominant k singular vectors of each slab. Every unknown sample is classified in accordance with 4.3. First, we fix $\bar{w} = 200$, $t = 8$, $k = 6$, and $T_{\text{test}} = 12$. We let T vary in $\{10, 20, 30, 40\}$ and compute the classification accuracy attained by both Tucker2 and L1-Tucker2. As a benchmark, we also include the performance of a plain NS classifier with $k = 6$ components. We repeat this study across 100 realizations of training/testing data. At each realization there is no overlap between training and testing data. In Fig. 4a, we plot the average classification accuracy versus number of training data samples (per class). We observe that L1-Tucker2 followed by NS attains the the highest performance across the board. The performance of the plain NS classifier follows. Tucker2 followed by NS exhibits comparable but lower performance than the plain NS classifier. Thereafter, we fix the number of training samples per class to $T = 30$ and let the number of principal components of the NS classifier k vary in $\{1, 2, \ldots, 7\}$. We compute the average classification accuracy (over 350 independent realizations of training/testing samples) versus k and illustrate the corresponding classification accuracy curves in Fig. 4b. Once again, L1-Tucker2 feature extraction followed by NS attains the highest accuracy across the board. For $k \leq 2$ Tucker2 followed by NS outperforms the plain NS classifier. Interestingly, L1-Tucker2 followed by NS attains the best performance for $k = 1$ component.

5.4 Numerical Study 4: Foreground Detection in Video Sequences

In this study, we operate on $T = 150$ frames of a selected video from the popular CAVIAR database [61]. Each frame is of size $(D_1 = 202)$-by-$(D_2 = 269)$ pixels. 130 of these frames include a *moving* foreground component and 20 are background-only. We organize the video frames in a 3-way tensor $\boldsymbol{\mathcal{Y}} \in \mathbb{R}^{D_1 \times D_2 \times T}$ and perform L1-Tucker2 (and Tucker2) on $\boldsymbol{\mathcal{Y}}$ to obtain bases $\widehat{\mathbf{Q}}_1 \in \mathbb{R}^{D1 \times d1}$ and $\widehat{\mathbf{Q}}_2 \in \mathbb{R}^{D2 \times d2}$. In accordance with the approach presented in Section 4.4, we classify each frame $\mathbf{Y}_i = [\boldsymbol{\mathcal{Y}}]_{:,:,i}$, $i \in \{1, 2, \ldots, 150\}$, as a background-only frame, if

$$\left\| (\mathbf{I}_{D_1} - \widehat{\mathbf{Q}}_1 \widehat{\mathbf{Q}}_1^\top) \mathbf{Y}_i (\mathbf{I}_{D_2} - \widehat{\mathbf{Q}}_2 \widehat{\mathbf{Q}}_2^\top) \right\|_F \leq \tau, \tag{16}$$

for some detection threshold τ taking values in $\mathcal{T} = \{0, t, 2t, \ldots, A\}$, for maximum value $A = \max_{j \in \{1,2,\ldots,150\}} \|\mathbf{Y}_j\|_F$ and step $t = \frac{A}{1000}$. If the inequality in (16) does not hold true for some frame $\mathbf{Y}_j$, then a foreground signal component is detected in that frame. Next, we denote by $\mathcal{B} \subseteq \{1, 2, \ldots, 150\}$ the set of all indices that correspond to frames that were classified as background-only (BG), and accordingly, we denote by $\mathcal{F} = \{1, 2, \ldots, 150\} \setminus \mathcal{B}$ the set of indices corresponding to frames in which a foreground component was detected (BG+FG). Then, we estimate the video background as $\widehat{\mathbf{B}} = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \mathbf{Y}_i$. In view of the background estimate, for every frame in

which a foreground component was detected, we estimate the foreground component as $\widehat{\mathbf{F}}_i = \mathbf{Y}_i - \widehat{\mathbf{B}},\ i \in \mathcal{F}$. It is important to note that, in this application, it is crucial that the classifier maintains low (ideally 0) false positives (FPs) –i.e., BG+FG frames detected as BG– in order to attain satisfactory background estimation. At the same time, a high number of true positives (TPs) further increases performance.

In our first study, we vary the detection threshold τ in $\mathcal{T}$ and identify a value τ_{best} that (i) maximizes the number of TPs (i.e., correctly detected BG frames), while (ii) restricts the number of FPs to 0. For Tucker2, the optimal threshold is $\tau_{\text{best}} = 532t$, attaining 18 TPs. For L1-Tucker2, the optimal threshold is $\tau_{\text{best}} = 550t$, attaining 20 TPs (i.e., all 20 BG frames are correctly identified). In Fig. 5a, we show the 50-th frame of the video sequence. In Figs. 5b-5e, we show the estimated background and the foreground of the 50-th frame, as computed by means of Tucker2 and L1-Tucker for the optimal thresholds τ_{best} (that yield FP equal to 0). The results are visually satisfying.

Then, we increase the detection thresholds and plot in Fig. 6a how the true positives and false positives vary for the two methods. We observe, for example, that for $\tau = \tau_{\text{best}} + 25t$, L1-Tucker2 still attains TP equal to 20, while there is a small increase of FP to 1 (i.e., one BG+FG frame was detected as BG). On the other hand, for the same threshold increase, Tucker2 is significantly affected, attaining FP equal to 11. The correspondingly estimated background/foregrounds for $\tau = \tau_{\text{best}} + 25t$ are shown in Figs. 5f-5i, where the robustness of L1-Tucker2 is clearly illustrated.

To offer an even broader view of the performance of the two methods, in Fig. 6b, we plot the receiver operating characteristic (ROC) curve, computed for τ varying in $\mathcal{T}$. We observe that HOSVD attains probability of detection (PD) equal to 0.9, for probability of false alarm (PFA) equal to 0. Moreover, HOSVD attains PD equal to 1, for PFA 0.1. On the other hand, L1-HOSVD attains PD equal to 1 for PFA equal to 0.

6. CONCLUSIONS

In this work, we presented a selection of successful Tucker tensor decomposition-based approaches for classification of multi-way data and combined them with L1-Tucker, a corruption resistant L1-norm reformulation of standard Tucker. Numerical studies on multiple real-world datasets corroborate the corruption resistance and classification accuracy afforded by L1-Tucker.

REFERENCES

[1] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn. (Springer)*, vol. 20, n. 3, pp. 273-297, 1995.

[2] T. M. Cover and P. E. Hart, "Nearest neighbor pattern classification," *IEEE Trans. Inf. Theory*, vol. 13, n. 1, pp. 21-27, 1967.

[3] K. P. Murphy, "Naive bayes classifiers," *University of British Columbia*, Technical report [Online]. Available: https://www.cs.ubc.ca/~murphyk/Teaching/CS340-Fall06/reading/NB.pdf.

[4] L. Breiman, "Random forests," *Mach. Learn. (Springer)*, vol. 45, n. 1, pp. 5-32, 2001.

[5] K.-C. Lee, J. Ho, and J. D. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, n. 5, pp. 684-698, 2005.

[6] S. Watanable and N. Pakvasa, "Subspace method of pattern recognition", in *Proc. Int. Joint. Conf. Pattern, Recogn.*, Washington, D.C., 1973, pp. 25-32.

[7] X. Glorot, A. Bordes, and Y. Bengio, "Domain adaptation for large-scale sentiment classification: A deep learning approach," in *Proc. Int. Conf. Mach. Learn. (ICML 2011)*, Bellevue, WA, Jun. 2011, pp. 513-520.

[8] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. Cambridge, MA: MIT press, 2016.

[9] A. Anandkumar, R. Ge, D. Hsu, S. M. Kakade, M. Sham, and M. Telgarsky, "Tensor decompositions for learning latent variable models," *J. Mach. Learn. Res.*, vol. 15, n. 1, pp. 2773-2832, 2014.

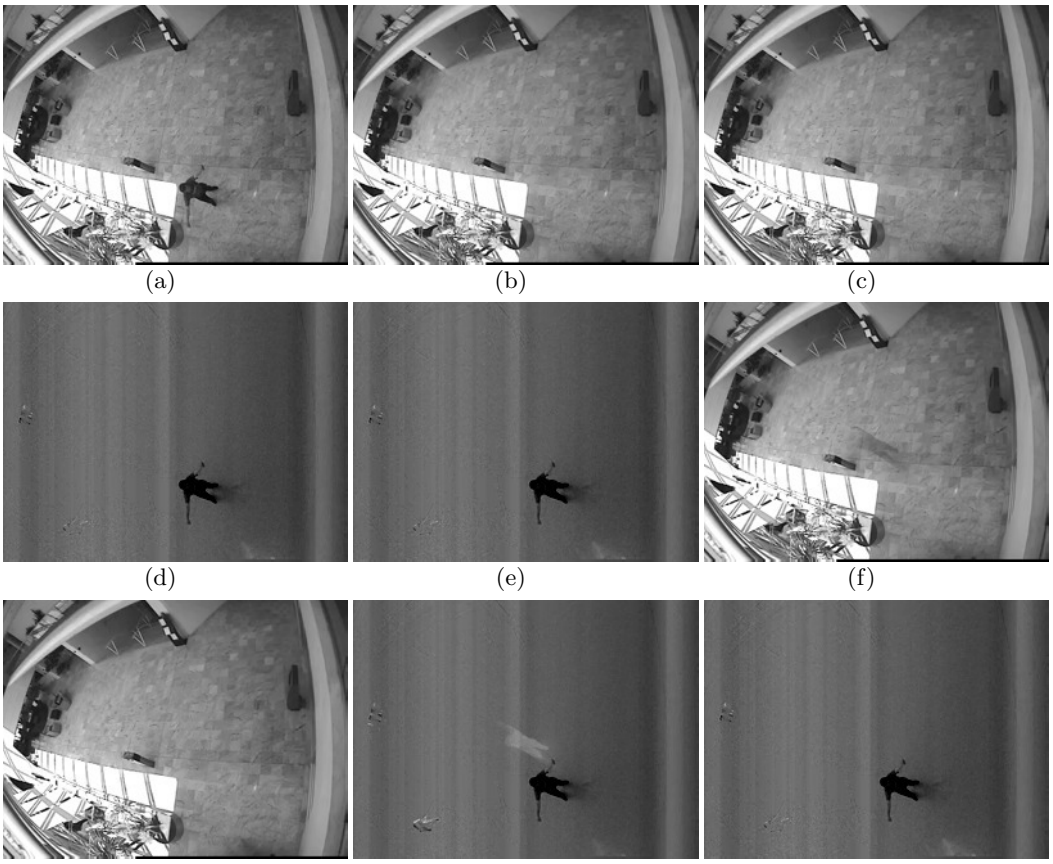
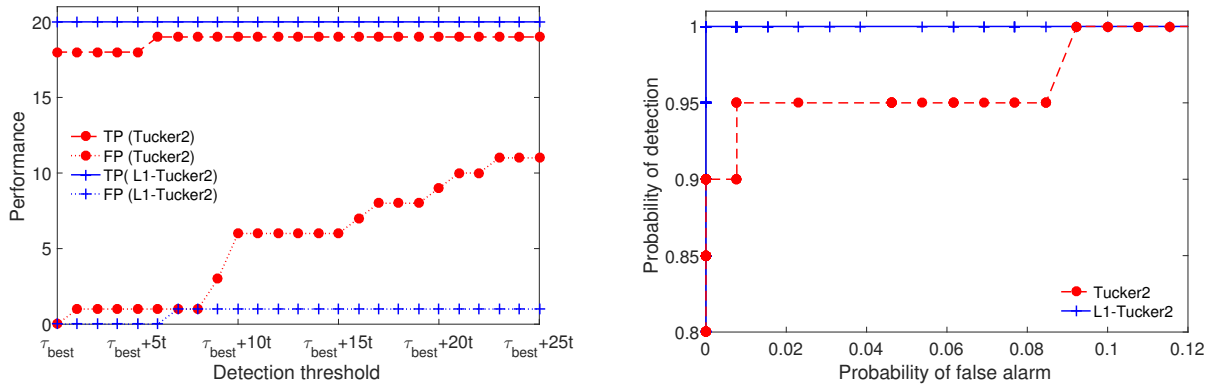


Figure 5: (a) 50-th frame of the processed video. Background extracted for optimal thresholds τ_{best} by (b) Tucker2 and (c) L1-Tucker2. Foreground extracted at τ_{best} by (d) Tucker2 and (e) L1-Tucker2. Background extracted at $\tau_{\text{best}} + 25t$ by (f) Tucker2 and (g) L1-Tucker2. Foreground extracted at $\tau_{\text{best}} + 25t$ by (h) Tucker2 and (i) L1-Tucker2.



(a) (b)

Figure 6: (a) TP/FP performance versus detection threshold and (b) ROC curves for the Tucker2-based and L1-Tucker2-based detectors.

[10] A. H. Phan and A. Cichocki, "Tensor decompositions for feature extraction and classification of high dimensional datasets," *IEICE Nonlinear Theory Appl.*, vol. 1, no. 1, pp. 37-68, 2010.

[11] L. R. Tucker, "Some mathematical notes on three-mode factor analysis," *Psychometrika*, vol. 31, n. 3, pp. 279-311, 1966.

[12] F. L. Hitchcock, "The expression of a tensor or a polyadic as a sum of products," *J. Mathematics and Physics*, vol. 6, n. 1-4, pp. 164-189, 1927.

[13] J. D. Carroll and J.-J. Chang, "Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition," *Psychometrika*, vol. 35, n. 3, pp. 283-319, 1970.

[14] 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, n. 23, pp. 6315-6328, 2015.

[15] P. P. Markopoulos, D. G. Chachlakis, and E. E. Papalexakis, "The exact solution to rank-1 L1-norm TUCKER2 decomposition," *IEEE Signal Process. Lett.*, vol. 25, n. 4, pp. 511-515, 2018.

[16] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, n. 3, pp. 455-500, 2009.

[17] L. De Lathauwer, B. De Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253-1278, 2000.

[18] P. P. Markopoulos, D. G. Chachlakis, and A. Prater-Bennette, "L1-norm higher-order singular-value decomposition," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP 2018)*, Anaheim, CA, Nov. 2018, pp. 1353-1357.

[19] Y. Pang, X. Li, and Y. Yuan, "Robust tensor analysis with L1-norm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, n. 2, pp. 172-178, 2010.

[20] D. G. Chachlakis and P. P. Markopoulos, "Novel algorithms for exact and efficient L1-norm-based TUCKER2 decomposition," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process. (IEEE ICASSP 2018)*, Calgary, Canada, Apr. 2018, pp. 6294-6298.

[21] D. G. Chachlakis and P. P. Markopoulos, "Robust decomposition of 3-way tensors based on L1-norm," in *Proc. SPIE Defense and Commercial Sens. (SPIE DCS 2018)*, Orlando, FL, Apr. 2018., pp. 1065807:1-1065807:15.

[22] 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.

[23] F. Nie, H. Huang, C. Ding, D. Luo, and H. Wang, "Robust principal component analysis with non-greedy l1-norm maximization," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI 2011)*, Barcelona, Spain, Jul. 2011, pp. 1433-1438.

[24] 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, n. 16, pp. 4252-4264, Aug. 2017.

[25] P. P. Markopoulos, M. Dhanaraj, and A. Savakis, "Adaptive L1-norm principal-component analysis with online outlier rejection," *IEEE J. Sel. Topics Signal Process. (IEEE JSTSP)*, vol. 12, no. 6, pp. 1131-1143, 2018.

[26] N. Tsagkarakis, P. P. Markopoulos, G. Sklivanitis, and D. A. Pados, "L1-norm principal-component analysis of complex data," *IEEE Trans. Signal Process.*, vol. 66, pp. 3256-3267, Jun. 2018.

[27] J. C. Gower and G. B. Dijksterhuis, *Procrustes Problems.* Oxford, UK: Oxford University Press, 2004.

[28] B. Savas and L. Eldén, "Handwritten digit classification using higher order singular value decomposition," *J. Pattern Recogn.*, vol. 40, pp. 993-1003, 2007.

[29] E. Newman, M. Kilmer, and L. Horesh, "Image classification using local tensor singular value decompositions," in *Proc. IEEE Int. Workshop Comput. Adv. Multi-Sensor Adapt. Process. (IEEE CAMSAP 2017)*, Curacao, Netherlands Antilles, Dec. 2017, pp. 1-5.

[30] M. E. Kilmer and C. D. Martin, "Factorization strategies for third-order tensors," *Linear Algebra Appl. (Elsevier)*, pp. 641-658, 2011.

[31] I. Kisil, A. Moniri, and D. P. Mandic, "Tensor Ensemble Learning for Multidimensional Data," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP 2018)*, Anaheim, CA, Nov. 2018, pp. 1358-1362.

[32] N. Renard and S. Bourennane, "Dimensionality reduction based on tensor modeling for classification methods," *IEEE Trans. Geoscience Remote Sens.*, vol. 47, no. 4, pp. 1123-1131, 2009.

[33] D. Li, X. Wang, and D. Deguang, "Deeprebirth: Accelerating deep neural network execution on mobile devices," in *Proc. Conf. Artificial Intell.*, New Orleans, LA, Feb. 2018, pp. 2322-2330.

[34] A. Prater, "Classification via tensor decompositions of echo state networks," in *Proc. IEEE Symp. Series Comput. Intell.*, Honolulu, HI, Dec. 2017, pp. 1-8.

[35] I. Kotsia and I. Patras, "Support tucker machines," in *Proc. IEEE Computer Society Conf. Comput. Vision Pattern Recogn. (IEEE CVPR 2011)*, Colorado Springs, CO, Jun. 2011, pp. 633-640.

[36] S. Yan, D. Xu, Q. Yang, L. Zhang, X. Tang, and H.-Z. Zhang, "Multilinear discriminant analysis for face recognition," *IEEE Trans. Image Process.*, vol. 16, no. 1, pp. 212-220, 2007.

[37] Y.-D. Kim, E. Park, S. Yoo, T. Choi, L. Yang, and D. Shin, "Compression of deep convolutional neural networks for fast and low power mobile applications," *arXiv preprint arXiv:1511.06530v2*, 2016.

[38] D. T. Tran, M. Gabbouj, and A. Iosifidis, "Multilinear class-specific discriminant analysis," *Pattern Recogn. Lett. (Elsevier)*, vol. 100, pp. 131-136, 2017.

[39] Q. Li and D. Schonfeld, "Multilinear discriminant analysis for higher-order tensor data classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 36, no. 12, pp. 2524-2537, 2014.

[40] S. Matteoli, M. Diani, G. Corsini, "A tutorial overview of anomaly detection in hyperspectral images," *IEEE A&E Systems Mag.*, vol. 25, no. 7, pp. 5-27, Jul. 2010.

[41] C. G. Tsinos and K. Berberidis, "Spectrum sensing in multi-antenna cognitive radio systems via distributed subspace tracking techniques," in *Handbook of Cognitive Radio*, Springer, Singapore, 2017.

[42] D. Brauckhoff, K. Salamatian, and M. May, "Applying PCA for traffic anomaly detection: problems and solutions," in *Proc. IEEE Int. Conf. Comput. Commun. (IEEE INFOCOM 2009)*, Rio de Janeiro, Brazil, Apr. 2009, pp. 2866-2870.

[43] K. Xie, X. Li, X. Wang, J. Cao, G. Xie, J. Wen, D. Zhang, and Z. Qin, "On-line anomaly detection with high accuracy," *IEEE/ACM Trans. Netw.*, vol. 26, no. 3, pp. 1222-1235, Jun. 2018.

[44] N. Vaswani, T. Bouwmans, S. Javed, and P. Narayanamurthy, "Robust subspace learning: Robust PCA, robust subspace tracking, and robust subspace recovery," in *IEEE Signal Process. Mag.*, vol. 35, no. 4, pp. 32-55, 2018.

[45] T. Bouwmans, N. Aybat, E. Zahzah. *Handbook of robust low-rank and sparse matrix decomposition: Applications in image and video processing.* Boca Raton, FL: CRC press, 2016.

[46] Y. Liu, and D. A. Pados, "Compressed-sensed-domain L1-PCA Video Surveillance," *IEEE Trans. Multimedia*, vol. 18, pp. 351-363, Mar. 2016.

[47] M. Pierantozzi, Y. Liu, D. A. Pados, and S. Colonnese, "Video background tracking and foreground extraction via L1-subspace updates," in *Proc. SPIE Commercial Scientific Sens. Imag.*, Baltimore, MD, Apr. 2016, pp. 985708:1–985708:16.

[48] H. Xu, C. Caramanis, and S. Sanghavi, "Robust PCA via outlier pursuit", in *Proc. Adv. Neural Inf. Process. Syst. (NIPS 2010)*, Vancouver, Canada, Dec. 2010, pp. 2496-2504.

[49] G. Mateos and G. Giannakis, "Robust PCA as bilinear decomposition with outlier-sparsity regularization," *IEEE Trans. Signal Process.*, vol. 60, no. 10, pp. 5176-5190, Oct. 2012.

[50] X. Geng, L. Ji, Y. Zhao, and F. Wang, "A small target detection method for the hyperspectral image based on Higher Order Singular Value Decomposition (HOSVD)," *IEEE Geoscience and Remote Sens. Lett.*, vol. 10, no. 6, pp. 1305-1308, Nov. 2013.

[51] H. Fanaee-T and J. Gama, "Tensor-based anomaly detection: An interdisciplinary survey," *Knowl.-Based Syst. (Elsevier)*, vol. 98, pp. 130-147, 2016.

[52] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms", in *SIAM J. Matrix Anal. Appl.*, vol. 35, n. 1, pp. 225-253, 2014.

[53] B. Biggio, B. Nelso, and P. Laskov, "Poisoning attacks against support vector machines", in *Proc. Int. Conf. Mach. Learn. (ICML 2012)*, Edinburgh, Scotland, Jun. 2012, pp. 1467-1474.

[54] J. Steinhardt, P. W. W. Koh, and P. S Liang, "Certified defenses for data poisoning attacks", in *Proc. Adv. Neural Inf. Process. Syst. (NIPS 2017)*, Long Beach, CA, Dec. 2017, pp. 3517-3529.

[55] The MNIST database of handwritten digits [Online]. Available: http://yann.lecun.com/exdb/mnist/.

[56] Columbia University Image Library (COIL-20) [Online]. Available: http://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php.

[57] Text datasets in matlab format [Online]. Available: http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html.

[58] C. Deng, W. Xuanhui, and H. Xiaofei, "Probabilistic Dyadic Data Analysis with Local and Global Consistency", in *Proc. Int. Conf. Mach. Learn. (ICML 2009)*, Montreal, Canada, Jun. 2009, pp. 105-112.

[59] C. Deng, M. Qiaozhu, H. Jiawei, and Z. Chengxiang, "Modeling Hidden Topics on Document Manifold", in *Proc. Conf. Inf. Knowl. Management (ACM CIKM 2008)*, Napa Valley, CA, Oct. 2008, pp. 911-920.

[60] C. Deng, H. Xiaofei, V. Wei, and H. Jiawei, "Regularized Locality Preserving Indexing via Spectral Regression", in *Proc. Conf. Inf. Knowl. Management (ACM CIKM 2007)*, Lisbon, Portugal, Nov. 2007, pp. 741-750.

[61] Context Aware Vision using Image-based Active Recognition (CAVIAR) [Online]. Available: http://homepages.inf.ed.ac.uk/rbf/CAVIAR/.