Improved L1-Tucker via L1-Fitting

Mahsa Mozaffari[1], Panos P. Markopoulos[1,*], and Ashley Prater-Bennette[2]

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

[2]Air Force Research Laboratory, Information Directorate, Rome, NY 13441, USA
E-mail: ashley.prater-bennette@us.af.mil

***Abstract*—Tucker decomposition is the generalization of Principal Component Analysis to high-order tensors. The L2-norm-based formulation of standard Tucker suffers from severe sensitivity to outliers. L1-norm-based Tucker (L1-Tucker) has been proposed as an outlier-resistant alternative with documented success in an array of applications. L1-HOSVD is a straightforward solver for L1-Tucker that computes each basis by conducting L1-PCA on the matrix that is derived by flattening the data tensor across the corresponding mode. In this paper, L1-HOSVD is further enhanced via an additional L1-fitting step. The proposed method combines the outlier-resistance of L1-HOSVD with that of L1-fitting and returns jointly-computed bases of enhanced robustness, as corroborated by our numerical studies on synthetic and real-world datasets.**

***Index Terms*—Tensors, Tucker, L1-norm, outliers, robustness.**

I. Introduction

Tensors are high-order arrays that naturally represent multi-way data in many real-world applications. Accordingly, tensor analysis methods are becoming increasingly popular in data science, machine learning, and signal processing. In these areas, tensor method applications include factor analysis, anomaly detection, missing data estimation, compression, compressive sensing, denoising, data fusion, co-clustering, and feature extraction for classification [1–8], to name a few. More recently, tensor methods have been successfully used for parameter compression in deep neural networks [9–11].

Tucker tensor decomposition [12, 13] is a standard generalization of Principal Component Analysis (PCA) to tensors and a tensor method of choice for problems including multilinear subspace analysis, dimensionality reduction, denoising, parameter estimation, fusion, and compression [14–18], among others. Higher Order Singular Value Decomposition (HOSVD) [12] and Higher Order Orthogonal Iterations (HOOI) [13] are the two most common algorithms for Tucker decomposition. Despite their success in processing nominal data, HOSVD and HOOI are known to be sensitive against data corruptions in the form of highly deviating entries, fibers, or slabs [19, 20]. The reason can be traced to the L2-norm formulation of Tucker decomposition, that quadratically emphasizes peripheral points. Similar to L1-norm based PCA alternatives [21–23], to remedy the impact of outliers, L1-norm-based Tucker (L1-Tucker) variants have been proposed in the literature, such as L1-HOSVD and L1-HOOI [20, 24–26]. These alternatives have already demonstrated marked robustness, for similar computational effort. L1-HOSVD computes each basis individually by performing L1-PCA [27] on the matrix that is derived by flattening the data tensor across the corresponding mode [20]. L1-HOOI initializes at L1-HOSVD and then jointly updates the bases in an iterative fashion. Both L1-HOSVD and L1-HOOI achieve similar performance as their standard L2-norm counterparts when applied on clean data, while they are much more robust when applied to corrupted data. Specifically, L1-HOOI performs better for moderate levels of outlier corruption, while L1-HOSVD is preferred for higher corruption levels.

This paper introduces a novel method for multi-linear subspace estimation, based on L1-Tucker decomposition, improved by a single, L1-fitting step. The proposed method, named L1T-L1F (which stands for "L1-Tucker followed by L1-Fitting") computes the bases jointly. Moreover, the L1-fitting step allows for a marked improvement in robustness, as corroborated by the presented numerical studies.

II. Preliminaries of Tucker Decomposition

Tensors are multi-way arrays. Vectors and matrices are 1-way and 2-way tensors, respectively. Each entry of an N-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1 \times D_2 \times \cdots \times D_N}$ is referenced by N indices $\{i_n\}_{n=1}^N$, where $i_n \in [D_n] := \{1, 2, \ldots, D_n\}$ $\forall n$. Analogous to matrix rows and columns, for any $n \in [N]$, mode-n fibers of a tensor are identified by fixing all the indexes in $\{i_m\}_{m \in [N] \setminus n}$. A tensor can be matricized by arranging all its mode-n fibers as columns of a matrix. The mode-n matricization (or "flattening") of $\boldsymbol{\mathcal{X}}$ is denoted by $\boldsymbol{\mathcal{X}}_{(n)} \in \mathbb{R}^{D_n \times \prod_{i \in [N] \setminus n} D_i}$. The order of fiber arrangements follows standard conventions [13]. The mode-n product of tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1 \times D_2 \times \cdots \times D_N}$ with matrix $\mathbf{U} \in \mathbb{R}^{r_n \times D_n}$, expressed as $\boldsymbol{\mathcal{Y}} = \boldsymbol{\mathcal{X}} \times_n \mathbf{U}$, is a tensor of size $D_1 \times D_2 \cdots \times r_n \cdots \times D_N$, such that $\boldsymbol{\mathcal{Y}}_{(n)} = \mathbf{U}\, \boldsymbol{\mathcal{X}}_{(n)}$. In shorthand notation, $\boldsymbol{\mathcal{X}} \times_{n \in [N]} \mathbf{U}_n$ is equal to $\boldsymbol{\mathcal{X}} \times_1 \mathbf{U}_1 \cdots \times_N \mathbf{U}_N$. Note that the order in which the mode-n products are applied does not matter [13].

Tucker tensor decomposition low-rank approximates a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1 \times D_2 \cdots \times D_N}$ as $\hat{\boldsymbol{\mathcal{X}}} \approx \boldsymbol{\mathcal{G}} \times_{n \in [N]} \mathbf{U}_n$, where $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{d_1 \times d_2 \cdots \times d_N}$ is the core tensor, $\{\mathbf{U}_i\}_{i \in [N]} \in \mathbb{S}(D_i, d_i)$ are the factor matrices (or bases), and $d_n < D_n$ is the (low) Tucker rank for mode n. Here $\mathbb{S}(D, d) = \{\mathbf{U} \in \mathbb{R}^{D \times d}; \mathbf{U}^\top \mathbf{U} = \mathbf{I}_d\}$ is the Stiefel manifold containing all orthonormal bases in

*Corresponding author.

This material is based upon work supported by the National Science Foundation under award OAC-1808582 and by the Air Force Office of Scientific Research under awards FA9550-20-1-0039 and 18RICOR029.

Cleared for public release by the Air Force Research Laboratory on 02 March 2021, case number AFRL-2021-0655.

$\mathbb{R}^{D\times d}$. In essence, $\mathcal{G}$ is the compressed version of $\mathcal{X}$. In its standard formulation, Tucker decomposition sets $\mathcal{G} = \mathcal{X} \times_{n\in[N]} \mathbf{U}_n^\top$ and chooses bases $\{\mathbf{U}_n\}_{n\in[N]}$ such that the L2-norm of the core (sum of its squared entries) is maximized:

$$\max_{\{\mathbf{U}_i\in\mathbb{S}(D_i,d_i)\}_{i\in[N]}} \left\| \mathcal{X} \times_{i\in[N]} \mathbf{U}_i^\top \right\|_F^2. \tag{1}$$

In general, the exact solution to (1) is not known, but HOSVD [12] and HOOI [13] are two standard approximate solvers.

Extending the L2-norm formulation of PCA to tensors, Tucker inherits its outlier sensitivity, by quadratically benefiting peripheral entries. To counteract this outlier sensitivity of Tucker, researchers have extended L1-PCA to L1-Tucker, for tensor processing [20]. L1-Tucker reformulates (1), seeking the bases that maximize instead the L1-norm of the core (sum of its absolute entries), $\left\| \mathcal{X} \times_{i\in[N]} \mathbf{U}_i^\top \right\|_1$. Analogous to HOSVD and HOOI, L1-HOSVD and L1-HOOI [20], respectively, are the two state-of-the-art algorithms for solving L1-Tucker. L1-HOSVD computes $\mathbf{U}_n$ individually by performing L1-PCA [27] of the mode-n flattening of the data, $\mathcal{X}_{(n)}$. L1-HOOI initializes at the L1-HOSVD solution and iteratively updates $\mathbf{U}_n$ by L1-PCA of the mode-n flattening of the projected data $[\mathcal{X} \times_{m\neq n} \mathbf{U}_m^\top]_{(n)}$. In all cases, L1-PCA can be solved exactly [27] or approximately [28], with low computational cost.

In a multitude of numerical studies, both L1-HOSVD and L1-HOOI have exhibited similar performance to that of their standard L2-norm counterparts when applied on clean data. At the same time, the L1-norm methods are much more robust than standard, L2-norm HOSVD and HOOI, when applied to corrupted data. Specifically, L1-HOOI has been seen to perform better for moderate levels of outlier corruption, while L1-HOSVD is preferred for higher corruption levels.

III. Proposed Method

This work proposes *L1T-L1F*: a novel algorithm for multilinear subspace estimation that improves the performance of L1-Tucker (implemented simply by L1-HOSVD) by supplementing it with an L1-norm Fitting step [29]. A step-by-step description of the proposed method follows.

First, the bases are initialized to the solution of L1-HOSVD. That is, for every n,

$$\tilde{\mathbf{U}}_n \leftarrow \texttt{L1-PCA}(\mathbf{X}_n, d_n), \tag{2}$$

where $\mathbf{X}_n = \mathcal{X}_{(n)}$ and $\texttt{L1-PCA}(\mathbf{X}_n, d_n)$ returns an exact or approximate solution to

$$\max_{\mathbf{U}\in\mathbb{S}(D_n,d_n)} \|\mathbf{U}^\top \mathbf{X}_n\|_1. \tag{3}$$

In this work, to maintain a low computational cost, L1-PCA is solved by means of the L1-BF algorithm of [28]. $\{\tilde{\mathbf{U}}_n\}_{n\in[N]}$ is the L1-HOSVD solution and its robustness has been documented. In this work, L1-HOSVD is further enhanced by means of the following L1-Fitting step.

For every $n \in [N]$, taken in increasing order (or even arbitrary), tensor $\mathcal{G}_n \in \mathbb{R}^{d_1\times d_2\times\cdots\times D_n\times\cdots\times d_N}$ is computed by the L1-fitting

$$\mathcal{G}_n = \underset{\mathcal{G}}{\operatorname{argmin}} \left\| \mathcal{X} - \mathcal{G} \times_{m<n} \mathbf{U}_m^* \times_{k>n} \tilde{\mathbf{U}}_k \right\|_1. \tag{4}$$

The mode-n fibers of $\mathcal{G}_n$ are expected to span the sought-after mode-n subspace. Also, they are expected to be nearly outlier-free, as they are computed by means of robust L1-fitting on the robust L1-HOSVD bases.

As a final step, we return the mode-n basis that derives by L1-PCA on the corresponding fitting matrix, as

$$\mathbf{U}_n^* \leftarrow \texttt{L1-PCA}(\mathbf{A}_n, d_n), \tag{5}$$

where $\mathbf{A}_n = [\mathcal{G}_n \times_{m<n} \mathbf{U}_m^* \times_{k>n} \tilde{\mathbf{U}}_k]_{(n)}$.

Note on L1-Fitting: The problem in (4) is a linear regression with L1-norm fitting error. The mode-n flattening of $\mathcal{G}_n$ in (4), $\mathbf{G}_n \in \mathbb{R}^{D_n\times p_n}$, for $p_n = \prod_{m\in[N]\setminus n} d_m$, can be directly found as

$$\mathbf{G}_n = \underset{\mathbf{G}\in\mathbb{R}^{D_n\times p_n}}{\operatorname{argmin}} \left\| \mathbf{W}_n \mathbf{G}^\top - {\mathbf{X}_n}^\top \right\|_1, \tag{6}$$

where $\mathbf{W}_n = (\mathbf{U}_N \otimes \mathbf{U}_{N-1} \otimes \cdots \otimes \mathbf{U}_{n+1} \otimes \mathbf{U}_{n-1} \otimes \cdots \otimes \mathbf{U}_1) \in \mathbb{R}^{P_n\times p_n}$, $P_n = \prod_{m\in[N]\setminus n} D_m$, and $\otimes$ is the Kronecker product. Accordingly, for each $i \in [D_n]$, the i-th row of $\mathbf{G}_n$, $\mathbf{g}_n(i) = [\mathbf{G}_n]_{i,:}^\top$, can be individually found as

$$\mathbf{g}_n(i) = \underset{\mathbf{g}\in\mathbb{R}^{p_n}}{\operatorname{argmin}} \|\mathbf{W}_n \mathbf{g} - \mathbf{x}_n(i)\|_1, \tag{7}$$

where $\mathbf{x}_n(i) = [\mathbf{X}_n]_{i,:}^\top$. It is worth noting that (7) can be rewritten as a standard linear program (LP) of the form

$$\min_{\mathbf{g}\in\mathbb{R}^{p_n};\ \mathbf{t}\in\mathbb{R}^{P_n}} \mathbf{t}^\top \mathbf{1}_{P_n} \tag{8}$$

$$\text{s.t. } -[\mathbf{t}]_j \leq [\mathbf{W}_n]_{j,:}\mathbf{g} - [\mathbf{x}_n(i)]_j \leq [\mathbf{t}]_j \ \forall j \in [P_n] \tag{9}$$

and solved by means of a standard LP solver, such as the simplex algorithm or the interior-point method [30]. This can be solved in polynomial time with respect to D_n and p_n. The problem in (7) can also be rewritten as

$$\min_{\mathbf{g}\in\mathbb{R}^{p_n};\ \mathbf{z}\in\mathbb{R}^{P_n}} \|\mathbf{z}\|_1 \tag{10}$$

$$\text{s.t. } \mathbf{W}_n\mathbf{g} - \mathbf{z} = \mathbf{x}_n(i) \tag{11}$$

and solved by Alternating Direction Method of Multipliers (ADMM) [31].

Note on scalability: The L1 regression problem in (7) is typically over-constrained ($P_n \gg p_n$) and can have prohibitively high computational cost, compared to that of L1-HOSVD/HOOI, as the size of the processed tensor increases. However, the problem is readily amenable to standard cost-reduction techniques for regression, such as sketching and sampling [32–35].

Note on additional iterations: After the bases $\{\mathbf{U}_n\}_{n\in[N]}$ are obtained, one could optionally repeat the steps 2-5 of the algorithm in Fig. 1 T times, in an iterative fashion. Numerical studies have shown that this could further improve the quality of the derived bases.

Algorithm 1. Proposed L1T-L1F algorithm.

Input: Data $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1 \times D_2 \times \cdots \times D_N}$, Subspace ranks $\{d_n\}_{n\in[N]}$.

1: $\tilde{\mathbf{U}}_n \leftarrow \texttt{L1-PCA}(\boldsymbol{\mathcal{X}}_{(n)}, d_n)\ \forall n$.
2: for $n = 1, 2, \ldots, N$:
3: $\quad \boldsymbol{\mathcal{G}}_n \leftarrow \text{argmin}_{\boldsymbol{\mathcal{G}}} \left\| \boldsymbol{\mathcal{X}} - \boldsymbol{\mathcal{G}} \times_{m<n} \mathbf{U}_m^* \times_{k>n} \tilde{\mathbf{U}}_k \right\|_1$.
4: $\quad \mathbf{A}_n = [\boldsymbol{\mathcal{G}}_n \times_{m<n} \mathbf{U}_m^* \times_{k>n} \tilde{\mathbf{U}}_k]_{(n)}$.
5: $\quad \mathbf{U}_n^* \leftarrow \texttt{L1-PCA}(\mathbf{A}_n, d_n)$.

Return: $\{\mathbf{U}_n^*\}_{n\in[N]}$.

Fig. 1. Pseudocode of proposed algorithm L1T-L1F that enhances the L1-HOSVD solution by an L1-norm Fitting step.

IV. Numerical Studies

A. Synthetic Data: Subspace Learning

In this section, the performance of the proposed algorithm L1T-L1F is evaluated on multi-linear subspace estimation, in comparison with HOSVD, HOOI, L1-HOSVD, and L1-HOOI, on synthetic data, in the presence of data corruptions. The model dimensions are set to $D_1 = D_3 = D_5 = 5$, $D_2 = D_4 = 8$, and $d_1 = d_2 = 3$, $d_3 = d_4 = d_5 = 2$. First, a low-rank Tucker-structured tensor is generated as $\boldsymbol{\mathcal{X}} = \boldsymbol{\mathcal{G}} \times_{n\in[5]} \mathbf{U}_n$, where the entries of $\boldsymbol{\mathcal{G}} \in \mathbb{R}^{3\times3\times2\times2\times2}$ are independently drawn from $\mathcal{N}(0, 9)$ and the factor matrices are arbitrarily fixed orthonormal bases $\{\mathbf{U}_n\}_{n\in[5]} \in \mathbb{S}(D_n, d_n)$. The clean tensor is additively corrupted by dense noise and sparse outliers. The resulting available data tensor is $\boldsymbol{\mathcal{X}}_{\text{corr}} = \boldsymbol{\mathcal{X}} + \boldsymbol{\mathcal{N}} + \boldsymbol{\mathcal{O}}$, where $\boldsymbol{\mathcal{N}}$ contains independent entries from $\mathcal{N}(0, 1)$ and $\boldsymbol{\mathcal{O}}$ is a sparse tensor with only $N_o = 10$ non-zero entries, drawn from $\mathcal{N}(0, \sigma_o^2)$ (i.e., outlier frequency of occurrence as low as 0.125%). Using the aforementioned methods, $\{\mathbf{U}_n\}_{n\in[5]}$ is estimated.

To quantify the intensity of corruption, with respect to the added noise, the Outlier-to-Noise Ratio (ONR) is defined as

$$\text{ONR} = \frac{\mathbb{E}\{\|\boldsymbol{\mathcal{O}}\|_F^2\}}{\mathbb{E}\{\|\boldsymbol{\mathcal{N}}\|_F^2\}} = \frac{N_o}{\prod_{n\in[N]} D_n} \frac{\sigma_o^2}{\sigma_n^2}. \tag{12}$$

In this study, σ_o is set to $0, 2, \ldots, 16$, corresponding to ONR values ranging between 0 and 0.32. The subspace-estimation performance is measured by means of the Mean Aggregated Normalized Subspace Squared Error (MANSSE)

$$\text{MANSSE} = \sum_{n\in[N]} \frac{1}{2Nd_n} \left\| \mathbf{U}_n \mathbf{U}_n^\top - \hat{\mathbf{U}}_n \hat{\mathbf{U}}_n^\top \right\|_F^2, \tag{13}$$

where for any $n \in [5]$, $\mathbf{U}_n$ is the sought-after mode-n basis and $\hat{\mathbf{U}}_n$ is the estimated one. Fig. 2 shows the performance curve of MANSSE vs. ONR over 1000 noise/outlier realizations. For ONR ≤ 0.05, HOOI, L1-HOOI, and L1T-L1F exhibit miniscule error. HOSVD and L1-HOSVD follow with somewhat inferior performance. For higher levels of corruption, both HOSVD and HOOI break down. L1-HOSVD maintains similar levels of robustness across the board. L1-HOOI performs better than L1-HOSVD for ONR $= 0.25$ but its performance deteriorate for higher levels of corruption. Quite interestingly, the proposed L1T-L1F maintains outstanding robustness and minimal error for every value of ONR.

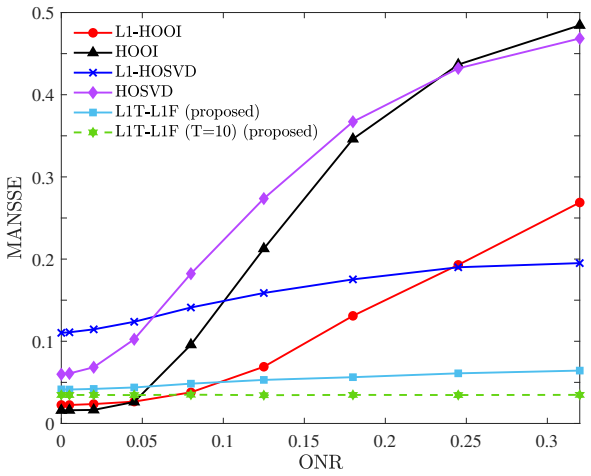

Fig. 2. Numerical experiment on subspace estimation (synthetic data). MANSSE attained by HOSVD, HOOI, L1-HOSVD, L1-HOO1, and the proposed L1T-L1F vs. ONR.

TABLE I
Runtime comparison of the algorithms on the synthetic data.

Algorithm	Runtime (s)
HOSVD	0.0032
L1-HOSVD	0.2212
HOOI	0.3282
L1-HOOI	0.7581
L1T-L1F (proposed)	3.1663

Moreover, we observe that iterating steps 2-5 of the proposed method $T = 10$ times can further improve the performance.

A runtime comparison of the algorithms is shown in Table I. L1 regression in (7) is solved by means of ADMM. We observe that L1T-L1F requires more computational effort compared to its counterparts. However, the computational cost of the regression can be reduced by means of randomization and other standard approximation techniques such as sketching and sampling [32–35].

Another robust method for Tucker is Higher-Order Robust PCA (HORPCA) [36], which seeks to decompose $\boldsymbol{\mathcal{X}}_{\text{corr}}$ into two additive components, a low-rank one $\hat{\boldsymbol{\mathcal{X}}}$ that estimates $\boldsymbol{\mathcal{X}}$ and a sparse one $\hat{\boldsymbol{\mathcal{O}}}$ that estimates $\boldsymbol{\mathcal{O}}$. A tunable parameter λ regularizes emphasis between the low-rank-ness of $\hat{\boldsymbol{\mathcal{X}}}$ and the sparsity of $\hat{\boldsymbol{\mathcal{O}}}$. HORPCA can be solved by means of the HORPCA-S algorithm introduced in [36]. HORPCA can demonstrate outstanding robustness, similar to L1T-L1F for very high SNR and optimally tuned parameter λ. For suboptimal choices of λ and/or lower levels of SNR (when $\boldsymbol{\mathcal{X}}_{\text{corr}}$ does not have a "low-rank plus sparse" structure), the performance of HORPCA significantly drops (the reader is referred to the numerical study in [20]). On the contrary, L1T-L1F suppresses both outliers and noise and its performance does not depend on any tunable parameter.

B. Uber Pickups Dataset: Compression

In this study, Tucker decomposition is used for tensor compression/reconstruction and the performance of L1T-L1F is compared with those of HOSVD, HOOI, L1-HOSVD, and L1-HOOI. Our study is on the "Uber Pickups" dataset [37], which consists of the number of Uber pickups in New York City over six months. The original tensor is a 4-way tensor of size $183 \times 24 \times 1140 \times 1717$, in which modes represent number of days, hours, longitude and latitude, respectively. In this study the tensor is reduced to a 3-way tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{D_1=125 \times D_2=125 \times D_3=183}$, where the first two modes represent location coordinates and third mode represents days. The reduced tensor is obtained by summing the number of Uber pickups in each day over all 24 hours, and undersampling each 1140×1717 slice of the tensor to size 125×125 by summing the entries over blocks.

In this study, the first 14 days of the data, corresponding to $\boldsymbol{\mathcal{X}}_{:,:,1:14}$, is used as training data, denoted by $\boldsymbol{\mathcal{X}}_{\text{tr}} \in \mathbb{R}^{125 \times 125 \times 14}$. The training data is corrupted by additive outliers as $\boldsymbol{\mathcal{X}}_{\text{corr}} = \boldsymbol{\mathcal{X}}_{\text{tr}} + \boldsymbol{\mathcal{O}}$. A single 125×125 slab of $\boldsymbol{\mathcal{X}}_{\text{tr}}$ is arbitrarily selected and entries within it are randomly corrupted with probability β. The corrupted entries within $\boldsymbol{\mathcal{O}}$ are randomly generated non-zero integer numbers generated by $\text{unif}(v/10, v)$, for $v \in V := \{10, 50, 100, 200, 250, 300, 350, 400, 450, 500\}$. The corrupted training data $\boldsymbol{\mathcal{X}}_{\text{corr}}$ is Tucker decomposed along first two modes with factor matrices $\mathbf{U}_1$, and $\mathbf{U}_2 \in \mathbb{S}(125, d)$ computed. The third mode is considered to be the sample mode with the third factor matrix constrained as $\mathbf{U}_3 = \mathbf{I}_{125 \times 125}$. The remaining entries of the collected data, $\boldsymbol{\mathcal{X}}_{:,:,15:183}$, form the test dataset, denoted by $\boldsymbol{\mathcal{X}}_{\text{te}}$, and is used to evaluate the effectiveness of the computed bases by each algorithm in performing compression and reconstruction of unseen data. To that end, the test data is reconstructed using the Tucker factors computed on the training data by $\hat{\boldsymbol{\mathcal{X}}} = \boldsymbol{\mathcal{X}}_{\text{te}} \times_{n \in [2]} \hat{\mathbf{U}}_n \hat{\mathbf{U}}_n^T$. The performance is measured by Mean Normalized Squared Error (MNSE) defined by $\|\hat{\boldsymbol{\mathcal{X}}} - \boldsymbol{\mathcal{X}}_{\text{te}}\|_F^2 \|\boldsymbol{\mathcal{X}}_{\text{te}}\|_F^{-2}$. In this study, the performances are evaluated over 1000 realizations of corruption.

Fig. 3 demonstrates MNSE versus d, for $2 \leq d \leq 10$, and $\beta = 0.2$, and $v = 500$. The figure suggest that as d increases, the error generally decreases for all methods. HOOI and HOSVD, achieve the highest reconstruction errors compared to the L1-based tucker decomposition counterparts. Among the L1-based methods, L1T-L1F attains the lowest reconstruction error for all d values.

Fig. 4 demonstrates MNSE versus the corruption magnitude v, for $d = 4$, $\beta = 0.2$, and $v \in V$. As the corruption magnitude increases, the reconstruction error of all methods increase. However for $v \geq 100$, L1T-L1F shows the most robustness against the corruptions among all the methods, by achieving the lowest error.

V. Conclusion

This work introduced *L1T-L1F*: a novel algorithm for multilinear subspace estimation that improves the performance of L1-Tucker (implemented by L1-HOSVD) by supplementing it with an L1-norm Fitting step. Aiming at robustness against corruptions, the proposed method utilizes L1-fitting for further reducing the effect of outliers in the processed data, and jointly computing the tucker factor matrices. The numerical studies on synthetic and real data, for subspace estimation, and tensor reconstruction, corroborate the effectiveness of L1T-L1F in all the tested levels of data corruption.

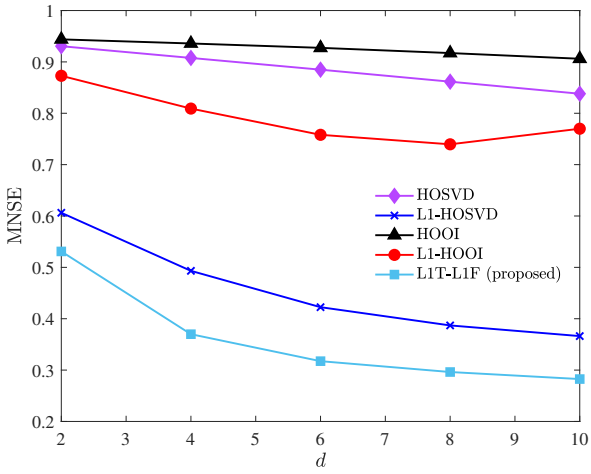

Fig. 3. Numerical Experiment on compression/reconstruction of tensor on Uber Pickup dataset. MNSE attained by HOSVD, L1-HOSVD, HOOI, L1-HOOI, and the proposed L1T-L1F vs. d. The corruption ratio is $\beta = 0.2$, and the corruption magnitude is $v = 500$.

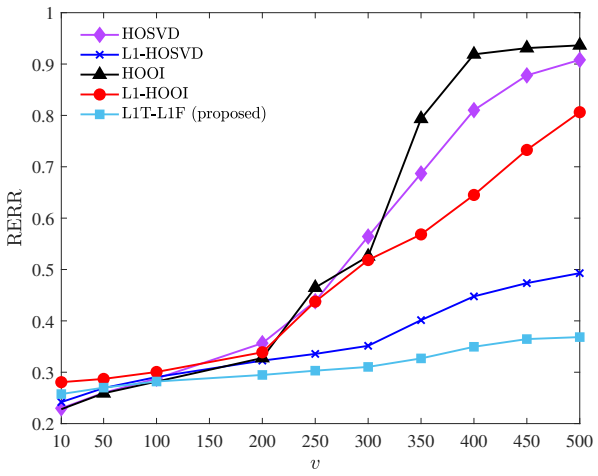

Fig. 4. Numerical experiment on compression/reconstruction of tensor on Uber pickup dataset. MNSE attained by HOSVD, L1-HOSVD, HOOI, L1-HOOI, and the proposed L1T-L1F vs. v (corruption magnitude). The corruption ratio within corrupted slab is $\beta = 0.2$, and $d = 4$.

References

[1] E. E. Papalexakis, C. Faloutsos, and N. D. Sidiropoulos, "Tensors for data mining and data fusion: Models, applications, and scalable algorithms," *ACM Trans. Intell. Syst. Tech.*, vol. 8, no. 2, pp. 1–44, 2016.

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

[3] M. Ashraphijuo and X. Wang, "Union of low-rank tensor spaces: Clustering and completion," *J. Mach. Learn. Res.*, vol. 21, no. 69, pp. 1–36, 2020.

[4] K. Gilman and L. Balzano, "Online tensor completion and free submodule tracking with the T-SVD," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Barcelona, Spain, 2020, pp. 3282–3286.

[5] M. S. Asif and A. Prater-Bennette, "Low-rank tensor ring model for completing missing visual data," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*, Barcelona, Spain, 2020, pp. 5415–5419.

[6] M. A. O. Vasilescu and D. Terzopoulos, "Multilinear analysis of image ensembles: Tensorfaces," in *Proc. Eur. Conf. Comput. Vision (ECCV)*, Copenhagen, Denmark, 2002, pp. 447–460.

[7] X. Fu, S. Ibrahim, H.-T. Wai, C. Gao, and K. Huang, "Block-randomized stochastic proximal gradient for low-rank tensor factorization," *IEEE Trans. Signal Process.*, vol. 68, pp. 2170–2185, 2020.

[8] K. Tountas, D. A. Pados, and M. J. Medley, "Conformity evaluation and L1-norm principal-component analysis of tensor data," in *Proc. SPIE Big Data: Learn. Analytics Appl.*, vol. 10989, Baltimore, MD, 2019, p. 109890P.

[9] A.-H. Phan, K. Sobolev, K. Sozykin, D. Ermilov, J. Gusak, P. Tichavskỳ, V. Glukhov, I. Oseledets, and A. Cichocki, "Stable low-rank tensor decomposition for compression of convolutional neural network," in *Proc. Eur. Conf. Comput. Vision (ECCV)*. Glasgow, UK: Springer, 2020, pp. 522–539.

[10] A. Kolbeinsson, J. Kossaifi, Y. Panagakis, A. Bulat, A. Anandkumar, I. Tzoulaki, and P. Matthews, "Robust deep networks with randomized tensor regression layers," *arXiv preprint arXiv:1902.10758*, 2019.

[11] J. Kossaifi, A. Bulat, G. Tzimiropoulos, and M. Pantic, "T-net: Parametrizing fully convolutional nets with a single high-order tensor," in *Proc. IEEE Conf. Comput. Vision Patt. Recognit. (CVPR)*, Long Beach, CA, 2019, pp. 7822–7831.

[12] 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.

[13] T. G. Kolda and B. W. Bader, "Tensor decompositions and applications," *SIAM Rev.*, vol. 51, no. 3, pp. 455–500, 2009.

[14] Y. Sun, Y. Guo, C. Luo, J. Tropp, and M. Udell, "Low-rank Tucker approximation of a tensor from streaming data," *SIAM J. Math. Data Sci.*, vol. 2, no. 4, pp. 1123–1150, 2020.

[15] M. B. Amin, W. Zirwas, and M. Haardt, "HOSVD-based denoising for improved channel prediction of weak massive MIMO channels," in *Proc. IEEE Veh. Technol. Conf. (VTC)*, Sydney, NSW, Australia, 2017, pp. 1–5.

[16] W. d. C. Freitas, G. Favier, and A. L. de Almeida, "Tensor-based joint channel and symbol estimation for two-way MIMO relaying systems," *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 227–231, 2018.

[17] H. Ben-Younes, R. Cadene, M. Cord, and N. Thome, "MUTAN: Multimodal Tucker fusion for visual question answering," in *Proc. IEEE Int. Conf. Comput. Vision (ICCV)*, Venice, Italy, 2017, pp. 2612–2620.

[18] R. Ballester-Ripoll, P. Lindstrom, and R. Pajarola, "TTHRESH: Tensor compression for multidimensional visual data," *IEEE Tran. Vis. Comput. Graphics*, vol. 26, no. 9, pp. 2891–2903, 2019.

[19] 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, no. 4, pp. 511–515, 2018.

[20] D. G. Chachlakis, A. Prater-Bennette, and P. P. Markopoulos, "L1-norm Tucker tensor decomposition," *IEEE Access*, vol. 7, pp. 178 454–178 465, 2019.

[21] J. P. Brooks and J. H. Dulá, "The L1-norm best-fit hyperplane problem," *Appl. Math. Lett.*, vol. 26, no. 1, pp. 51–55, 2013.

[22] J. P. Brooks, J. H. Dulá, and E. L. Boone, "A pure L1-norm principal component analysis," *Comput. Statist. Data Anal.*, vol. 61, pp. 83–98, 2013.

[23] N. Gillis and S. A. Vavasis, "On the complexity of robust PCA and ℓ_1-norm low-rank matrix approximation," *Math. Oper. Res.*, vol. 43, no. 4, pp. 1072–1084, 2018.

[24] 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)*, Anaheim, CA, 2018, pp. 1353–1357.

[25] D. G. Chachlakis, M. Dhanaraj, A. Prater-Bennette, and P. P. Markopoulos, "Dynamic L1-norm Tucker tensor decomposition," *IEEE J. Selected Topics Signal Process.*, vol. 15, no. 3, pp. 587–602, 2021.

[26] D. G. Chachlakis, A. Prater-Bennette, and P. P. Markopoulos, "L1-norm higher-order orthogonal iterations for robust tensor analysis," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process. (ICASSP)*. Barcelona, Spain: IEEE, 2020, pp. 4826–4830.

[27] P. P. Markopoulos, G. N. Karystinos, and D. A. Pados, "Optimal algorithms for L1-subspace signal processing," *IEEE Tran. Signal Process.*, vol. 62, no. 19, pp. 5046–5058, 2014.

[28] P. P. Markopoulos, S. Kundu, S. Chamadia, and D. A. Pados, "Efficient L1-norm principal-component analysis via bit flipping," *IEEE Tran. Signal Process.*, vol. 65, no. 16, pp. 4252–4264, 2017.

[29] N. Tsagkarakis, P. P. Markopoulos, and D. A. Pados, "On the L1-norm approximation of a matrix by another of lower rank," in *Proc. IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Anaheim, CA, 2016, pp. 768–773.

[30] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[31] S. Boyd, N. Parikh, and E. Chu, *Distributed optimization and statistical learning via the alternating direction method of multipliers*. Now Publishers Inc., 2011.

[32] M. B. Cohen and R. Peng, "Lp row sampling by lewis weights," in *Proc. Annu. ACM SIGACT Symp. Theory Comput. (STOC)*, Portland, OR, 2015, pp. 183–192.

[33] K. L. Clarkson and D. P. Woodruff, "Sketching for m-estimators: A unified approach to robust regression," in *Proc. Annu. ACM-SIAM Symp. Discrete Algorithms (SODA)*. San Diego, CA: SIAM, 2014, pp. 921–939.

[34] ——, "Low-rank approximation and regression in input sparsity time," *J. ACM (JACM)*, vol. 63, no. 6, pp. 1–45, 2017.

[35] D. P. Woodruff, "Sketching as a tool for numerical linear algebra," *arXiv preprint arXiv:1411.4357*, 2014.

[36] D. Goldfarb and Z. Qin, "Robust low-rank tensor recovery: Models and algorithms," *SIAM J. Matrix Anal. Appl.*, vol. 35, no. 1, pp. 225–253, 2014.

[37] S. Smith, J. W. Choi, J. Li, R. Vuduc, J. Park, X. Liu, and G. Karypis, "FROSTT: The formidable repository of open sparse tensors and tools," 2017, [Online]. Available: http://frostt.io/.