SPARSE REGULARIZATION OF TENSOR DECOMPOSITIONS

Hyon-Jung Kim[†,‡]*, Esa Ollila*[†] *and Visa Koivunen*[†]

[†] Aalto University, Dept. of Signal Processing and Acoustics, P.O.Box 13000, FI-00076 Aalto, Finland
[‡] University of Oulu, Dept. of Mathematical Sciences, P.O .Box 3000, FI-90014 Oulun yliopisto Finland

ABSTRACT

Multi-linear techniques using tensor decompositions provide a unifying framework for the high-dimensional data analysis. Sparsity in tensor decompositions clearly improves the analysis and inference of multi-dimensional data. Other than non-negative tensor factorizations, the literature on tensor estimation using sparsity is limited. In this paper, we introduce sparse regularization methods for tensor decompositions which are useful for dimensionality reduction, feature selection as well as signal recovery. One major challenge in most of the tensor decomposition algorithms is their heavy dependence on good initializations. To alleviate such a critical problem we propose a reliable method based on the ridge regression to provide good starting values taking advantage of sparsity. Combined with such initializations our sparse regularization methods show highly improved performance over the conventional methods in the demonstrated simulation studies.

Index Terms— tensors, CANDECOMP, PARAFAC, regularization, sparsity, LASSO

1. INTRODUCTION

Multi-dimensional data is becoming more common and pervasive with advances in computer storage and information management. Tensors accommodating such data as multiway arrays have increased interest in big data and turned our attention to tensor-based scientific computations from the familiar matrix decompositions such as singular value decompositions (SVD) and principal components analysis (PCA). Tensor decompositions of multilinear models like CANDECOMP/PARAFAC (CP) [4, 9] or Tucker models [20] provide a unifying framework for multidimensional data analysis with simplified notations and algebras [12]. While the massive amounts of data often lead to limitations and challenges in analysis, sparsity in tensor decompositions clearly improves the analysis and inference of multi-dimensional data. For example, sparsity can be used for accurate signal recovery (e.g. compressed sensing) [17] or to eliminate unnecessary redundant features (dimensions) of many modern data sets (e.g. financial and consumer data, DNA micro arrays, internet network traffic flows, functional MRIs) allowing simple visualization and exploration of the data [1].

We first address two different notions of sparsity. The first notion of sparsity refers to the case in which the considerable number of data elements are zero or close to zero in their relative magnitude [13]. The second notion appears in the regularization methods such as the ridge regression and LASSO (least absolute shrinkage and selection operator) where the estimated regression parameters are either shrunk towards zero or driven to zero by increasing the penalty of model complexity [18]. Even though two notions of sparsity are used in different settings, they are related to a certain degree in the context of tensor data. The underlying sparsity of the tensor data naturally implies that the factor matrices of a decomposed tensor are sparse as well. Thus, when the tensor data are themselves sparse or the main features and aspects of a high-dimensional tensor data involve some sparse structure, the regularization methods successfully estimate the tensor factors of CP decompositions instead of the usual least squares estimation.

Other than non-negative tensor factorizations in [16, 15, 6], we note that the tensor decompositions using sparsity has been considered rather recently in [10, 5, 17, 2] among others. In this paper, we first propose regularization methods for tensor factor decompositions with the popular CP model. One major challenge in most of the tensor decomposition algorithms is their heavy dependence on good initializations [14]. In order to alleviate such a critical problem we propose a reliable method based on the ridge regression to provide good starting values taking advantage of sparsity. Combined with such initializations our sparse regularization methods show highly improved performance compared to the conventional decomposition algorithms as illustrated in the simulation studies.

Relations to prior work: We note that Lasso regularization in non-negative CP decomposition was proposed for clustering purposes in [8]. Recently, the CP decomposition with sparse factors using Lasso and ALS, called the Sparse CP-ALS method, has also been addressed (but not recommended) in [2]. Although the idea of solving the regularized criterion (8) in the ALS algorithm coincides with ours, the details on the implementation of the method are missing in [2]. Furthermore, the results (using an essentially same simulation setting) depicted in [2, Table I] are not in line with our simulation results, implying significant differences in the imple-

mentation and initializations.

1.1. Notations for Tensors

A tensor $\mathcal{A}$ of order d usually seen as a d-way array with d indices, represents a multi-linear operator with coordinates $\mathcal{A}_{i_1..i_d}$. Tensors are denoted by boldface Euler script letters, e.g. $\mathcal{A}$, matrices by $\mathbf{A}$, vectors by $\mathbf{a}$ and scalars by a. Then $\mathbf{a}_i$ will denote the ith column vector of a matrix $\mathbf{A}$ and $\mathbf{a}_{(i)}$ its (transposed) row vector. Let $\circ$ denote the *outer product*, *i.e.* $\mathbf{a}\circ\mathbf{b} = \mathbf{a}\mathbf{b}^T$ and $\mathbf{a}\circ\mathbf{b}\circ\mathbf{c}$ has $a_ib_jc_k$ as its (i,j,k)th element, $\odot$ denotes the *Khatri-Rao product*, *i.e.* for any matrices $\mathbf{B} \in \mathbb{R}^{J\times R}$ and $\mathbf{C} \in \mathbb{R}^{K\times R}$, $\mathbf{C}\odot\mathbf{B}$ is a $JK \times R$ matrix of the form

$$\mathbf{C}\odot\mathbf{B} = \begin{pmatrix}\mathbf{c}_1\otimes\mathbf{b}_1 & \cdots & \mathbf{c}_R\otimes\mathbf{b}_R\end{pmatrix},$$

where $\otimes$ denotes the Kronecker product; in the vector case, $\mathbf{c}\otimes\mathbf{b} = \mathrm{vec}(\mathbf{b}\mathbf{c}^T)$. Let $\|\cdot\|_2$ (resp. $\|\cdot\|_1$) denote the ℓ_2-norm (resp. ℓ_1-norm) defined as $\|\mathbf{A}\|_2^2 = \mathrm{Tr}(\mathbf{A}\mathbf{A}^T) = \sum_i\sum_j a_{ij}^2$ (resp. $\|\mathbf{A}\|_1 = \sum_i\sum_j |a_{ij}|$) for any matrix $\mathbf{A}$. For simplicity, we first illustrate the methods for 3-way tensors, but the extensions to multiway tensors are trivial.

1.2. The CANDECOMP/PARAFAC (CP) Models

The *CANDECOMP/PARAFAC (CP) decomposition* [4, 9] approximates a tensor $\mathcal{X} \in \mathbb{R}^{I\times J\times K}$ by a predicted tensor $\hat{\mathcal{X}}$ consisting of a sum of $R \in \mathbb{N}^+$ rank-1 tensors (outer products):

$$\hat{\mathcal{X}} \equiv [\![\boldsymbol{\gamma}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!] \triangleq \sum_{r=1}^{R}\gamma_r\mathbf{a}_r\circ\mathbf{b}_r\circ\mathbf{c}_r.$$

Thus, we model $\mathcal{X}$ as

$$\mathcal{X} = \sum_{r=1}^{R}\gamma_r\mathbf{a}_r\circ\mathbf{b}_r\circ\mathbf{c}_r + \mathcal{E} \tag{1}$$

where $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$ and $\mathbf{c}_r \in \mathbb{R}^K$ for $r = 1,\ldots,R$ form the unit-norm column vectors of the *factor matrices* $\mathbf{A} \in \mathbb{R}^{I\times R}$, $\mathbf{B} \in \mathbb{R}^{J\times R}$, and $\mathbf{C} \in \mathbb{R}^{K\times R}$ and the tensor $\mathcal{E} \in \mathbb{R}^{I\times J\times K}$ contains the *error terms*. Note that the factor matrices are not assumed to be orthonormal and not necessarily of full rank. The model (1) can be expressed in a *matrix* form by unfolding the tensor into a matrix along any of the three modes. Unfolding the tensor $\mathcal{X}$ along the first mode yields a $I \times JK$-matrix denoted as $\mathbf{X}_{(1)}$ so that the equivalent representation of (1) is

$$\mathbf{X}_{(1)} = \mathbf{A}\boldsymbol{\Gamma}(\mathbf{C}\odot\mathbf{B})^T + \mathbf{E}_{(1)}, \tag{2}$$

where $\boldsymbol{\Gamma} = \mathrm{diag}(\boldsymbol{\gamma})$ and $\mathbf{E}_{(1)}$ denotes the unfolded $I \times JK$ matrix of $\mathcal{E}$. The mode-2 and mode-3 unfoldings of the tensor $\mathcal{X}$ are obtained similarly:

$$\mathbf{X}_{(2)} = \mathbf{B}\boldsymbol{\Gamma}(\mathbf{C}\odot\mathbf{A})^T + \mathbf{E}_{(2)},$$
$$\mathbf{X}_{(3)} = \mathbf{C}\boldsymbol{\Gamma}(\mathbf{B}\odot\mathbf{A})^T + \mathbf{E}_{(3)},$$

where $\mathbf{E}_{(2)}$ denotes the unfolded $J \times IK$ matrix of $\mathcal{E}$ and $\mathbf{E}_{(3)}$ denotes the unfolded $K \times IJ$ matrix of $\mathcal{E}$.

Consider the case that $\mathbf{B}$ and $\mathbf{C}$ are fixed and that γ_r's are the scales of the columns of $\mathbf{A}$, i.e., $\mathbf{a}_r$'s are no-longer unit vectors, but $\gamma_r = \|\mathbf{a}_r\|$. Then,

$$\min_{\mathbf{A}}\|\mathcal{X} - [\![\boldsymbol{\gamma}; \mathbf{A}, \mathbf{B}, \mathbf{C}]\!]\|_2 = \min_{\mathbf{A}}\|\mathbf{X}_{(1)} - \mathbf{A}(\mathbf{C}\odot\mathbf{B})^T\|_2^2 \tag{3}$$

of which the least squares (LS) solution is found as

$$\hat{\mathbf{A}} = \mathbf{X}_{(1)}(\mathbf{C}\odot\mathbf{B})((\mathbf{C}^T\mathbf{C}) * (\mathbf{B}^T\mathbf{B}))^\dagger \tag{4}$$

with $\dagger$ denoting the Moore-Penrose inverse. Now the estimates of γ_r's are simply $\hat{\gamma}_r = \|\hat{\mathbf{a}}_r\|$. Assuming that the number of factors R is known, one of the most popular algorithm to compute a CP decomposition is the alternating least squares (ALS) method, i.e. CP-ALS algorithm. The ALS algorithm successively estimates the LS solutions for each component $\mathbf{B}$ and $\mathbf{C}$ in turn keeping others fixed until an appropriate convergence criterion is satisfied.

2. THE SPARSE REGULARIZATION METHODS

The underlying sparsity of a tensor $\mathcal{X}$ leads us to a simple remedy of using the regularization methods instead of solving the conventional LS-criterion as in (3). The aforementioned strong sensitivity to the choice of initial estimates is unavoidable with most of the well-known algorithmic tensor factorization methods including CP-ALS, HOSVD (Higher Order SVD), HOPCA (Higer Order PCA), and HOOI (Higher Order Orthogonal Iteration). To provide good initial estimates in tensor factor estimation we first propose the CP alternating ridge regression (ARR) method with an ℓ_2-penalization. We note that these estimates can be used not only for the good initial guesses but also for stand-alone estimates when the underlying structure of tensor data demands shrinkage with non-zero values instead of sparsity with many zero values. With those good starting values the CP alternating LASSO method is desirable in the sense that the LASSO is computationally feasible for high-dimensional data with the fast LARS (Least Angle Regression) algorithm in practice [18, 7].

2.1. The Alternating Ridge Regression (ARR) method

With the continuing framework of the CP-ALS algorithm, we first let $\mathbf{Z} = \mathbf{C}\odot\mathbf{B}$ for brevity in updating $\hat{\mathbf{A}}$ fixing the others fixed. Then, the minimization in (3) simplifies to $\min_{\mathbf{A}}\|\mathbf{X}_{(1)} - \mathbf{A}\mathbf{Z}^T\|_2^2$. The *ridge regression* [11] (also called *Tikhonov regularization* in the fields of Bayesian estimation and the inverse problems) can be formulated in our context

$$\hat{\mathbf{A}} \equiv \mathrm{RR}(\mathbf{X}_{(1)}, \mathbf{Z}, \lambda) = \arg\min_{\mathbf{A}}\|\mathbf{X}_{(1)} - \mathbf{A}\mathbf{Z}^T\|_2^2 + \lambda\|\mathbf{A}\|_2^2. \tag{5}$$

This approach (commonly formulated in the univariate response regression problems) extends to the multivariate response case in a straightforward manner. The solution to the above optimization problem is easily found to be

$$\hat{\mathbf{A}} = \mathbf{X}_{(1)}\mathbf{Z}(\mathbf{Z}^T\mathbf{Z} + \lambda\mathbf{I})^{-1},$$

where, as in (3), the term $\mathbf{Z}^T\mathbf{Z}$ can be computed efficiently as $\mathbf{Z}^T\mathbf{Z} = (\mathbf{C}^T\mathbf{C}) * (\mathbf{B}^T\mathbf{B})$. In general, the bigger the *ridge (shrinkage) parameter* λ, the greater the amount of shrinkage of coefficients towards zero.

The appropriate choice of the penalty parameter is an open problem in the regularization methods. We suggest two approaches to estimate λ. The first, called the *high level shrinkage (HLS-) estimator*, is defined as $\hat{\lambda} = \frac{1}{R}\sum_{j=1}^{R} d_j^2$, where $d_1 \geq d_2 \geq \cdots \geq d_R \geq 0$ are the singular values of $\mathbf{Z}$. Note that the penalty parameter simplifies further as $\hat{\lambda} = JK$ due to the fact that in the ridge regression the rows of $\mathbf{X}_{(1)}$ are centered and the columns of $\mathbf{Z}$ are centered and standardized.

In the second approach, we estimate the penalty parameter λ using the Bayesian information criteria (BIC) [19]. Since we have a multivariate regression problem, we formulate the BIC criterion as

$$\mathrm{BIC}(\lambda) = N \ln \hat{\sigma}^2 + \mathrm{df}(\lambda) \cdot \ln N \tag{6}$$

where $N = JK$ is the number of columns in $\mathbf{X}_{(1)}$, $\hat{\sigma}^2 = \frac{1}{I}\|\mathbf{X}_{(1)} - \mathbf{A}\mathbf{Z}^T\|_2^2$ is the average squared residuals and $\mathrm{df}(\lambda)$ is the degrees of freedom of the model. $\mathrm{df}(\lambda)$ is defined as $\mathrm{df}(\lambda) = I \cdot \mathrm{Tr}\{\mathbf{H}_\lambda\} = I \cdot \sum_{j=1}^{R} d_j^2(d_j^2 + \lambda)^{-1}$ where $\mathbf{H}_\lambda = \mathbf{Z}(\mathbf{Z}^T\mathbf{Z} + \lambda\mathbf{I})^{-1}\mathbf{Z}^T$ denotes the "hat matrix". Then, the BIC penalty parameter estimate of λ is

$$\hat{\lambda} = \arg\min_{\lambda \in \Lambda_n} \mathrm{BIC}(\lambda) \tag{7}$$

where Λ_n is a grid of n values $\lambda_{n-1} < \lambda_{n-2} < \cdots < \lambda_1 < \lambda_0$. We set $\lambda_0 = 10 \cdot d_1$, i.e., a value sufficient to shrink the parameters close to a zero and $\lambda_i = \epsilon^{i/(n-1)}\lambda_0$ for $i = 1, \ldots, n-1$. The smallest λ value equals $\lambda_{n-1} = \epsilon\lambda_0$ and the larger n one chooses, the finer is the grid. In our simulations, we use $\epsilon = 1.0e^{-4}$ and $n = 100$.

The Alternating Ridge Regression (ARR) method with the HLS estimator of λ is explained in Table 1. When $\hat{\lambda}$ is calculated in steps 2-4 using (7), the ARR method becomes *BIC-ARR method.*

2.2. The CP Alternating LASSO method

To obtain sparse solutions, one may utilize ℓ_1-penalty as in the celebrated LASSO [18] method. When we emphasize the sparse nature of a tensor, we solve for $\mathbf{A}$ using $\ell_2 - \ell_1$ criterion function instead of ℓ_2 criterion in (5):

$$\begin{aligned}\hat{\mathbf{A}} &\equiv \mathrm{LASSO}(\mathbf{X}_{(1)}, \mathbf{Z}, \lambda) \\ &= \arg\min_{\mathbf{A}} \|\mathbf{X}_{(1)} - \mathbf{A}\mathbf{Z}^T\|_2^2 + \lambda\|\mathbf{A}\|_1 \end{aligned} \tag{8}$$

Table 1. The HLS-ARR method.

1. Initialize $\mathbf{B}$ and $\mathbf{C}$ by $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ using the CP-ALS estimates or random matrices.
2. $\hat{\mathbf{A}} = \mathbf{X}_{(1)}\mathbf{Z}(\mathbf{Z}^T\mathbf{Z} + \hat{\lambda}\mathbf{I})^{-1}$, where $\mathbf{Z} = \hat{\mathbf{C}} \odot \hat{\mathbf{B}}$ has standardized columns and $\hat{\lambda} = JK$.
3. $\hat{\mathbf{B}} = \mathbf{X}_{(2)}\mathbf{Z}(\mathbf{Z}^T\mathbf{Z} + \hat{\lambda}\mathbf{I})^{-1}$, where $\mathbf{Z} = \hat{\mathbf{C}} \odot \hat{\mathbf{A}}$ has standardized columns and $\hat{\lambda} = IK$.
4. $\hat{\mathbf{C}} = \mathbf{X}_{(3)}\mathbf{Z}(\mathbf{Z}^T\mathbf{Z} + \hat{\lambda}\mathbf{I})^{-1}$, where $\mathbf{Z} = \hat{\mathbf{B}} \odot \hat{\mathbf{A}}$ has standardized columns and and $\hat{\lambda} = IJ$.
5. Repeat steps 2–4 until the relative change in fit is small.

Table 2. CP Alternating LASSO method.

1. Initialize $\mathbf{B}$ and $\mathbf{C}$ by $\hat{\mathbf{B}}$ and $\hat{\mathbf{C}}$ using the strategy below.
2. Set $\mathbf{X}_{(1)}$ and $\mathbf{Z} = \hat{\mathbf{C}} \odot \hat{\mathbf{B}}$. Compute $\hat{\mathbf{A}}^T = (\hat{\mathbf{a}}_{(1)} \cdots \hat{\mathbf{a}}_{(I)})$ as $\hat{\mathbf{a}}_{(i)} = \mathrm{LASSO}(\mathbf{x}_i, \mathbf{Z}, \hat{\lambda})$, where $\hat{\lambda}$ is chosen using BIC method for $i = 1, \ldots, I$.
3. Set $\mathbf{X}_{(2)}$ and $\mathbf{Z} = \hat{\mathbf{C}} \odot \hat{\mathbf{A}}$. Compute $\hat{\mathbf{B}}^T = (\hat{\mathbf{b}}_{(1)} \cdots \hat{\mathbf{b}}_{(J)})$ as $\hat{\mathbf{b}}_{(i)} = \mathrm{LASSO}(\mathbf{x}_i, \mathbf{Z}, \hat{\lambda})$, where $\hat{\lambda}$ is chosen using BIC method for $i = 1, \ldots, J$.
4. Set $\mathbf{X}_{(3)}$ and $\mathbf{Z} = \hat{\mathbf{B}} \odot \hat{\mathbf{A}}$. Compute $\hat{\mathbf{C}}^T = (\hat{\mathbf{c}}_{(1)} \cdots \hat{\mathbf{c}}_{(K)})$ as $\hat{\mathbf{c}}_{(i)} = \mathrm{LASSO}(\mathbf{x}_i, \mathbf{Z}, \hat{\lambda})$, where $\hat{\lambda}$ is chosen using BIC method for $i = 1, \ldots, K$.
5. Repeat steps 2–4 until the relative change in fit is small.

The optimization problem decomposes to $i = 1, \ldots, I$ separate LASSO estimation problems and $\hat{\mathbf{A}}^T = (\hat{\mathbf{a}}_{(1)} \cdots \hat{\mathbf{a}}_{(I)})$ in (8) can be found by solving $\hat{\mathbf{a}}_{(i)} = \mathrm{LASSO}(\mathbf{x}_i, \mathbf{Z}, \lambda)$ for $i = 1, \ldots, I$. In the above, the ith (transposed) row vector of $\mathbf{X}_{(1)}$, denoted by $\mathbf{x}_i \in \mathbb{R}^{JK}$, plays the role of the 'response variable' regressed by the R columns ('explanatory variables') of $\mathbf{Z} = \hat{\mathbf{C}} \odot \hat{\mathbf{B}}$.

For the choice of the penalty parameter λ, we utilize the BIC criterion (6), where N is now the number of elements in $\mathbf{x}_i$, $\hat{\sigma}^2 = \frac{1}{N}\|\mathbf{x}_i - \mathbf{Z}\hat{\mathbf{a}}_{(i)}^{\lambda}\|_2^2$ and $\mathrm{df}(\lambda)$ is the number of non-zero estimated parameters in the obtained LASSO estimate $\hat{\mathbf{a}}_{(i)}^{\lambda}$. Then, the penalty parameter estimate is chosen as in (7) with the same grid, but here λ_0 is the penalty parameter that shrinks sufficiently all the parameters to zero. We set $\epsilon = 1.0e^{-4}$ but choose $n = 30$ instead of $n = 100$ as in BIC-ARR (i.e., less dense grid) to reduce the computation time for practical purposes. Note that a larger n would increase the optimality of the LASSO fit and this value can be increased when the long computational time is not an issue.

The proposed method is explained in Table 2. The initialization strategy is as follows. If BIC-ARR estimator $\hat{\mathcal{X}}_2$ provided a better fit than the CP-ALS estimator $\hat{\mathcal{X}}_1$ in the sense that $\|\mathcal{X} - \hat{\mathcal{X}}_1\|_2 > \|\mathcal{X} - \hat{\mathcal{X}}_2\|_2 + 0.01\|\mathcal{X}\|_2$, then initialize the algorithm using the BIC-ARR estimator $\hat{\mathcal{X}}_2$, otherwise initialize using the HLS-ARR estimator $\hat{\mathcal{X}}_2$. Thus, if the BIC-ARR estimator does not provide sufficiently better initialization estimates than the ALS estimates it is safer to use the estimates with the guaranteed sparse-nature.

3. SIMULATIONS

We consider model (1), when the factor matrices and hence the true noise-free three-way tensor $\mathcal{X}_0$ is sparse. The observed three-way tensor is generated as $\mathcal{X} = \mathcal{X}_0 + \mathcal{E}$, where $\mathcal{X}_0 = \sum_{r=1}^{R} \lambda_r \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r$ is the Kruskal tensor, $\mathcal{E}$ is the noise tensor and the rank R is assumed to be known. The accuracy of the obtained estimate $\hat{\mathcal{X}}$ is calculated by the relative mean squared error $\mathrm{RMSE}(\hat{\mathcal{X}}) = \|\mathcal{X}_0 - \hat{\mathcal{X}}\|_2^2 / \|\mathcal{X}_0\|_2^2$. When a factor matrix, say $\mathbf{A} \in \mathbb{R}^{I \times R}$, is sparse, then it is of interest to measure the classification performance of the method, or in other words, the performance for correctly estimating zero/non-zero features. Such information is conveniently summarized via the 2×2 *confusion matrix* of the following form:

		Estimate of $\mathbf{A}$		
		0	$\neq 0$	sum
True $\mathbf{A}$	0	n_{1C}	n_{1M}	n_1
	$\neq 0$	n_{2M}	n_{2C}	n_2
	sum	n_1'	n_2'	$I \cdot R$

where n_{1C} (resp. n_{2C}) is the number of entries in the estimate $\hat{\mathbf{A}}$ correctly "classified" as being zero (resp. non-zero) and n_{1M} (resp. n_{2M}) is the number of entries in $\hat{\mathbf{A}}$ "misclassified" as being non-zero (resp. zero). Then, $I \cdot R = n_1 + n_2 = n_1' + n_2'$ is the total number of entries in matrix $\mathbf{A} \in \mathbb{R}^{I \times R}$. To obtain a single measure of the accuracy, we can summarize the results further by the *classification error rate*, defined as $\mathrm{CER}(\hat{\mathbf{A}}) = (n_{1M} + n_{2M})/(I \cdot R)$ or *recovery rate* $\mathrm{RER}(\hat{\mathbf{A}}) = 1 - \mathrm{CER}(\hat{\mathbf{A}})$.

In our first simulation setting $I = 1000, J = 20$ and $K = 20$, the noise tensor $\mathcal{E} \in \mathbb{R}^{1000 \times 20 \times 20}$ has independent elements from $N(0,1)$ distributions, the Kruskal tensor $\mathcal{X}_0$ has rank $R = 3$, and only the factor matrix $\mathbf{A} \in \mathbb{R}^{1000 \times 3}$ is sparse. The factor matrix $\mathbf{A}$ is generated in a way that each element A_{ij} is either equal to a zero or an independent random deviate from $N(0,1)$ with equal probability 1/2. The entries of non-sparse factor matrices $\mathbf{B} \in \mathbb{R}^{20 \times 3}$ and $\mathbf{C} \in \mathbb{R}^{20 \times 3}$ are independent deviates from $N(0,1)$ distribution. The columns of $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ are then normalized to have unit length and the values of the loadings are $\gamma_1 = 1000, \gamma_2 = 500$ and $\gamma_3 = 500$. We simulated $M = 50$ tensors according to the above setup.

The *sparsity factor* (SF), defined as the average number (based on M Monte Carlo trials) of zero elements in $\mathcal{X}_0$ is $\mathrm{SF} = 12.6\%$ and the *signal to noise ratio (SNR)*, defined as the average value of $\|\mathcal{X}_0\|^2 / \|\tilde{\mathcal{E}}\|^2$ is $\mathrm{SNR} = 4.2894$. The estimated noise tensor $\tilde{\mathcal{E}}$ keeps the same sparse structure of $\mathcal{X}_0$. The average RMSE of the CP-ALS and the CP alternating LASSO were, 0.0652 (0.0814) and 0.0088 (0.0218) respectively, with the standard deviations of each in the parenthesis. The average confusion matrices for estimating the sparse factor matrix $\mathbf{A} \in \mathbb{R}^{1000 \times 3}$ were

$$\begin{pmatrix} 0 & 1507 \\ 0 & 1493 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 1290 & 216 \\ 83 & 1411 \end{pmatrix}$$

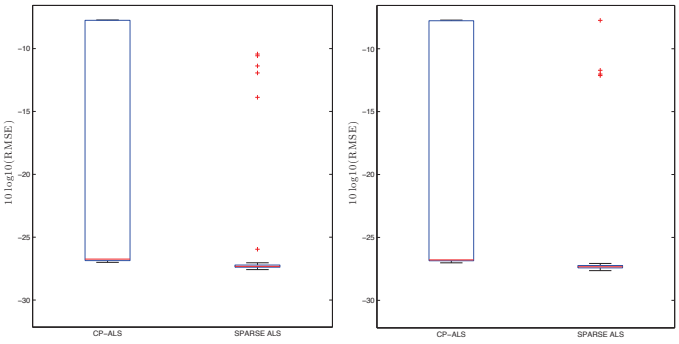

Fig. 1. Boxplots of the $10\log_{10}(\mathrm{RMSE})$ values for the simulation setting 1 (left plot) and setting 2 (right plot) for the ALS and the Sparse ALS (CP alternating LASSO) methods

in the case of CP-ALS method and the CP Alternating LASSO method, respectively. As expected the conventional CP-ALS method does not set any of the elements of $\hat{\mathbf{A}}$ to zero. On the contrary, the proposed CP Alternating LASSO method has the rate of $1290/1507 \approx 85.6\%$ of correctly selecting zero features and its overall recovery rate is $\mathrm{RER}(\hat{\mathbf{A}}) = (1290 + 141)/3000 = 90.3\%$. Naturally, both CER and RER are about 50% and 50% for the ALS method indicating that the method serves as a random guess classifier. The first two boxplots in Figure 1 with $10\log_{10}(\mathrm{RMSE})$ of the obtained values for the CP-ALS and CP Alternating LASSO method display the considerably improved accuracy of our proposed sparse method.

In the second simulation setting, two factor matrices $\mathbf{B}$ and $\mathbf{C}$ are sparse and generated as the above, i.e. each element is either equal to a zero or or an independent random deviate from $N(0,1)$ with equal probability 1/2. Other parameters remain the same as before and $M = 50$ tensors were simulated according to the above model. The sparsity factor is much higher in this simulation setting, reaching the level of $\mathrm{SF} = 68\%$ on average. The average RMSE of the ALS and the sparse-ALS method were, 0.0486 (0.0753) and 0.0089 (0.0274) respectively, where the values in the parentheses are the standard deviations. The average confusion matrices for estimating the sparse factor matrices $\mathbf{A} \in \mathbb{R}^{1000 \times 3}$, $\mathbf{B} \in \mathbb{R}^{20 \times 3}$ and $\mathbf{C} \in \mathbb{R}^{2 \times 3}$ were

$$\begin{pmatrix} 1306 & 197 \\ 88 & 1409 \end{pmatrix}, \quad \begin{pmatrix} 28 & 2 \\ 1 & 29 \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 28 & 2 \\ 1 & 29 \end{pmatrix}$$

for the CP alternating LASSO method. Thus, RER values were 91%, 95% and 95% for correctly estimating zero/non-zero features of the factor matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$, respectively. As in the first simulation the conventional CP-ALS method performed as a random guess classifier. The third and fourth boxplots of $10\log_{10}(\mathrm{RMSE})$ illustrate that the sparse-ALS method offers highly more accurate estimates by exploiting the knowledge of sparsity.

4. REFERENCES

[1] G. I. Allen, "Sparse higher-order principal components analysis," in *Proc. 15th International Conference on Artificial Intelligence and Statistics (AISTATS)*, La Palma, Canary Islands, Apr 21–23 2012.

[2] G. I. Allen, "Regularized tensor factorizations and higher-order principal components analysis," Tech. Rep. TR2012-01, Rice University, 2012.

[3] B. W. Bader, T. G. Kolda, MATLAB tensor toolbox version 2.4, *http://csmr.ca.sandia.gov/ tgkolda/TensorToolbox/*

[4] 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, pp. 283–319, 1970.

[5] E. C. Chi and T. G. Kolda "On Tensors, Sparsity, and Nonnegative Factorizations, SIAM Journal on Matrix Analysis and Applications, to appear.

[6] A. Cichocki, R. Zdunek, A. Phan, and S. Amari, Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation. *Wiley*, 2009.

[7] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "least angle regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–451, 2004.

[8] N. D. Sidiropoulos and E. E. Papalexakis, "Co-clustering as multilinear decomposition with sparse latent factors," *ICASSP 2013: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2064 – 2067, Prague, Czech Republic, May 22 - 27, 2011.

[9] R. A. Harshman, "Foundations of the PARAFAC procedure: models and conditions for an explanatory multi-modal factor analysis," *UCLA working papers in phonetics*, vol. 16, pp. 1–84, 1970.

[10] T. Hazan, S. Polak, and A. Shashua, "Sparse image coding using a 3d non-negative tensor factorization, *ICCV*, Volume 1, pp. 50-57, 2005.

[11] A. E. Hoerl and R. W. Kennard, "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, pp. 55–67, 1970.

[12] T. G. Kolda, and B. W. Bader, "Tensor Decompositions and Applications, *SIAM Review*, 51(3), pp. 455-500, 2009.

[13] T. G. Kolda, and J. Sun, "Scalable Tensor Decompositions for Multi-aspect Data Mining, *ICDM 2008: Proceedings of the 8th IEEE International Conference on Data Mining,* pp. 363-372, 2008.

[14] P. Kroonenberg, Applied multiway data analysis, Wiley Online Library, Volume 702. 2008.

[15] J. Liu, P. Wonka, and J. Ye, "Sparse non-negative tensor factorization using columnwise coordinate descent, *Pattern Recognition* 45 (1), pp. 649-656, 2012.

[16] M. Morup, L. K. Hansen, S. M. Arnfred, " Algorithms for Sparse Non-negative TUCKER, *Neural Computation*, vol. 20(8), pp. 2112–2131, 2008

[17] N. D. Sidiropoulos, and A. Kyrillidis, "Multi-way compressed sensing for sparse low-rank tensors, *IEEE Signal Processing Letters,* 19(11) pp.757–760, 2012.

[18] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Royal Stat. Soc., Ser. B*, vol. 58, pp. 267–288, 1996.

[19] P. Stoica and Y. Selen, "Model-order selection: a review of information criterion rules," *IEEE Signal Proc. Mag.*, vol. 21, no. 4, pp. 36 – 47, 2004.

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