Multi-way clustering via tensor block models

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider the problem of simultaneously clustering each mode of a large noisy tensor. We assume that the tensor elements are distributed with a block-specific mean and propose a tensor block model for multi-way clustering. The statistical convergence of the estimator is established, and we show that the resulting clustering achieves partition consistency. A sparse estimation is further develop for identifying the important blocks. Our proposal amounts to a sparse, higher-order generalization of k-mean clustering, and a relaxation of our proposal yields the tensor Tucker decomposition. The performance of our proposals are demonstrated in simulations and on multi-tissue gene expression datasets.

1 Introduction

Higher-order tensors have recently received increasing attention in many fields, such as neuroscience [1, 2], social networks [3, 4], computer vision [5, 6], and genomics [7, 8]. ... real value. In a different setting.

The goal of the present paper is to develop a method and related theory for the estimation of tensors with block structure. We propose a unified estimation procedure for tensors with multi-way clustering structure. We establish high probability upper bounds for the mean squared errors of the resulting estimators. We consider the task of simultaneously clustering each mode of a large noisy tensor. Figure (1) shows an example of tensor clustering by using our proposed method.

We extend our method to sparse block model. We impose and such model is able to handle. ...

If only a low rank constraint is imposed on the mean tensor, then the problem becomes what is known in the literature as low-rank tensor estimation. An impressive list of methods and theories have been develop for this problem, including but not limit to tensor CP decomposition, Tucker decomposition, and train decomposition. In this paper we investigate an alternative block structure assumption, which was first proposed by in the case of matrices... Note that a block structure automatically implies low-rankness. However, if one applied a low rank tensor estimation algorithm directly in the current setting, the resulting estimator suffers an inferior error bound. Thus, a full exploitation of the block structure is necessary, which is the focus of the current paper.

The results of our paper also facilitates the detection of large blocks. .. There are also a line of works on three-way clustering. Most of the work takes two-step procedure, by. As we shown in the Analysis and Simulation, we...

2 Preliminaries

A clustering of d objects can be represented by a partition of the index set $[d] = \{1, \ldots, d\}$ into R disjoint non-empty subsets. We refer to R the clustering size. It is often convenient to represent the clustering (or partition) using the "membership matrix". A membership matrix $\boldsymbol{M}$ is an d-by-R

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

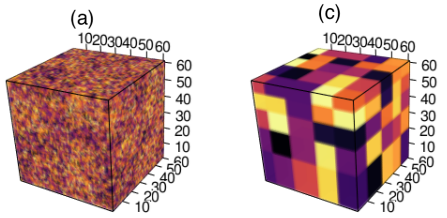

Figure 1: (a). a $60 \times 60 \times 60$ noisy tensor with 5 clusters in each mode; (b). Mean signal estimated by our proposed estimator. (c). a $60 \times 60 \times 60$ noisy tensor with sparse multi-way blocks. (d) Mean signal estimated by our proposed estimator.

matrix whose (i, j)-entry is 1 if and only if the element i belongs to the cluster j, and 0 otherwise. The membership matrix $\boldsymbol{M}$ can also be viewed as a mapping $\boldsymbol{M} : [d] \mapsto [R]$. With a little abuse of notation, we still use the $\boldsymbol{M}$ to denote the mapping, and use $\boldsymbol{M}(i)$ to denote the cluster label that entry i belongs to. Throughout the paper, we will use the terms "partition", "clustering", and "membership matrix" exchangeably.

For a higher-order tensor, the above concepts can be applied to each of the modes. We use the term "cluster" to refer to the partition along the k-th mode of the tensor, and reserve "block" for the multi-way block in the tensor. We say that an event A occurs "with high probability" if $\mathbb{P}(\boldsymbol{A})$ tends to 1 as the dimension $d_{\min} = \min\{d_1, \ldots, d_k\}$ tends to infinity. We say that A occurs "with very high probability" if $\mathbb{P}(A)$ tends to 1 faster than any polynomial of d.

We use lower-case letters $(a, b, u, v, \ldots)$ for scalars and vectors. We use upper-case boldface letters $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{C}, \ldots)$ for matrices, and calligraphy letter $(\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots)$ for tensors of order $K \geq 3$. $\boldsymbol{x} \otimes \boldsymbol{y}$ is the Kronecker product of two vectors. For any set J, $|J|$ denotes its cardinality. $[d]$ represents the set $\{1, 2, \ldots, d\}$.

3 Tensor block model

Let $\mathcal{Y} = [\![y_{i_1,\ldots,i_K}]\!] \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ denote an order-K, $(d_1, \ldots, d_K)$-dimensional data tensor. The main assumption on tensor block model is that the observed data tensor $\mathcal{Y}$ is a noisy realization of an underlying tensor that exhibits a checkerbox structure (see Figure 1). Specifically, suppose that there are R_k clusters along the k-th mode of the tensor for $k \in [K]$. If the tensor entry $y_{i_1,\ldots,i_K}$ belongs to the block jointly determined by the r_k-th mode-k cluster with $r_k \in [R_k]$, then we assume that

$$y_{i_1,\ldots,i_K} = c_{r_1,\ldots,r_K} + \varepsilon_{i_1,\ldots,i_K}, \quad \text{for } (i_1, \ldots, i_K) \in [d_1] \times \cdots \times [d_K], \tag{1}$$

where $\mu_{r_1,\ldots,r_K}$ is the mean of the tensor block indexed by $(r_1, \ldots, r_K)$, and $\varepsilon_{i_1,\ldots,i_K}$'s are independent, mean-zero noise terms to be specified later. Our goal is to (i) find partitions along each of the modes, and (ii) estimate the block means $\{c_{r_1,\ldots,r_K}\}$, such that a corresponding blockwise-constant checkerbox structure emerges in the data tensor.

The above tensor block model (1) falls into a larger class of non-overlapping, constant-mean clustering models [9], in that each tensor entry belongs to exactly one block with a common mean. The model (1) can be equivalently expressed as a special tensor Tucker model,

$$\mathcal{Y} = \mathcal{C} \times_1 \boldsymbol{M}_1 \times_2 \cdots \times_K \boldsymbol{M}_K + \mathcal{E}, \tag{2}$$

where $\mathcal{C} \in \mathbb{R}^{R_1 \times \cdots \times R_K}$ is a core tensor consisting of block means, $\boldsymbol{M}_k \in \{0, 1\}^{R_k \times d_k}$ are membership matrices indicating the block allocations along mode k for $k \in [K]$, and $\mathcal{E} = [\![\varepsilon_{i_1,\ldots,i_K}]\!]$ is the noise tensor. The distinction between our model (2) and a classical Tucker model is that we require the factors $\boldsymbol{M}_k$ to be membership matrices. Our model (2) can be viewed as a super-sparse Tucker model, in the sense that the each column of $\boldsymbol{M}_k$ consists of one copy of 1's and massive 0's.

We now introduce the assumptions on the noise tensor $\mathcal{E}$. We assume that $\varepsilon_{i_1,\ldots,i_K}$'s are independent, mean-zero, σ-subgaussian noises, where $\sigma > 0$ is the subgaussianity parameter. More precisely,

$$\mathbb{E}e^{\lambda \varepsilon_{i_1,\ldots,i_K}} \leq e^{\lambda^2 \sigma^2 / 2}, \quad \text{for all } (i_1, \ldots, i_K) \in [d_1] \times \cdots \times [d_K] \text{ and } \lambda \in \mathbb{R}. \tag{3}$$

Th assumption (3) is fairly general, which includes many common noises, such as Gaussian errors, Bernoulli errors, bounded errors, or even combinations of them. In particular, we consider two examples of the tensor block model that commonly appear in the literature:

Example 1 (Gaussian Multi-Cluster Model) *Let $\mathcal{Y}$ be a continuous-valued tensor. The Gaussian Multi-cluster model $y_{i_1,\ldots,i_K} \sim_{i.i.d.} N(\mu_{r_1,\ldots,r_K}, \sigma^2)$ is a special case of model* (1) *with the subgaussianity parameter σ equal to the error variance.*

Example 2 (Stochastic Block Model) *Let $\mathcal{Y}$ be a binary-valued tensor. The multiway stochastic block model $y_{i_1,\ldots,i_K} \sim_{i.i.d.}$ Bernoulli$(\mu_{r_1,\ldots,r_K})$ is a special case of model* (1) *with the subgaussianity parameter σ equal to $\frac{1}{4}$.*

More generally, our model also applied to hybrid error distributions in which different types of distribution can be allowed for different portions of the data. This scenario may happen, for example, when the data tensor $\mathcal{Y}$ represents concatenated measurements from multiple data sources.

We consider a least-square approach for estimating model (1). Let $\Theta = \mathcal{C} \times_1 \boldsymbol{M}_1 \times_2 \cdots \times_K \boldsymbol{M}_K$ denote the mean signal tensor with block structure. The mean tensor is assumed to belong to the following parameter space

$$\mathcal{P}_{R_1,\ldots,R_K} = \big\{\Theta \in \mathbb{R}^{d_1 \times \cdots \times d_K} : \Theta = \mathcal{C} \times_1 \boldsymbol{M}_1 \times_2 \cdots \times_K \boldsymbol{M}_K, \text{with some} \tag{4}$$

$$\text{membership matrices } \boldsymbol{M}_k\text{'s and a core tensor } \mathcal{C} \in \mathbb{R}^{R_1 \times \cdots \times R_K} \big\}. \tag{5}$$

As in most previous work on tensor clustering, we assume that clustering size $\boldsymbol{R} = (R_1, \ldots, R_K)$ are known in our theoretical analysis and simply write $\mathcal{P}$ for short. In practice, $\boldsymbol{R}$ needs to be determined from data; we address this general case in Section 5.2. The least-square estimator for model (1) is

$$\hat{\Theta} = \underset{\Theta \in \mathcal{P}}{\arg\min} \left\{-2\langle \mathcal{Y}, \Theta \rangle + \|\Theta\|_F^2\right\}. \tag{6}$$

The objective is equal (ignoring constants) to the sum of squares $\|\mathcal{Y} - \Theta\|_F^2$ and hence the name of our estimator. Estimating Θ consists of finding both the core tensor $\mathcal{C}$ and the membership matrix estimates $\boldsymbol{M}_k$'s. Before we discuss the properties of $\hat{\Theta}$, we present the identifiability of $\boldsymbol{M}_k$'s and $\mathcal{C}$ from Θ.

The following irreducible assumption is necessary for the tensor block model to be identifiable.

Assumption 1 (Irreducible cores) *The core tensor $\mathcal{C}$ is called irreducible if it cannot be written as a block tensor with the number of mode-k clusters smaller than R_k, for any $k \in [K]$.*

In the matrix case ($K = 2$), the assumption is equivalent to saying that $\mathcal{C}$ has no two identical rows and no two identical columns. In the higher-order case, it requires that none of order-(K-1) fibers of $\mathcal{C}$ are identical. Note that the being irreducible is a weaker assumption than being full rank.

Proposition 1 (Identifiability) *Consider a Gaussian or Bernoulli tensor block model* (2)*. Suppose the core tensor satisfies Assumption 1. Then every factor matrix $\boldsymbol{M}_k$ is identifiable up to permutations of cluster labels.*

Our identifiability result is stronger than the classical Tucker model. In a classical Tucker model [10, 11] and many other factor analyses [12, 13], the factors are identifiable only up to orthogonal rotations. In those models, the (column) space spanned by $\boldsymbol{M}_k$ can be recovered, but not the individual factors. In contrast, our model does not suffer from rotational invariance, and as we show in Section 4, every single factor can be consistently estimated in high dimensions. This brings a benefit to the interpretation of tensor factors in the block model.

4 Statistical convergence

In this section, we assess the estimation accuracy of the least-squares estimator (6). The estimation accuracy is assessed using mean squared error (MSE):

$$\text{MSE}(\Theta_{\text{true}}, \hat{\Theta}) = \frac{1}{\prod_k d_k} \|\Theta_{\text{true}} - \hat{\Theta}\|_F^2, \tag{7}$$

where $\Theta_{\text{true}} \in \mathcal{P}$ is the true mean tensor and $\hat{\Theta} \in \mathcal{P}$ is the estimator.

Theorem 1 (Convergence rate) *Let $\hat{\Theta}$ be the least-square estimator of Θ_{true} under model* (1)*. There exist two constants $C_1, C_2 > 0$ such that, with very high probability,*

$$MSE(\Theta_{true}, \hat{\Theta}) \leq \frac{C_1 \sigma^2}{\prod_k d_k} \left(\prod_k R_k + \sum_k d_k \log R_k \right), \tag{8}$$

holds uniformly over $\Theta_{true} \in \mathcal{P}_{\boldsymbol{R}}$ and all error distribution satisfying (3)*.*

The convergence rate in (8) consists of two parts. The first part $\prod_k R_k$ reflects the complexity for estimating the core tensor $\mathcal{C}$, while the second part $\sum_k d_k \log R_k$ results from the complexity for estimating the supports of $\boldsymbol{M}_k$'s. It is the price that one has to pay for not knowing the locations of the blocks.

We now compare our bound with existing literature. The classical Tucker tensor decomposition has a minimax convergence rate $\sum_k d_k R_k'$ [10], where R_k' is the multilinear rank at mode k. In the case of block model, this yields $\sum_k d_k R_k$, because the mode-k rank is bounded by the number of clusters in mode-k. Now, as both the dimension $d_{\min} = \min_k d_k$ and clustering size $R_{\min} = \min_k R_k$ tend to infinity, we have $\prod_k R_k + \sum_k d_k \log R_k \ll \sum_k d_k R_k$. Therefore, by fully exploiting the block structure, we obtain a better convergence rate than previously possible.

Our bound generalizes the previous results on structured matrix estimation in network analysis [14, 15]. The optimal convergence rate for estimating the (matrix) stochastic block model was $R_1 R_2 + d_1 \log R_1 + d_2 \log R_2$ [14], which fits into our special case when $K = 2$. Earlier work [15] suggests the following heuristics on the sample complexity for high-dimensional matrix problems:

$$\frac{\text{(number of parameters)} + \log \text{(complexity of models)}}{\text{number of samples}}. \tag{9}$$

Our result supports this important principle for general $K \geq 2$. Note that, in tensor estimation, the total number of entries corresponds to the sample size $\prod_k d_k$, the number of parameters is $\prod_k R_k$, and combinatoric complexity for estimating block models is of order $\prod_k R_k^{d_k}$. The principle (9) thus provide an intuition for (8).

We next study the clustering consistency of our method. Let $\boldsymbol{A}_k, \boldsymbol{B}_k$ be the two membership matrices along mode-k. We define the misclassification rate as $\text{MCR}(\boldsymbol{A}_k, \boldsymbol{B}_k) = d_k^{-1} \sum_{i \in [d_k]} \mathbb{1}_{\{\hat{\boldsymbol{A}}_k(i) = \boldsymbol{B}_k(i)\}}$. The following Theorem implies that our method achieves clustering consistency.

Theorem 2 (Clustering consistency) *Suppose the Assumption* (1) *holds. Let $\hat{\boldsymbol{M}}_k$'s be the estimators from* (6)*. Then the proportions of misclassified indices goes to zero in probability; i.e. there exist permutation matrices $\boldsymbol{P}_k$'s such that*

$$\sum_k MCR(\hat{\boldsymbol{M}}_k, \boldsymbol{P}_k \boldsymbol{M}_{k,true}) \to 0, \quad \textit{in probability},$$

(add the proof) Under stronger distribution assumptions on $\mathcal{E}$, we can establish the finite-sample convergence rate for the clustering accuracy. See more results in the Supplements.

5 Numerical Implementation

5.1 Alternating optimization

We introduce an alternating optimization for solving (6). Note that the optimization (6) can be written as

$$(\hat{\mathcal{C}}, \{\hat{\boldsymbol{M}}_k\}) = \underset{\mathcal{C} \in \mathbb{R}^{R_1 \times \cdots \times R_K}, \text{ membership matrices } \boldsymbol{M}_k\text{'s}}{\arg\min} f(\mathcal{C}, \{\boldsymbol{M}_k\}),$$
$$\text{where} \quad f(\mathcal{C}, \{\boldsymbol{M}_k\}) = \|\mathcal{Y} - \mathcal{C} \times_1 \boldsymbol{M}_1 \times_2 \ldots \times_K \boldsymbol{M}_K\|_F^2.$$

The decision variables consists of $K + 1$ blocks of variables, one for the core tensor $\mathcal{C}$ and K for the membership matrices $\boldsymbol{M}_k$'s. We notice that, if any K out of the $K + 1$ blocks of variables are known, then the last block of variables can be solved explicitly. This observation suggests that we

can iteratively update one block of variables at a time while keeping other others fixed. Specifically, given the collection of $\hat{\boldsymbol{M}}_k$'s, the core tensor estimate $\hat{\mathcal{C}} = \arg\min_{\mathcal{C}} f(\mathcal{C}, \{\hat{\boldsymbol{M}}_k\})$ collects the sample averages within each tensor block. Given the block mean $\hat{\mathcal{C}}$ and $K-1$ membership matrices, the last membership matrix can be solved using simple nearest neighbor search over only R_k discrete points. The full procedure is described in Algorithm 1.

Algorithm 1 Multiway clustering based on tensor block models

Input: Data tensor $\mathcal{Y} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$, clustering size $\boldsymbol{R} = (R_1, \ldots, R_K)$.

Output: Block mean tensor $\hat{\mathcal{C}} \in \mathbb{R}^{R_1 \times \cdots \times R_K}$, and the membership matrices $\hat{\boldsymbol{M}}_k$.

1: Initialize the marginal clustering by performing independent k-means on each of the K modes.
2: **repeat**
3: Update the core tensor $\hat{\mathcal{C}} = [\![\hat{c}_{r_1,\ldots,r_K}]\!]$. Specifically, for each $(r_1, \ldots, r_K) \in [R_1] \times \cdots [R_K]$,

$$\hat{c}_{r_1,\ldots,r_K} = \frac{1}{n_{r_1,\ldots,r_K}} \sum_{\hat{\boldsymbol{M}}_1^{-1}(r_1) \times \cdots \times \hat{\boldsymbol{M}}_K^{-1}(r_K)} \mathcal{Y}_{i_1,\ldots,i_K}, \tag{10}$$

where $n_{r_1,\ldots,r_K} = \prod_k |\hat{\boldsymbol{M}}_k^{-1}(r_k)|$ is the number of entries in the block indexed by $(r_1, \ldots, r_K)$.
4: Update membership matrices $\hat{\boldsymbol{M}}_k$'s:
5: **for** k in $\{1, 2, ..., K\}$ **do**
6: Update the mode-k membership matrix $\hat{\boldsymbol{M}}_k$. Specifically, for each $a \in [d_k]$, assign the cluster label $\hat{\boldsymbol{M}}_k(a) \in [R_k]$ for which

$$\hat{\boldsymbol{M}}_k(a) = \arg\min_{r \in [R_K]} \sum_{\boldsymbol{I}_{-k}} \left(c_{\hat{\boldsymbol{M}}_1(i_1),\ldots,r,\ldots,\hat{\boldsymbol{M}}_K(i_K)} - \mathcal{Y}_{i_1,\ldots,a,\ldots,i_K} \right)^2,$$

where $\boldsymbol{I}_{-k} = (i_1, \ldots, i_{k-1}, i_{k+1}, \ldots, i_K)$ denotes the coordinates except the k-th mode.
7: **end for**
8: **until** Convergence

The above algorithm can be viewed as a higher-order extension for the ordinary (one-way) k-means algorithm. The core tensor $\mathcal{C}$ serves as the role of centroids. As each iteration reduces the value of the objective function, which is bounded below, convergence of the algorithm is guaranteed. On the other hand, obtaining the global optimizer for such a non-convex optimization is typically difficult, even for the one-way k-means [16]. As such, we run the algorithm multiple times, using random initializations of the independent one-way k-mean on each of the K modes.

5.2 Tuning parameter selection

Our algorithm 1 takes the number of clusters $\boldsymbol{R}$ as an input. In practice such information is often unknown and $\boldsymbol{R}$ needs to be estimated from the data $\mathcal{Y}$. We propose to select this tuning parameter using Bayesian information criterion (BIC),

$$\text{BIC}(\boldsymbol{R}) = \log\left(\|\mathcal{Y} - \hat{\Theta}\|_F^2 \right) + \frac{\sum_k \log d_k}{\prod_k d_k} p_e, \tag{11}$$

where p_e is the effective number of parameters in the model. In our case we take $p_e = \prod_k R_k + \sum_k d_k \log R_k$, which is inspired from (9). We choose $\hat{\boldsymbol{R}}$ that minimizes $\text{BIC}(\boldsymbol{R})$ via grid search. Our choice of BIC is based on heretics, and we test its empirical performance in Section 7.

6 Extension to regularized estimation

In some large-scale applications, not every block in a data tensor is of equally importance. Examples include genome expression data, in which only a few entries represent the signals while the majority comes from the background noise (see Figure **??**). Another example is community detection in a large sparse network, where only a few blocks represent the communities of interest and others represent small, noisy groups with weak connection. Although our estimator (6) can still handle

this scenario by assigning small value to some of the $\hat{c}_{r_1,\ldots,r_K}$'s, the estimates may suffer from high variance. It is thus beneficial to introduce regularized estimation for better bias-variance trade-off and improved interpretability.

Here we illustrate the regularized estimation using *sparsity* on the block means for localizing important blocks in the data tensor. This problem can be cast into variable selection on the block parameters. We propose the following regularized least-square estimation:

$$\hat{\Theta}^{\text{sparse}} = \arg\min_{\Theta \in \mathcal{P}} \left\{ \|\mathcal{Y} - \Theta\|_F^2 + \lambda \|\mathcal{C}\|_\rho \right\},$$

where $\|\mathcal{C}\|_\rho$ is the penalty function with ρ being an index for the tensor norm, $\mathcal{C} \in \mathbb{R}^{R_1 \times \cdots \times R_K}$ is the block means, and λ is the penalty tuning parameter. Some widely used penalties include Lasso penalty ($\rho = 1$), sparse subset penalty ($\rho = 0$), ridge penalty ($\rho =$ Frobenius norm), elastic net (linear combination of $\rho = 1$ and $\rho =$ Frobenius norm), among many others.

For parsimony purpose, we only discuss the properties for Lasso and sparse subset penalties here; other penalizations can be derived similarly. Sparse estimation incurs slight changes to Algorithm 1. When updating the core tensor $\mathcal{C}$ in (10), we fit a penalized least square problem with respect to $\mathcal{C}$. The closed form for the entry-wise sparse estimate $\hat{c}^{\text{sparse}}_{r_1,\ldots,r_K}$ is (See Lemma 1 in Appendix):

$$\hat{c}^{\text{sparse}}_{r_1,\ldots,r_K} = \begin{cases} \hat{c}^{\text{ols}}_{r_1,\ldots,r_K} \mathbb{1}_{\{|\hat{c}^{\text{ols}}_{r_1,\ldots,r_K}| \geq \frac{2\sqrt{\lambda}}{\sqrt{n_{r_1,\ldots,r_K}}}\}} & \text{if } \rho = 1, \\ \text{sign}(\hat{c}^{\text{ols}}_{r_1,\ldots,r_K}) \left(\hat{c}^{\text{ols}}_{r_1,\ldots,r_K} - \frac{2\lambda}{n_{r_1,\ldots,r_K}} \right) & \text{if } \rho = 0. \end{cases} \tag{12}$$

where $\hat{c}^{\text{ols}}_{r_1,\ldots,r_K}$ denotes the ordinary least-square estimate in (10), for all $(r_1, \ldots, r_K) \in [d_1] \times \cdots [d_K]$. The choice of penalty function ρ often depends on the goals and interpretations in specific applications. Given a penalization, we choose to select the tuning parameter λ via BIC (11), where we modify p_e into $p_e^{\text{sparse}} = \|\hat{\mathcal{C}}^{\text{sparse}}\|_0 + \sum_k d_k \log R_k$, where $\|\cdot\|_0$ denotes the number of non-zero entries in the tensor. The empirical performance of this proposal will be evaluated in Section 7.

7 Experiments

In this section, we evaluate the empirical performance of our method. We consider both non-sparse and sparse tensors, and compare the recovery accuracy with other tensor-based methods.

7.1 Finite-sample performance

We generate noisy order-3 tensors under the tensor block model (1). We consider various values of dimension $\boldsymbol{d} = (d_1, d_2, d_3)$ and clustering size $\boldsymbol{R} = (R_1, R_2, R_3)$ as we described below. Along each mode, the tensor entries are randomly assigned into clusters with uniform probability. The block means are generated i.i.d. from Unif[-3,3]. The entries in the noise tensor $\mathcal{E}$ are generated from i.i.d. Gaussian $(0, \sigma^2)$. In each simulation study, we report the summary statistics across $n_{\text{sim}} = 50$ replications.

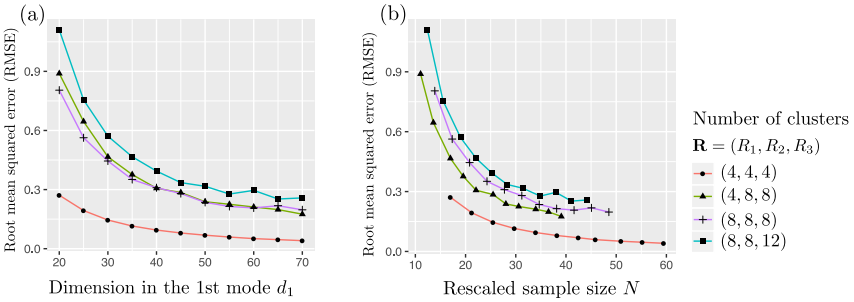

Figure 2: Estimation error for block tensors with Gaussian noise. Each curve corresponds to a fixed clustering size $\boldsymbol{R}$. (a) Plot of average RMSE against d_1. (b) Plot of average RMSE against rescaled sample size $N = \sqrt{d_2 d_3 / \log R_1}$.

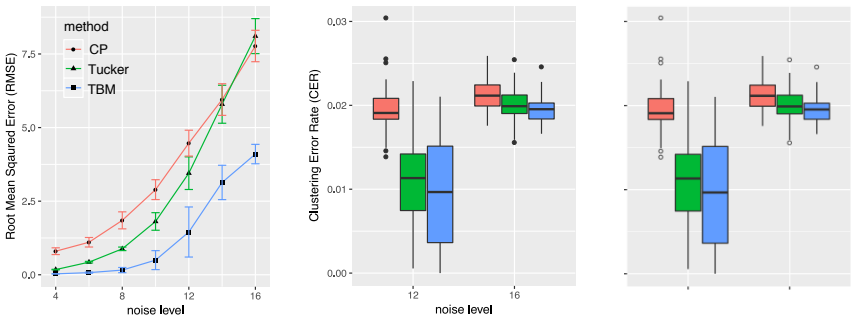

Figure 3: Performance comparison in terms of RMSE and CER. (a) Plot of the estimation error against noise. (b) Boxplot of the clustering error against noise for tensors of dimension $(40, 40, 40)$. (c) Boxplot of the clustering error against noise for tensors of dimension $(60, 60, 60)$.

In the first experiment, we assess the empirical relationship between the root mean squared error (RMSE) and the dimension. We set $\sigma = 3$ and consider four different $\boldsymbol{R}$ settings (see Figure 2). We increases d_1 from 20 to 70, and for each choice of d_1, we set the other two dimensions (d_2, d_3) such that $d_1 \log R_1 \approx d_2 \log R_2 \approx d_3 \log R_3$. Our theoretical analysis suggests that the RMSE converges at least at the rate of $\sqrt{\log R_1 / d_2 d_3}$ in this case. Figure 2a plots the recovery error versus the dimension d_1. We rescaled the x-axis in Figure 2b, and find that RMSE decreases roughly at the rate of $1/N$, where $N = \sqrt{d_2 d_3 / \log R_1}$ is the rescaled sample size. This is consistent to our theoretical result. It is observed that tensors with a higher number of blocks tend to yield higher recovery errors, as reflected by the upward shift of the curves as $\boldsymbol{R}$ increases. Indeed, a higher $\boldsymbol{R}$ means a higher intrinsic dimension of the problem, thus increasing the difficulty of the estimation.

In the second experiment, we evaluate the selection performance of our BIC criterion (11). Supplementary Table 1 reports the selected numbers of clusters various combinations of dimension $\boldsymbol{d}$, clustering size $\boldsymbol{R}$, and noise σ. We find that, for the case $\boldsymbol{d} = (40, 40, 40)$ and $\boldsymbol{R} = (4, 4, 4)$, the BIC selection is accurate in the low-to-moderate noise setting. In the case of high noise $\sigma = 12$, the selected number of clusters is slightly smaller than the true number, but the accuracy increases when either the dimension increases to $\boldsymbol{d} = (40, 40, 80)$ or the clustering size reduces to $\boldsymbol{R} = (2, 3, 4)$. Within a tensor, the selection seems to be easier for shorter modes with smaller number of clusters, and this is to be expected, since shorter mode has more effective samples for clustering.

7.2 Comparison with alternative methods

Next, we compare our method with two popular low-rank tensor estimation methods: (i) CP decomposition and (ii) Tucker decomposition. The CP model decomposes a tensor into a sum of rank-1 tensors, whereas Tucker model decomposes a tensor into a core tensor multiplied by orthogonal matrices in each mode. Following the literature [17], we perform the clustering by applying the k-means to the resulting factors along each of the modes. We refer to such techniques as CP+k-means and Tucker+k-means.

For easy reference, we denote our method by TBM (tensor block model)[1]. We generate noisy block tensors with 5 clusters on each of the modes, and then assess both the estimation and clustering performance for each method. Note that our method takes a single step to perform estimation and clustering simultaneously, whereas the CP and Tucker-based approaches take a two-step procedure. We use the RMSE to assess the estimation accuracy and use the clustering error rate (CER) to measure the clustering accuracy. The CER is calculated using the disagreements (i.e., one minus rand index) between the true and estimated block partitions in the three-way data. For fair comparison, we provide all methods the true number of clusters.

Figure 3 compares the performance. We find that TBM achieves the lowest estimation error among the three methods. The gain in accuracy is more pronounced as the noise grows. Both CP and Tucker fail to accurately estimate the signal tensor, but Tucker appears to have a modest clustering performance. One possible explanation is that Tucker enforces orthogonality in the factors which

[1]TBM implementation: https://github.com/wanglab/tensorsparse

make the subsequent k-means clustering easier. Figure 3 shows that the clustering error increases with noise but decreases with dimension. This agrees with our expectation, as in tensorial data analysis, a larger dimension implies a larger sample size.

Sparse case. We then assess the performance when the signal tensor is sparse. We modify the generation of block means using a mixture distribution of zero mass and Unif[-3,3], with probability p and $1-p$ respectively. The p is called the sparsity rate. We generate noisy tensors of dimension $\boldsymbol{d} = (40, 40, 40)$ with varying levels of sparsity and noise. The initial clustering size is set $\boldsymbol{R} = (3, 4, 5)$, but the actual number of clusters may be smaller because zero-mean blocks could be merged together. We primarily focus on the estimation and selection accuracy. The selection accuracy is quantified via the the sparsity error rate, which is the proportion of entries that were incorrectly set to zero or incorrectly set to non-zero. We also report the proportion of true zero's that were correctly identified (correct zeros).

We utilize $\ell 0$-penalized TBM and select the sparse parameter λ via BIC. Table 1 reports the selected λ averaged across 50 simulations. We see a substantial benefit obtained by penalization. Our proposed λ is able to guide the algorithm to correctly identify zero's while maintaining good accuracy in identifying non-zero's. The resulting sparsity level is close to the ground truth. The rows with $\lambda = 0$ correspond to the three non-sparse algorithms (CP, Tucker, and non-sparse TBM). Because non-sparse algorithms fail to identify zero's, they show equally poor performance in all metrics. Supplementary Figure **??** shows the the estimation error in terms of RMSE. Again, the sparse TBM outer-performs the other two methods.

Sparsity (ρ)	Noise (σ)	Penalization (λ)	Estimated Sparsity Rate	Correct Zero Rate	Sparsity Error Rate
0.5	4	$\lambda = 0$	$0(0)$	$0(0)$	$0.49(0.07)$
		$\bar{\lambda} = 86.6$	$\mathbf{0.56(0.07)}$	$\mathbf{0.99(0.01)}$	$\mathbf{0.07(0.04)}$
0.5	8	$\lambda = 0$	$0(0)$	$0(0)$	$0.49(0.07)$
		$\bar{\lambda} = 344.4$	$\mathbf{0.63(0.07)}$	$\mathbf{0.99(0.01)}$	$0.14(0.05)$
0.8	8	$\lambda = 0$	$0(0)$	$0(0)$	$0.80(0.05)$
		$\bar{\lambda} = 246.9$	$\mathbf{0.83(0.06)}$	$\mathbf{0.95(0.04)}$	$\mathbf{0.12(0.06)}$

Table 1: Results for sparse tensor block estimation under dimension $\boldsymbol{d} = (40, 40, 40)$. The reported $\bar{\lambda}$ is the mean of λ selected across 50 simulations using our proposed BIC criterion. Number in bold indicates no significant difference between the estimate and the ground truth, based on a z-test with a level 0.05.

8 Real data analysis

Lastly, we applied our method on two real data sets, one is a real-valued tensor and another is a Bernoulli-valued tensor. The real-valued dataset consists of gene expressions from 13 brain tissues, 193 individuals, and 362 genes. The dataset is obtained from GTEx contortion, and the gene list comes from... We subtract the overall mean and apply the penalized TBM method on this tensor. Tensor blocks are identified. We select 10 blocks and . We find that tissues are collected.

Another dataset we consider is the *Nations* data. The nations data set is a $14 \times 14 \times 56$ binary tensor consisting of 56 political relationships of 14 countries between 1950 and 1965. There are 78.9% of entries are zero. We applied the BIC criterion and choose $\boldsymbol{R} = (5, 5, 9)$ and $\lambda = 0.4$. The countries are clustered into 5 groups: USA, UK, east, nural

We order the block by the mead and find that there are 17 blocks with mean 1.

9 Conclusion

Sparsity is only one form of regularization. In specific applications, prior knowledge often suggests various constraints among parameters, which may be exploited to regularize parameter estimates. For example, in the stochastic block model, sometimes it may be reasonable to impose symmetry on the parameters along certain subsets of modes, which further reduces the dimension of the problem. In some other applications, non-negativity of parameter of parameter values may be enforced. In our software, we implement the common penalizations but leave .. .to further study.

References

[1] Fengyu Cong, Qiu-Hua Lin, Li-Dan Kuang, Xiao-Feng Gong, Piia Astikainen, and Tapani Ristaniemi. Tensor decomposition of EEG signals: a brief review. *Journal of neuroscience methods*, 248:59–69, 2015.

[2] Hua Zhou, Lexin Li, and Hongtu Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.

[3] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *International Conference on Machine Learning*, volume 11, pages 809–816, 2011.

[4] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013.

[5] Yichuan Tang, Ruslan Salakhutdinov, and Geoffrey Hinton. Tensor analyzers. In *International Conference on Machine Learning*, pages 163–171, 2013.

[6] Ji Liu, Przemyslaw Musialski, Peter Wonka, and Jieping Ye. Tensor completion for estimating missing values in visual data. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):208–220, 2013.

[7] Miaoyan Wang, Jonathan Fischer, and Yun S Song. Three-way clustering of multi-tissue multi-individual gene expression data using constrained tensor decomposition. *bioRxiv 229245*, 2017.

[8] Victoria Hore, Ana Viñuela, Alfonso Buil, Julian Knight, Mark I McCarthy, Kerrin Small, and Jonathan Marchini. Tensor decomposition for multiple-tissue gene expression experiments. *Nature genetics*, 48(9):1094, 2016.

[9] Sara C Madeira and Arlindo L Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(1):24–45, 2004.

[10] Anru Zhang and Dong Xia. Tensor SVD: Statistical and computational limits. *IEEE Transactions on Information Theory*, 2018.

[11] Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

[12] Robin A Darton. Rotation in factor analysis. *Journal of the Royal Statistical Society: Series D (The Statistician)*, 29(3):167–194, 1980.

[13] Hervé Abdi. Factor rotations in factor analyses.

[14] Chao Gao, Yu Lu, Zongming Ma, and Harrison H Zhou. Optimal estimation and completion of matrices with biclustering structures. *The Journal of Machine Learning Research*, 17(1):5602–5630, 2016.

[15] Chao Gao and Zongming Ma. Minimax rates in network analysis: Graphon estimation, community detection and hypothesis testing. *arXiv preprint arXiv:1811.06055*, 2018.

[16] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. Np-hardness of euclidean sum-of-squares clustering. *Machine learning*, 75(2):245–248, 2009.

[17] Eric C Chi, Brian R Gaines, Will Wei Sun, Hua Zhou, and Jian Yang. Provable convex co-clustering of tensors. *arXiv preprint arXiv:1803.06518*, 2018.