Multiway clustering via tensor block models

Anonymous Author(s)
Affiliation
Address
email

Abstract

We consider the problem of identifying multiway block structure from a large noisy tensor. Such problems arise frequently in applications such as genomics, recommendation system, topic modeling, and sensor network localization. We propose a tensor block model, develop a unified least-square estimation, and obtain the theoretical accuracy guarantees for multiway clustering. The statistical convergence of the estimator is established, and we show that the associated clustering procedure achieves partition consistency. A sparse regularization is further developed for identifying important blocks with elevated means. The proposal applies to a broad range of data types, including binary, continuous, and hybrid observations. We demonstrate our procedure with applications to gene expression data and multi-layer network analysis.

1 Introduction

Higher-order tensors have recently attracted increased attention in data-intensive fields such as neuroscience [1, 2], social networks [3, 4], computer vision [5, 6], and genomics [7, 8]. In many applications, the data tensors are often expected to have underlying block structures. One example is multi-tissue expression data [7], in which genome-wide expression profiles are collected from different tissues in a number of individuals. There may be groups of genes that are similarly expressed in subsets of tissues and individuals; mathematically, this implies an underlying three-way block structure in the data tensor. In a different context, block structure can emerge in a binary-valued tensor. Examples include multilayer network data [3], with the nodes representing the individuals and the layers representing the multiple types of relations. Here a planted block represents a community of individuals that are highly connected within a class of relationships.

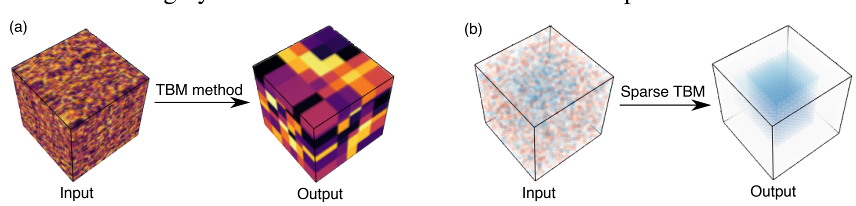

Figure 1: Examples of tensor block model (TBM). (a) Our TBM method can be used for multiway clustering and for revealing the underlying checkerbox structure in a noisy tensor. (b) The sparse TBM method can be used for detecting sub-tensors of elevated means.

This paper presents a new method and the associated theory for tensors with block structure. We develop a unified least-square estimation procedure for identifying multiway block structures. The proposal applies to a broad range of data types, including binary, continuous, and hybrid observations. We establish a high-probability error bound for the resulting estimator, and show that the procedure enjoys consistency guarantees on the block structure recovery as the dimension of the data tensor grows. Furthermore, we develop a sparse extension of the tensor block model for block selections. Figure 1 shows two immediate examples of our method. When the data tensor possesses a checkerbox pattern modulo some unknown reordering of entries, our method amounts to multiway clustering that simultaneously clusters each mode of the tensor (Figure 1a). When the data tensor has no full

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

checkerbox structure but contains a small numbers of sub-tensors of elevated means, we can use the the sparse version of our method to detect these sub-tensors of interest (Figure 1b).

Related work. Our work is closely related to, but also clearly distinctive from, the low-rank tensor decomposition. A number of methods have been developed for low-rank tensor estimation, including CANDECOMP/PARAFAC (CP) decomposition [9] and Tucker decomposition [10]. 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. In this paper we investigate an alternative block structure assumption, which is much less studied for higher-order tensors. Note that a block structure automatically implies Tucker low-rankness. However, as we will shown in Section 4, a direct application of low rank estimation to the current setting will result in an inferior estimator. Therefore, a full exploitation of the block structure is necessary; this is the focus of the current paper.

Our work is also connected to biclustering and its higher-order extensions [11, 12, 13]. Existing multiway clustering methods [12, 13, 8] typically take a two-step procedure, by first estimating a low-dimension representation of the data tensor and then applying clustering algorithms to the tensor factors. In contrast, our tensor block model takes a single shot to perform estimation and clustering simultaneously. We will show that this approach achieves a higher accuracy and an improved interpretability. Moreover, earlier solutions to multiway clustering [14, 12] focused on the algorithm effectiveness, but did not address the statistical optimality of the estimators. Very recently, [15] provides the first attempt to study the statistical properties of tensor block models. We will show that our estimator obtains a much faster convergence rate than theirs, and that the power can be further boosted with a sparse regularity.

2 Preliminaries

We begin by reviewing a few basic factors about tensors [16]. We use $\mathcal{Y} = [\![y_{i_1,\ldots,i_K}]\!] \in \mathbb{R}^{d_1\times\cdots\times d_K}$ to denote an order-K $(d_1,\ldots,d_K)$-dimensional tensor. The multilinear multiplication of a tensor $\mathcal{Y} \in \mathbb{R}^{d_1\times\cdots\times d_K}$ by matrice $\boldsymbol{M}_k = [\![m^{(k)}_{i_k,j_k}]\!] \in \mathbb{R}^{d_k\times s_k}$ is defined as

$$\mathcal{Y} \times_1 \boldsymbol{M}_1 \ldots \times_K \boldsymbol{M}_K = [\![\sum_{i_1,\ldots,i_K} y_{i_1,\ldots,i_K} m^{(1)}_{i_1,j_1} \ldots m^{(K)}_{i_K,j_K}]\!],$$

which results in an order-K tensor $(s_1,\ldots,s_K)$-dimensional tensor. For any two tensors $\mathcal{Y} = [\![y_{i_1,\ldots,i_K}]\!]$, $\mathcal{Y}' = [\![y'_{i_1,\ldots,i_K}]\!]$ of identical order and dimensions, their inner product is defined as $\langle \mathcal{Y}, \mathcal{Y}' \rangle = \sum_{i_1,\ldots,i_K} y_{i_1,\ldots,i_K} y'_{i_1,\ldots,i_K}$. The Frobenius norm of tensor $\mathcal{Y}$ is defined as $\|\mathcal{Y}\|_F = \langle \mathcal{Y}, \mathcal{Y} \rangle^{1/2}$; it is the Euclidean norm of $\mathcal{Y}$ regarded as an $\prod_k d_k$-dimensional vector. A fiber of $\mathcal{Y}$ is an order-(K-1) sub-tensor of $\mathcal{Y}$ obtained by holding the index in one mode fixed while letting other indices vary.

A clustering of d objects can be viewed as a partition of the index set $[d] := \{1, 2, \ldots, d\}$ into R disjoint non-empty subsets. We refer to the number of clusters, R, as the clustering size. Equivalently, the clustering (or partition) can be represented using the "membership matrix". A membership matrix $\boldsymbol{M} \in \mathbb{R}^{R\times d}$ is an incidence matrix whose (i, j)-entry is 1 if and only if the element j belongs to the cluster i, and 0 otherwise. Throughout the paper, we will use the terms "clustering", "partition", and "membership matrix" exchangeably. For a higher-order tensor, the concept of index partition applies to each of the modes. A block is a sub-tensor induced by the index partitions along each of the K modes. We use the term "cluster" to refer to the marginal partition on mode k, and reserve the term "block" for the multiway partition of the tensor.

We say that an event A occurs "with high probability" if $\mathbb{P}(A)$ 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 $d_{\min}$.

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 of 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 1a). Specifically, suppose the k-th mode of the tensor consists of R_k clusters. If the tensor entry $y_{i_1,\ldots,i_K}$ belongs to the block determined by the r_kth cluster in the mode k for $r_k \in [R_k]$, 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 $c_{r_1,\ldots,r_K}$ 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 the clustering 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 tensor block model (1) falls into a general class of non-overlapping, constant-mean clustering models [17], in that each tensor entry belongs to exactly one block with a common mean. 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. 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 make a fairly general assumption on the noise tensor $\mathcal{E}$. The noise terms $\varepsilon_{i_1,\ldots,i_K}$'s are assumed to be independent, mean-zero σ-subgaussian, where $\sigma > 0$ is the subgaussianity parameter. More precisely,

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

Th assumption (3) includes many common situations such as Gaussian noise, Bernoulli noise, and noise with bounded support. In particular, we consider two examples of the tensor block model that have important implications in applications:

Example 1 (Gaussian tensor block model) *Let $\mathcal{Y}$ be a continuous-valued tensor. The Gaussian tensor block model (GTBM) $y_{i_1,\ldots,i_K} \sim_{\text{i.i.d.}} N(c_{r_1,\ldots,r_K},\sigma^2)$ is a special case of model* (1)*, with the subgaussianity parameter σ equal to the error variance. The GTBM serves as the foundation for many tensor clustering algorithms [14, 15].*

Example 2 (Stochastic tensor block model) *Let $\mathcal{Y}$ be a binary-valued tensor. The stochastic tensor block model (STBM) $y_{i_1,\ldots,i_K} \sim_{\text{i.i.d.}}$ Bernoulli$(c_{r_1,\ldots,r_K})$ is a special case of model* (1)*, with the subgaussianity parameter σ equal to $\frac{1}{4}$. The STBM can be viewed as an extension, to higher-order tensors, of the popular stochastic block model [18] for matrix-based network analysis.*

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

Before we discuss the estimation, we present the identifiability of the tensor block model.

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

Proposition 1 (Identifiability) *Consider a Gaussian or Bernoulli tensor block model* (2)*. Under Assumption 1, the factor matrices $\boldsymbol{M}_k$'s are identifiable up to permutations of cluster labels.*

Our identifiability result is stronger than that in the classical Tucker model. In the Tucker [19, 16] and many other factor analyses [20, 21], the factors are identifiable only up to orthogonal rotations. Those models can recover only the (column) space spanned by $\boldsymbol{M}_k$, 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.

We propose 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

$$\begin{aligned}\mathcal{P}_{R_1,\ldots,R_K} = \big\{\Theta\in\mathbb{R}^{d_1\times\cdots\times d_K} : \Theta = \mathcal{C}\times_1 \boldsymbol{M}_1\times_2\cdots\times_K \boldsymbol{M}_K, &\text{ with some} \\ \text{membership matrices } \boldsymbol{M}_k\text{'s and a core tensor } \mathcal{C}\in\mathbb{R}^{R_1\times\cdots\times R_K}&\big\}.\end{aligned}$$

In the following theoretical analysis, we assume that the clustering size $\boldsymbol{R} = (R_1, \ldots, R_K)$ is known and simply write $\mathcal{P}$ for short. The adaptation of unknown $\boldsymbol{R}$ will be addressed in Section 5.2. The least-square estimator for model (1) is

$$\hat{\Theta} = \operatorname*{arg\,min}_{\Theta \in \mathcal{P}} \left\{ -2\langle \mathcal{Y}, \Theta \rangle + \|\Theta\|_F^2 \right\}. \tag{4}$$

The objective is equal (ignoring constants) to the sum of squares $\|\mathcal{Y} - \Theta\|_F^2$ and hence the name of our estimator.

4 Theory

In this section, we establish the convergence rate of the least-squares estimator (4). While the loss function corresponds to the likelihood for Gaussian tensor model, the same assertion does not hold for other types of distribution such as stochastic tensor block model. Surprisingly, we will show that, with very high probability, a simple least-square estimator can achieve a nearly optimal convergence rate in a general class of block tensor models.

We define the estimation accuracy using the mean squared error (MSE):

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

where $\Theta_{\text{true}}, \hat{\Theta} \in \mathcal{P}$ are the true and estimated mean tensor, respectively.

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

$$\mathit{MSE}(\Theta_{\mathit{true}}, \hat{\Theta}) \leq \frac{C\sigma^2}{\prod_k d_k} \left(\prod_k R_k + \sum_k d_k \log R_k \right), \tag{5}$$

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

The convergence rate in (5) consists of two parts. The first part $\prod_k R_k$ is the number of parameters in the core tensor $\mathcal{C}$, while the second part $\sum_k d_k \log R_k$ reflects the the complexity for estimating $\boldsymbol{M}_k$'s. It is the price that one has to pay for not knowing the locations of the blocks.

We compare our bound with existing literature. The Tucker tensor decomposition has a minimax convergence rate proportional to $\sum_k d_k R'_k$ [19], where R'_k is the multilinear rank in the mode k. Applying the Tucker model to the current setting yields $\sum_k d_k R_k$, because the mode-k rank is bounded by the number of clusters in the 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.

Recently, [15] proposed a convex-relaxation for estimating tensor block model. In the special case when the tensor dimensions are equal at every mode $d_1 = \ldots = d_K = d$, their estimator has a convergence rate of order $O(d^{-1})$ for all $K \geq 2$. As we see from (5), our estimate obtains a much better convergence rate $O(d^{-(K-1)})$, which is especially favorably as the order increases.

The bound (5) generalizes the previous results on structured matrix estimation in network analysis [22, 23]. Earlier work [23] suggests the following heuristics on the sample complexity for the matrix case:

$$\frac{(\text{number of parameters}) + \log(\text{complexity of models})}{\text{number of samples}}. \tag{6}$$

Our result supports this important principle for general $K \geq 2$. Note that, in the tensor block model, the sample size is the total number of entries $\prod_k d_k$, the number of parameters is $\prod_k R_k$, and combinatoric complexity for estimating block structure is of order $\prod_k R_k^{d_k}$.

We next study the clustering consistency of our method. Let $\boldsymbol{M}_k, \boldsymbol{M}'_k$ be two membership matrices in the mode k. We define the misclassification rate as $\text{MCR}(\boldsymbol{M}_k, \boldsymbol{M}'_k) = d_k^{-1} \sum_{i \in [d_k]} \mathbb{1}_{\{\hat{\boldsymbol{M}}_k(i) = \boldsymbol{M}'_k(i)\}}$. Here $\boldsymbol{M}_k(i)$ (respectively, $\boldsymbol{M}'_k(i)$) denotes the cluster label that entry i belongs to, based on the partition induced by $\boldsymbol{M}_k$ (respectively, $\boldsymbol{M}'_k$).

Theorem 2 (Clustering consistency) *Let $M_{k,true}$ be the true mode-k membership matrix and $\hat{M}_k$ the estimator from* (4)*. Under Assumption* (1)*, the proportions of misclassified entries go to zero in probability; i.e. there exist permutation matrices P_k's such that*

$$\sum_k MCR(\hat{M}_k, P_k M_{k,true}) \to 0, \quad \textit{in probability}.$$

The above theorem shows that our estimate achieves consistency block structure recovery as the dimension of the data tensor grows.

5 Numerical Implementation

5.1 Alternating optimization

We introduce an alternating optimization for solving (4). Estimating Θ consists of finding both the core tensor $\mathcal{C}$ and the membership matrices M_k's. The optimization (4) can be written as

$$(\hat{\mathcal{C}}, \{\hat{M}_k\}) = \mathop{\arg\min}_{\mathcal{C}\in\mathbb{R}^{R_1\times\cdots\times R_K},\text{ membership matrices } M_k\text{'s}} f(\mathcal{C}, \{M_k\}),$$
$$\text{where} \quad f(\mathcal{C}, \{M_k\}) = \|\mathcal{Y} - \mathcal{C} \times_1 M_1 \times_2 \ldots \times_K M_K\|_F^2.$$

The decision variables consist of $K+1$ blocks of variables, one for the core tensor $\mathcal{C}$ and K for the membership matrices 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 others fixed. Specifically, given the collection of $\hat{M}_k$'s, the core tensor estimate $\hat{\mathcal{C}} = \arg\min_{\mathcal{C}} f(\mathcal{C}, \{\hat{M}_k\})$ consists of the sample averages of each tensor block. Given the block mean $\hat{\mathcal{C}}$ and $K-1$ membership matrices, the last membership matrix can be solved using a 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 $\hat{M}_k$'s.

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 $(r_1, \ldots, r_K) \in [R_1]\times\cdots[R_K]$,

$$\hat{c}_{r_1,\ldots,r_K} = \frac{1}{n_{r_1,\ldots,r_K}} \sum_{\hat{M}_1^{-1}(r_1)\times\cdots\times\hat{M}_K^{-1}(r_K)} y_{i_1,\ldots,i_K}, \tag{7}$$

where $M_k^{-1}(r_k)$ denotes the indices that belong to the r_kth cluster in the mode k, and $n_{r_1,\ldots,r_K} = \prod_k |\hat{M}_k^{-1}(r_k)|$ denotes the number of entries in the block indexed by $(r_1, \ldots, r_K)$.
4: **for** k in $\{1, 2, ..., K\}$ **do**
5: Update the mode-k membership matrix $\hat{M}_k$. Specifically, for each $a \in [d_k]$, assign the cluster label $\hat{M}_k(a) \in [R_k]$:

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

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

Algorithm 1 can be viewed as a higher-order extension of 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. We recognize that obtaining the global optimizer for such a non-convex optimization is typically difficult [24]. Following the common practice in non-convex optimization [2], we run the algorithm multiple times using random initialization on independent one-way k-means on each of the 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),

$$\text{BIC}(\boldsymbol{R}) = \log\left(\|\mathcal{Y} - \hat{\Theta}\|_F^2\right) + \frac{\sum_k \log d_k}{\prod_k d_k} p_e, \tag{8}$$

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 (6). We choose $\hat{\boldsymbol{R}}$ that minimizes BIC($\boldsymbol{R}$) via grid search. Our choice of BIC aims to balance between the goodness-of-fit for the data and the degree of freedom in the population model. 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 equal importance. For example, in the genome-wise expression data analysis, only a few entries represent the signals while the majority come from the background noise (see Figure 1b). While our estimator (4) can still handle this scenario by assigning small values 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 formulated as a variable selection on the block parameters. We propose the following regularized least-square estimation:

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

where $\mathcal{C} \in \mathbb{R}^{R_1 \times \cdots \times R_K}$ is the block-mean tensor, $\|\mathcal{C}\|_\rho$ is the penalty function with ρ being an index for the tensor norm, 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 Lasso and sparse subset penalties; other penalizations can be derived similarly. Sparse estimation incurs slight changes to Algorithm 1. When updating the core tensor $\mathcal{C}$ in (7), 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 Proofs 5 in Appendix):

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

where $\hat{c}^{\text{ols}}_{r_1,\ldots,r_K}$ denotes the ordinary least-square estimate in (7), and $a_+ = \max(a, 0)$. The choice of penalties ρ often depends on the study goals and interpretations in specific applications. Given a penalty function, we select the tuning parameter λ via BIC (8), where we modify p_e into $p_e^{\text{sparse}} = \|\hat{\mathcal{C}}^{\text{sparse}}\|_0 + \sum_k d_k \log R_k$. Here $\|\cdot\|_0$ 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. Unless otherwise stated, we generate order-3 tensors under the Gaussian tensor block model (1). The block means are generated from i.i.d. 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.

7.1 Finite-sample performance

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 increase 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 $d_1 \log R_1 \approx d_2 \log R_2 \approx d_3 \log R_3$. Recall that our theoretical analysis suggests that the RMSE converges 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. After rescaling the x-axis as in Figure 2b, we find that the 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.

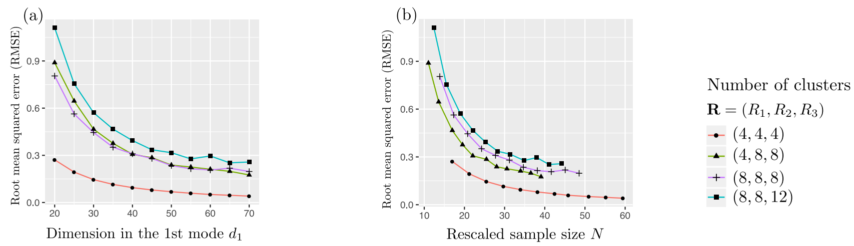

Figure 2: Estimation error for block tensors with Gaussian noise. Each curve corresponds to a fixed clustering size $\boldsymbol{R}$. (a) Average RMSE against d_1. (b) Average RMSE against rescaled sample size $N = \sqrt{d_2 d_3 / \log R_1}$.

In the second experiment, we evaluate the selection performance of our BIC criterion (8). Supplementary Table 2 reports the selected numbers of clusters under 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 high-noise noise setting with $\sigma = 12$, 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. This phenomenon 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. Following the literature [15], 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).

We generate noisy block tensors with five clusters on each of the modes, and then assess both the estimation and clustering performance for each method. Note that TBM takes a single shot to perform estimation and clustering simultaneously, whereas CP and Tucker-based methods separate these two tasks in two steps. 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 tensor. For fair comparison, we provide all methods the true number of clusters.

Figure 3(a) shows 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, although Tucker appears to have a modest clustering performance (Figure 3(b)). One possible explanation is that Tucker factors have the orthogonality property which makes the subsequent k-means clustering easier than that for the CP factors. Figure 3(b)-(c) 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. The simulated model is the same as before, except that we generate block means from a mixture of zero mass and Unif[-3,3], with probability p (sparsity rate) and $1 - p$ respectively. The performance 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).

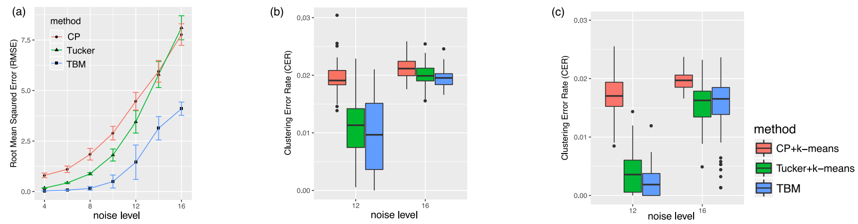

Figure 3: Performance comparison in terms of RMSE and CER. (a) Estimation error against noise for tensors of dimension $(40, 40, 40)$. (b) Clustering error against noise for tensors of dimension $(40, 40, 40)$. (c) Clustering error against noise for tensors of dimension $(40, 50, 60)$.

Table 1 reports the BIC-selected λ averaged across 50 simulations. We see a substantial benefit obtained by penalization. The 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 1 shows the estimation error RMSE against σ when $\rho = 0.8$. Again, the sparse TBM outer-performs the other 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: 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 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.

7.3 Real data analysis

Lastly, we apply our method on two real datasets. We briefly summarize the main findings here; the detailed results can be found in the Supplements.

The first dataset is a real-valued tensor, consisting of approximate 1 million expression values from 13 brain tissues, 193 individuals, and 362 genes [7]. We subtracted the overall mean expression from the data, and applied the $\ell 0$-penalized TBM to identify important blocks in the resulting tensor. The top blocks exhibit clear tissues $\times$ genes specificity. In particular, all the top 5 blocks with highly positive means are driven by tissues {*Substantia nigra, Spinal cord*} and genes {*GFAP, MBP*}, suggesting their elevated expression across many individuals. In fact, *GFAP* encodes filament proteins for mature astrocytes and *MBP* encodes myelin sheath for oligodendrocytes, both of which play important roles in the central nervous system. The top blocks with extreme negative means are driven by tissue located in the hindbrain {*Cerebellum, Cerebellar Hemisphere*} and genes {*CDH9, GPR6, RXFP1, CRH, DLX5/6, NKX2-1, SLC17A8*}. Interestingly, the identified gene *DLX6* is known to encode proteins in the forebrain development, which explains the observed under expression in the hindbrain.

The second dataset we consider is the *Nations* data [3]. This is a $14 \times 14 \times 56$ binary tensor consisting of 56 political relationships of 14 countries between 1950 and 1965. We note that 78.9% of the entries are zero. Again, we applied the $\ell 0$-penalized TBM to identify important blocks in the data. We found that the 14 countries are naturally partitioned into 5 clusters, two representing neutral countries {*Brazil, Egypt, India, Israel, Netherlands*} and {*Burma, Indonesia, Jordan*}, one eastern bloc {*China, Cuba, Poland, USSA*}, and two western blocs, {*USA*} and {*UK*}. The relation types are partitioned into 7 clusters, among which the exports-related activities {*reltreaties, book translations, relbooktranslations, exports3, relexporsts*} and NGO-related activities {*relintergovorgs, relngo, intergovorgs3, ngoorgs3*} are two major clusters that involve the connection between neutral and western blocs.

8 Conclusion

We have developed a statistical setting for studying the tensor block model. Our algorithm can be applied to a broad range of data distributions and can handle both sparse and sense data tensor. In specific applications, prior knowledge can suggests other constraints among parameters. For example, in the multi-layer network analysis, sometimes it may be reasonable to impose symmetry on the clustering along certain modes. In some other applications, non-negativity of parameter values may be enforced. We leave these directions for future 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. *Annals of Applied Statistics, in press*, 2019.

[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] Frank L Hitchcock. The expression of a tensor or a polyadic as a sum of products. *Journal of Mathematics and Physics*, 6(1-4):164–189, 1927.

[10] Ledyard R Tucker. Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31(3):279–311, 1966.

[11] Kean Ming Tan and Daniela M Witten. Sparse biclustering of transposable data. *Journal of Computational and Graphical Statistics*, 23(4):985–1008, 2014.

[12] Tamara G Kolda and Jimeng Sun. Scalable tensor decompositions for multi-aspect data mining. In *2008 Eighth IEEE international conference on data mining*, pages 363–372. IEEE, 2008.

[13] Chang-Dong Wang, Jian-Huang Lai, and S Yu Philip. Multi-view clustering based on belief propagation. *IEEE Transactions on Knowledge and Data Engineering*, 28(4):1007–1021, 2015.

[14] Stefanie Jegelka, Suvrit Sra, and Arindam Banerjee. Approximation algorithms for tensor clustering. In *International Conference on Algorithmic Learning Theory*, pages 368–383. Springer, 2009.

[15] 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.

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

[17] 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.

[18] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

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

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

[21] Hervé Abdi. Factor rotations in factor analyses. *Encyclopedia for Research Methods for the Social Sciences, Sage: Thousand Oaks*, pages 792–795, 2003.

[22] 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.

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

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