Multiway clustering via tensor block models

Miaoyan Wang
University of Wisconsin – Madison
miaoyan.wang@wisc.edu

Yuchen Zeng
University of Wisconsin – Madison
yzeng58@wisc.edu

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 handles a broad range of data types, including binary, continuous, and hybrid observations. Through simulation and application to two real datasets, we demonstrate the outperformance of our approach over previous methods.

1 Introduction

Higher-order tensors have recently attracted increased attention in data-intensive fields such as neuroscience [1], social networks [2], computer vision [3], and genomics [4, 5]. In many applications, the data tensors are often expected to have underlying block structure. One example is multi-tissue expression data [4], in which genome-wide expression profiles are collected from different tissues in a number of individuals. There may be groups of genes 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 may emerge in a binary-valued tensor. Examples include multilayer network data [2], 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.

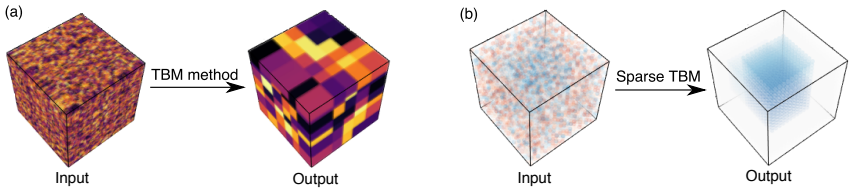

Figure 1: Examples of tensor block model (TBM). (a) Our TBM method is used for multiway clustering and for revealing the underlying checkerbox structure in a noisy tensor. (b) The sparse TBM method is used for detecting sub-tensors of elevated means.

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 structure. 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.

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

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 checkerbox structure but contains a small numbers of sub-tensors of elevated means, we develop a 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 [6] and Tucker decomposition [7]. 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 has yet to be studied for higher-order tensors. Note that a block structure automatically implies low-rankness. However, as we will show 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 [8] and its higher-order extensions [9, 10]. Existing multiway clustering methods [9, 10, 5, 11] 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. This approach achieves a higher accuracy and an improved interpretability. Moreover, earlier solutions to multiway clustering [12, 9] focus on the algorithm effectiveness, leaving the statistical optimality of the estimators unaddressed. Very recently, Chi et al [13] provides an attempt to study the statistical properties of the tensor block model. We will show that our estimator obtains a faster convergence rate than theirs, and the power is further boosted with a sparse regularity.

2 Preliminaries

We begin by reviewing a few basic factors about tensors [14]. 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 matrices $\boldsymbol{M}_k = [\![m^{(k)}_{i_k,j_k}]\!] \in \mathbb{R}^{s_k\times d_k}$ 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. An order-(K-1) slice of $\mathcal{Y}$ is a sub-tensor of $\mathcal{Y}$ obtained by holding the index in one mode fixed while letting other indices vary.

A clustering of d objects is 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.

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 (TBM) 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 that 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 [15], in that each tensor entry belongs to exactly one block with a common mean. The TBM 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 $\mathcal{C} = [\![c_{r_1,\ldots,r_K}]\!] \in \mathbb{R}^{R_1\times\cdots\times R_K}$ is a core tensor consisting of block means, $\boldsymbol{M}_k \in \{0,1\}^{d_k\times R_k}$ is a membership matrix indicating the block allocations along mode k for $k \in [K]$, and $\mathcal{E} = [\![\varepsilon_{i_1,\ldots,i_K}]\!]$ is the noise tensor. We view the TBM (2) 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 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) incorporates common situations such as Gaussian noise, Bernoulli noise, and noise with bounded support. In particular, we consider two important examples of the TBM:

Example 1 (Gaussian tensor block model) *Let $\mathcal{Y}$ be a continuous-valued tensor. The Gaussian tensor block model (GTBM) $y_{i_1,\ldots,i_K} \sim_{i.i.d.} N(c_{r_1,\ldots,r_K}, \sigma^2)$ 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 [12, 4, 13].*

Example 2 (Stochastic tensor block model) *Let $\mathcal{Y}$ be a binary-valued tensor. The stochastic tensor block model (STBM) $y_{i_1,\ldots,i_K} \sim_{i.i.d.}$ 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 [16, 17] for matrix-based network analysis. In the filed of community detection, multi-layer stochastic model has also been developed for multi-relational network data analysis [18, 19].*

More generally, our model also applied to hybrid error distributions, in which different types of distribution are 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 TBM.

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) slices of $\mathcal{C}$ are identical. Note that irreducibility is a weaker assumption than full-rankness.

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

The identifiability property for the TBM outperforms that for the classical factor model [20, 21]. In the Tucker [22, 14] and many other factor analyses [20, 21], the factors are identifiable only up to orthogonal rotations. Those models 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 individual factor is consistently estimated in high dimensions. This brings a benefit to the interpretation of factors in the tensor block model.

We propose a least-square approach for estimating the TBM. 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

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

In the following theoretical analysis, we assume 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 the TBM (1) is

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

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

4 Statistical convergence

In this section, we establish the convergence rate of the least-squares estimator (4) for two measurements. The first measurement is 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 tensors, respectively. While the loss function corresponds to the likelihood for the Gaussian tensor model, the same assertion does not hold for other types of distribution such as stochastic tensor block model. We will show that, with very high probability, a simple least-square estimator achieves a fast convergence rate in a general class of block tensor models.

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

$$MSE(\Theta_{true}, \hat{\Theta}) \le \frac{C_1\sigma^2}{\prod_k d_k}\left(\prod_k R_k + \sum_k d_k \log R_k\right) \tag{5}$$

holds with probability at least $1 - \exp(-C_2(\prod_k R_k + \sum_k d_k \log R_k))$ uniformly over $\Theta_{true} \in \mathcal{P}$ and all error distribution satisfying (3).

The convergence rate of MSE 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 $\sum_k d_k R_k'$ [22], where R_k' is the multilinear rank in the mode k. Applying Tucker decomposition to the TBM yields $\sum_k d_k R_k$, because the mode-k rank is bounded by the number of mode-k clusters. 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, [13] proposed a convex relaxation for estimating the TBM. 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 $\mathcal{O}(d^{-1})$ for all $K \ge 2$. As we see from (5), our estimate obtains a much better convergence rate $\mathcal{O}(d^{-(K-1)})$, which is especially favorable as the order increases.

The bound (5) generalizes the previous results on structured matrix estimation in network analysis [23, 16]. Earlier work [16] 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 $K \ge 2$. Note that, in the TBM, the sample size is the total number of entries $\prod_k d_k$, the number of parameters is $\prod_k R_k$, and the combinatoric complexity for estimating block structure is of order $\prod_k R_k^{d_k}$.

Next we study the consistency of partition. To define the misclassification rate (MCR), we need to introduce some additional notation. Let $\boldsymbol{M}_k = [\![m_{i,r}^{(k)}]\!], \hat{\boldsymbol{M}}_k = [\![\hat{m}_{i,r'}^{(k)}]\!]$ be two mode-k membership matrices, and $\boldsymbol{D}^{(k)} = [\![D_{r,r'}^{(k)}]\!]$ be the mode-k confusion matrix with element $D_{r,r'}^{(k)} = \frac{1}{d_k}\sum_{i=1}^{d_k} \mathbb{1}\{m_{i,r}^{(k)} = \hat{m}_{i,r'}^{(k)} = 1\}$, where $r, r' \in [R_k]$. Note that the row/column sum of $\boldsymbol{D}^{(k)}$ represents the nodes proportion in each cluster defined by $\boldsymbol{M}_k$ or $\hat{\boldsymbol{M}}_k$. We restrict ourselves to non-degenerating clusterings; that is, the row/column sums of $\boldsymbol{D}^{(k)}$ are lowered bounded by $\tau > 0$. With a little abuse of notation, we still use $\mathcal{P} = \mathcal{P}(\tau)$ to denote the parameter space with the non-degenerating assumption. The least-square estimator (4) should also be interpreted with this constraint imposed.

We define the mode-k misclassification rate (MCR) as

$$\mathrm{MCR}(\boldsymbol{M}_k, \hat{\boldsymbol{M}}_k) = \max_{r\in[R_k], a\neq a'\in[R_k]} \min\left\{D_{a,r}^{(k)},\ D_{a',r}^{(k)}\right\}.$$

In other words, MCR is the element-wise maximum of the confusion matrix after removing the largest entry from each column. Under the non-degenerating assumption, MCR $= 0$ if and only if the confusion matrix $\boldsymbol{D}^{(k)}$ is a permutation of a diagonal matrix; that is, the estimated partition matches with the true partition, up to permutations of cluster labels.

Theorem 2 (Convergence rate of MCR) *Consider a tensor block model (2) with sub-Gaussian parameter σ. Define the minimal gap between the blocks $\delta_{\min} = \min_k \delta^{(k)}$, where $\delta^{(k)} = \min_{r_k\neq r_k'} \max_{r_1,\ldots,r_{k-1},r_{k+1},\ldots,r_K} (c_{r_1,\ldots,r_k,\ldots,r_K} - c_{r_1,\ldots,r_k',\ldots,r_K})^2$. Let $\boldsymbol{M}_{k,true}$ be the true mode-k membership, $\hat{\boldsymbol{M}}_k$ be the estimator from (4). Then, for any $\varepsilon \in [0,1]$,*

$$\mathbb{P}(MCR(\hat{\boldsymbol{M}}_k, \boldsymbol{M}_{k,true}) \geq \varepsilon) \leq 2^{1+\sum_k d_k} \exp\left(-\frac{C\varepsilon^2\delta_{\min}^2\tau^{3K-2}\prod_{k=1}^K d_k}{\sigma^2}\right),$$

where $C > 0$ is a positive constant, and $\tau > 0$ the lower bound of cluster proportions.

The above theorem shows that our estimator consistently recovers the block structure as the dimension of the data tensor grows. The block-mean gap $\delta_{\min}$ serves the role of the eigen-separation as in the classical tensor Tucker decomposition [22]. Table 1 summarizes the comparison of various tensor methods in the special case when $d_1 = \cdots = d_K = d$ and $R_1 = \cdots = R_K = R$.

Method	Recovery error (MSE)	Clustering error (MCR)	Block detection (see Section 6)
Tucker [22]	dR	-	No
CoCo [13]	d^{K-1}	-	No
TBM (this paper)	$d\log R$	$\frac{\sigma}{\delta_{\min}} d^{-(K-1)/2}$	Yes

Table 1: Comparison of various tensor decomposition methods.

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 $\boldsymbol{M}_k$'s. The optimization (4) 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 consist 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 others fixed. Specifically, given the collection of $\hat{\boldsymbol{M}}_k$'s, the core tensor estimate $\hat{\mathcal{C}} = \arg\min_{\mathcal{C}} f(\mathcal{C}, \{\hat{\boldsymbol{M}}_k\})$ 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 $\hat{\boldsymbol{M}}_k$'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 $(r_1, \ldots, r_K) \in [R_1] \times \cdots [R_K]$,

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

where $\boldsymbol{M}_k^{-1}(r_k)$ denotes the indices that belong to the r_kth cluster in the mode k, and $n_{r_1,\ldots,r_K} = \prod_k |\hat{\boldsymbol{M}}_k^{-1}(r_k)|$ 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 $\hat{\boldsymbol{M}}_k$. Specifically, for each $a \in [d_k]$, assign the cluster label $\hat{\boldsymbol{M}}_k(a) \in [R_k]$:

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

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. The per-iteration computational cost scales linearly with the sample size, $d = \prod_k d_k$, and this complexity matches the classical tensor methods [24, 25, 22]. We recognize that obtaining the global optimizer for such a non-convex optimization is typically difficult [26, 1]. Following the common practice in non-convex optimization [1], we run the algorithm multiple times, using random initializations with independent one-way k-means on each of the modes.

5.2 Tuning parameter selection

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),

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

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 $\mathrm{BIC}(\boldsymbol{R})$ 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 sparse estimation

In some large-scale applications, not every block in a data tensor is of equal importance. For example, in the genome-wide 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) is still able to 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 a sparse version of TBM by imposing regularity on 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:

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

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 Lemma 2 in the Supplements):

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

where $a_+ = \max(a, 0)$ and $\hat{c}^{\text{ols}}_{r_1,\ldots,r_K}$ denotes the ordinary least-square estimate in (7). The choice of penalty 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 TBM method. We consider both non-sparse and sparse tensors, and compare the recovery accuracy with other tensor-based methods. Unless otherwise stated, we generate Gaussian tensors under the block model (1). The block means are generated from i.i.d. Uniform[-3,3]. The entries in the noise tensor $\mathcal{E}$ are generated from i.i.d. $N(0, \sigma^2)$. 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 tensors of order 3 and order 4 (see Figure 2). In the case of order-3 tensors, 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 a convergence rate $\mathcal{O}(\sqrt{\log R_1 / d_2 d_3})$ for our estimator. Figure 2a plots the recovery error versus the rescaled sample size $N_1 = \sqrt{d_2 d_3 / \log R_1}$. We find that the RMSE decreases roughly at the rate of $1/N_1$. This is consistent with 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. Similar behavior can be observed in the order-4 case from Figure 2b, where the rescaled sample size is $N_2 = \sqrt{d_2 d_3 d_4 / \log R_1}$.

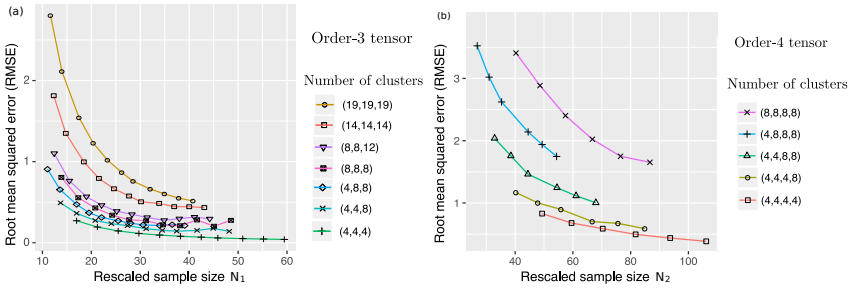

Figure 2: Estimation error for block tensors with Gaussian noise. Each curve corresponds to a fixed clustering size $\boldsymbol{R}$. (a) Average RMSE against rescaled sample size $N_1 = \sqrt{d_2 d_3 / \log R_1}$ for order-3 tensors. (b) Average RMSE against rescaled sample size $N_2 = \sqrt{d_2 d_3 d_4 / \log R_1}$ for order-4 tensors.

In the second experiment, we evaluate the selection performance of our BIC criterion (8). Supplementary Table S1 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 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 TBM method with two popular low-rank tensor estimation methods: (i) CP decomposition and (ii) Tucker decomposition. Following the literature [13, 5, 9], 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.

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 3a shows that TBM achieves the lowest estimation error among the three methods. The gain in accuracy is more pronounced as the noise grows. Neither CP nor Tucker recovers the signal tensor, although Tucker appears to result in a modest clustering performance (Figure 3b). One possible explanation is that the Tucker model imposes orthogonality to the factors, which make the subsequent k-means clustering easier than that for the CP factors. Figure 3b-c shows that the clustering error increases with noise but decreases with dimension. This agrees with our expectation, as in tensor data analysis, a larger dimension implies a larger sample size.

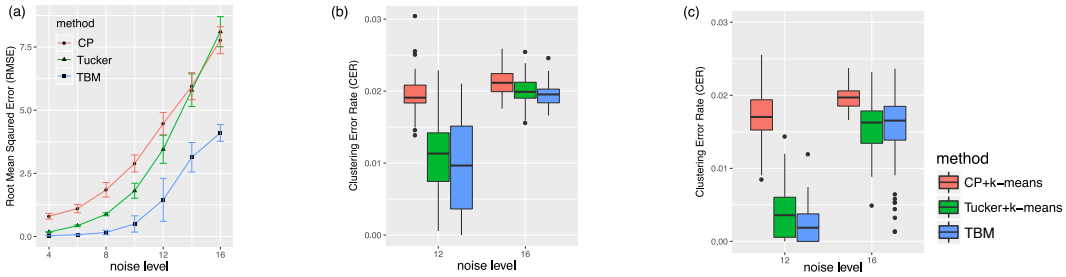

Figure 3: Performance comparison in terms of RMSE and CER. (a) Estimation error against noise for tensors of dimension $(40, 40, 40)$. (b) Clustering error against noise for tensors of dimension $(40, 40, 40)$. (c) Clustering error against noise for tensors of dimension $(40, 50, 60)$.

Sparse case. We then evaluate 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 Uniform[-3,3], with probability p (sparsity rate) and $1 - p$ respectively. We generate noisy tensors of dimension $\boldsymbol{d} = (40, 40, 40)$ with varying levels of sparsity and noise. We utilize $\ell 0$-penalized TBM and primarily focus on the selection accuracy. The performance 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).

Table 2 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. Supplementary Figure S1 shows the estimation error and sparsity error against σ when $p = 0.8$. Again, the sparse TBM outperforms the other methods.

Sparsity (p)	Noise (σ)	BIC-selected λ	Estimated Sparsity Rate	Correct Zero Rate	Sparsity Error Rate
0.5	4	136.0(37.5)	**0.55(0.04)**	**1.00(0.02)**	**0.06(0.03)**
0.5	8	439.2(80.2)	**0.58(0.06)**	**0.94(0.08)**	0.15(0.07)
0.8	8	458.0(63.3)	**0.81(0.15)**	**0.87(0.16)**	**0.21(0.13)**

Table 2: Sparse TBM for estimating tensors of dimension $\boldsymbol{d} = (40, 40, 40)$. The reported statistics are averaged across 50 simulations with standard deviation given in parentheses. Number in bold indicates the ground truth is within 2 standard deviations of the sample average.

7.3 Real data analysis

Lastly, we apply our method on two real datasets. The first dataset is a real-valued tensor, consisting of approximate 1 million expression values from 13 brain tissues, 193 individuals, and 362 genes [4]. 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 a clear tissues $\times$ genes specificity (Supplementary Table S2). In particular, the top over-expressed block is driven by tissues {*Substantia nigra, Spinal cord*} and genes {*GFAP, MBP*}, suggesting their elevated expression across 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 [27]. Our method also identifies blocks with extremely negative means (i.e. under-expressed blocks). The top under-expressed block is driven by tissues {*Cerebellum, Cerebellar Hemisphere*} and genes {*CDH9, GPR6, RXFP1, CRH, DLX5/6, NKX2-1, SLC17A8*}. The gene *DLX6* encodes proteins in the forebrain development [27], whereas cerebellum tissues are located in the hindbrain brain. The opposite spatial function is consistent with the observed under-expression pattern.

The second dataset we consider is the *Nations* data [2]. 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. Other top blocks are described in the Supplement.

We compared the goodness-of-fit of various clustering methods on the *Brain expression* and *Nations* datasets. Because the code of CoCo method [13] is not yet available, we excluded it from our numerical comparison (See Section 4 for the theoretical comparison with CoCo). The Table 3 summarizes the proportion of variance explained by each clustering method:

Dataset	TBM	TBM-sparse	CP+k-means	Tucker+k-means	CoTeC [12]
Brain expression	0.856	0.855	0.576	0.434	0.849
Nations	0.439	0.433	0.324	0.253	0.419

Table 3: Comparison of goodness-of-fit in the *Brain* expression and *Nations* datasets.

Our method (TBM) achieves the highest variance proportion, suggesting that the entries within the same cluster are close (i.e., a good clustering). As expected, the sparse TBM results in a slightly lower proportion, because it has a lower model complexity at the cost of small bias. It is remarkable that the sparse TBM still achieves a higher goodness-of-fit than others. The improved interpretability with little loss of accuracy makes the sparse TBM appealing in applications.

8 Conclusion

We have developed a statistical setting for studying the tensor block model. Under the assumption that tensor entries are distributed with a block-specific mean, our estimator achieves a convergence rate $\mathcal{O}(\sum_k d_k \log R_k)$ which is faster than previously possible. Our TBM method applies to a broad range of data distributions and can handle both sparse and sense data tensor. We demonstrate the benefit of sparse regularity in power of detection. In specific applications, prior knowledge may suggest other regularities for parameters. For example, in the multi-layer network analysis, sometimes it may be reasonable to impose symmetry on the parameters along certain modes. In some other applications, non-negativity of parameter values may be enforced. We leave these directions for future study.

Acknowledgements

This research was supported by NSF grant DMS-1915978 and the University of Wisconsin-Madison, Office of the Vice Chancellor for Research and Graduate Education with funding from the Wisconsin Alumni Research Foundation.

References

[1] 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.

[2] 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.

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

[4] 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.

[5] 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.

[6] 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.

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

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

[9] 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.

[10] 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.

[11] Miaoyan Wang and Lexin Li. Learning from binary multiway data: Probabilistic tensor decomposition and its statistical optimality. *arXiv preprint arXiv:1811.05076*, 2018.

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

[13] 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.

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

[15] 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.

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

[17] Peter J Bickel and Aiyou Chen. A nonparametric view of network models and newman–girvan and other modularities. *Proceedings of the National Academy of Sciences*, 106(50):21068–21073, 2009.

[18] Kehui Chen Jing Lei and Brian Lynch. Consistent community detection in multi-layer network data. *Biometrika*, to appear, 2019.

[19] Subhadeep Paul, Yuguo Chen, et al. Consistent community detection in multi-relational data through restricted multi-layer stochastic blockmodel. *Electronic Journal of Statistics*, 10(2):3807–3870, 2016.

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

[23] 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.

[24] Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *The Journal of Machine Learning Research*, 15(1):2773–2832, 2014.

[25] Miaoyan Wang and Yun Song. Tensor decompositions via two-mode higher-order SVD (HOSVD). In *Artificial Intelligence and Statistics*, pages 614–622, 2017.

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

[27] Nuala A O'Leary, Mathew W Wright, J Rodney Brister, Stacy Ciufo, Diana Haddad, Rich McVeigh, Bhanu Rajput, Barbara Robbertse, Brian Smith-White, Danso Ako-Adjei, et al. Reference sequence (refseq) database at ncbi: current status, taxonomic expansion, and functional annotation. *Nucleic acids research*, 44(D1):D733–D745, 2015.

Supplements for "Multiway clustering via tensor block models"

A Proofs

A.1 Stochastic tensor block model

The following property shows that Bernoulli distribution belongs to the sub-Gaussian family with a subgaussianity parameter σ equal to $1/4$.

Property 1. *Suppose $x \sim$ Bernoulli(p), then $x \sim$ sub-Gaussian$(\frac{1}{4})$.*

Proof. For all $\lambda \in \mathbb{R}$, we have

$$\ln(\mathbb{E}(e^{\lambda(x-p)}) = \ln\left(pe^{\lambda(1-p)} + (1-p)e^{-p\lambda}\right) = -p\lambda + \ln(1 + pe^{\lambda} - p) \leq \frac{\lambda^2}{8}.$$

Therefore $\mathbb{E}(e^{\lambda(x-p)}) \leq e^{\lambda^2(1/4)/2}$. □

A.2 Proof of Proposition 1

Proof. Let $\mathbb{P}_\Theta$ denotes the (either Gaussian or Bernoulli) tensor block model, where $\Theta = \mathcal{C} \times_1 \boldsymbol{M}_1 \times_2 \cdots \times_K \boldsymbol{M}_K$ parameterizes the mean tensor. Since the mapping $\Theta \mapsto \mathbb{P}_\Theta$ is one-to-one, Θ is identifiable. Now suppose that Θ can be decomposed in two ways, $\Theta = \Theta(\{\boldsymbol{M}_k\}, \mathcal{C}) = \Theta(\{\tilde{\boldsymbol{M}}_k\}, \tilde{\mathcal{C}})$. Based on the Assumption 1, we have

$$\Theta = \mathcal{C} \times_1 \boldsymbol{M}_1 \times_2 \cdots \times_K \boldsymbol{M}_K = \tilde{\mathcal{C}} \times_1 \tilde{\boldsymbol{M}}_1 \times_2 \cdots \times_K \tilde{\boldsymbol{M}}_K, \tag{1}$$

where $\mathcal{C}, \tilde{\mathcal{C}} \in \mathbb{R}^{R_1 \times \cdots \times R_K}$ are two irreducible cores, and $\boldsymbol{M}_k, \tilde{\boldsymbol{M}}_k \in \{0,1\}^{R_k \times d_k}$ are membership matrices for all $k \in [K]$. We will prove by contradiction that $\boldsymbol{M}_k$ and $\tilde{\boldsymbol{M}}_k$ induce the same partition of $[d_k]$, for all $k \in [K]$.

Suppose the above claim does not hold. Then there exists a mode $k \in [K]$ such that the $\boldsymbol{M}_k, \tilde{\boldsymbol{M}}_k$ induce two different partitions of $[d_k]$. Without loss of generality, we assume $k = 1$. The definition of partition implies that there exists a pair of indices $i \neq j$, $i, j \in [d_1]$, such that, i, j belong to the same cluster based on $\boldsymbol{M}_1$, but they belong to different clusters based on $\tilde{\boldsymbol{M}}_1$. Let $\mathcal{A} \neq \mathcal{B}, \mathcal{A}, \mathcal{B} \subset [d_1]$ respectively denote the clusters that i and j belong to, based on $\tilde{\boldsymbol{M}}_1$. The left-hand side of (1) implies

$$\Theta_{i,i_2,\ldots,i_K} = \Theta_{j,i_2,\ldots,i_K}, \quad \text{for all } (i_2,\ldots,i_K) \in [d_2] \times \cdots \times [d_K]. \tag{2}$$

On the other hand, (1) implies

$$\Theta_{i,i_2,\ldots,i_K} = \Theta_{k,i_2,\ldots,i_K}, \quad \text{for all } k \in \mathcal{A} \text{ and all } (i_2,\ldots,i_K) \in [d_2] \times \cdots \times [d_K], \tag{3}$$

and

$$\Theta_{j,i_2,\ldots,i_K} = \Theta_{k,i_2,\ldots,i_K}, \quad \text{for all } k \in \mathcal{B} \text{ and all } (i_2,\ldots,i_K) \in [d_2] \times \cdots \times [d_K]. \tag{4}$$

Combining (2), (3) and (4), we have

$$\Theta_{i,i_2,\ldots,i_K} = \Theta_{k,i_2,\ldots,i_K}, \quad \text{for all } k \in \mathcal{A} \cup \mathcal{B} \text{ and all } (i_2,\ldots,i_K) \in [d_2] \times \cdots \times [d_K]. \tag{5}$$

Equation (5) implies that $\mathcal{A}$ and $\mathcal{B}$ can be merged into one cluster. This contradicts the irreducibility assumption of the core tensor $\tilde{\mathcal{C}}$. Therefore, $\boldsymbol{M}_1$ and $\tilde{\boldsymbol{M}}_1$ induce a same partition of $[d_1]$, and thus they are equal up to permutation of cluster labels. The proof is now complete. □

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

A.3 Proof of Theorem 1

The following lemma is useful for the proof of Theorem 1.

Lemma 1. *Suppose* $\mathcal{Y} = \Theta_{true} + \mathcal{E}$ *with* $\Theta_{true} \in \mathcal{P}$. *Let* $\hat{\Theta} = \arg\min_{\Theta \in \mathcal{P}} \|\hat{\Theta} - \mathcal{Y}\|_F^2$ *be the least-square estimator of* Θ_{true}. *We have*

$$\|\hat{\Theta} - \Theta_{true}\|_F \leq 2 \sup_{\mu \in \frac{\mathcal{P}-\mathcal{P}'}{|\mathcal{P}-\mathcal{P}'|}} \langle \mu, \mathcal{E} \rangle,$$

where $\mathcal{P} - \mathcal{P}' = \{\Theta - \Theta' \colon \Theta, \Theta' \in \mathcal{P}\}$ and $\mathcal{S}/|\mathcal{S}| = \{s/\|s\|_2 \colon s \in \mathcal{S}\}$.

Proof. Based on the definition of least-square estimator, we have

$$\|\hat{\Theta} - \mathcal{Y}\|_F^2 \leq \|\Theta_{\text{true}} - \mathcal{Y}\|_F^2. \tag{6}$$

Combining (6) with the fact

$$\begin{aligned}\|\hat{\Theta} - \mathcal{Y}\|_F^2 &= \|\hat{\Theta} - \Theta_{\text{true}} + \Theta_{\text{true}} - \mathcal{Y}\|_F^2 \\ &= \|\hat{\Theta} - \Theta_{\text{true}}\|_F^2 + \|\Theta_{\text{true}} - \mathcal{Y}\|_F^2 + 2\langle \hat{\Theta} - \Theta_{\text{true}}, \Theta_{\text{true}} - \mathcal{Y} \rangle,\end{aligned}$$

yields

$$\|\hat{\Theta} - \Theta_{\text{true}}\|_F^2 \leq 2\langle \hat{\Theta} - \Theta_{\text{true}}, \mathcal{Y} - \Theta_{\text{true}} \rangle = 2\langle \hat{\Theta} - \Theta_{\text{true}}, \mathcal{E} \rangle.$$

Dividing each side by $\|\hat{\Theta} - \Theta_{\text{true}}\|_F$, we have

$$\|\hat{\Theta} - \Theta_{\text{true}}\|_F \leq 2 \left\langle \frac{\hat{\Theta} - \Theta_{\text{true}}}{\|\hat{\Theta} - \Theta_{\text{true}}\|_F}, \mathcal{E} \right\rangle.$$

The desired inequality follows by noting $\frac{\hat{\Theta} - \Theta_{\text{true}}}{\|\hat{\Theta} - \Theta_{\text{true}}\|_F} \in \frac{\mathcal{P}-\mathcal{P}'}{|\mathcal{P}-\mathcal{P}'|}$. □

Proof of Theorem 1. To study the performance of the least-square estimator $\hat{\Theta}$, we need to introduce some additional notation. We view the membership matrix $\boldsymbol{M}_k$ as an onto function $\boldsymbol{M}_k \colon [d_k] \mapsto [R_k]$. With a little abuse of notation, we still use $\boldsymbol{M}_k$ to denote the mapping function and write $\boldsymbol{M}_k \in R_k^{d_k}$ by convention. We use $\boldsymbol{M} = \{\boldsymbol{M}_k\}_{k \in [K]}$ to denote the collection of K membership matrices, and write $\mathcal{M} = \{\boldsymbol{M} \colon \boldsymbol{M}$ is the collection of membership matrices $\boldsymbol{M}_k$'s$\}$. For any set J, $|J|$ denotes its cardinality. Note that $|\mathcal{M}| \leq \prod_k R_k^{d_k}$, because each $\boldsymbol{M}_k$ can be identified by a partition of $[d_k]$ into R_k disjoint non-empty sets.

For ease of notation, we define $d = \prod_k d_k$ and $R = \prod_k R_k$. We sometimes identify a tensor in $\mathbb{R}^{d_1 \times \cdots \times d_K}$ with a vector in $\mathbb{R}^d$. By the definition of the parameter space $\mathcal{P}$, the element $\Theta \in \mathcal{P}$ can be equivalently identified by $\Theta = \Theta(\boldsymbol{M}, \boldsymbol{C})$, where $\boldsymbol{M} \in \mathcal{M}$ is the collection of K membership matrices and $\boldsymbol{C} = \text{vec}(\mathcal{C}) \in \mathbb{R}^R$ is the core tensor. Note that, for a fixed clustering structure $\boldsymbol{M}$, the space consisting of $\Theta = \Theta(\boldsymbol{M}, \cdot)$ is a linear space of dimension R.

Now consider the least-square estimator

$$\hat{\Theta} = \arg\min_{\Theta \in \mathcal{P}} \{-2\langle \mathcal{Y}, \Theta \rangle + \|\Theta\|_F^2\} = \arg\min_{\Theta \in \mathcal{P}} \{\|\mathcal{Y} - \Theta\|_F^2\}.$$

Based on the Lemma 1,

$$\begin{aligned}\|\hat{\Theta} - \Theta_{\text{true}}\|_F &\leq 2 \sup_{\Theta \in \mathcal{P}} \sup_{\Theta' \in \mathcal{P}} \left\langle \frac{\Theta - \Theta'}{\|\Theta - \Theta'\|_F}, \mathcal{E} \right\rangle \\ &\leq 2 \sup_{\boldsymbol{M}, \boldsymbol{M}' \in \mathcal{M}} \sup_{\boldsymbol{C}, \boldsymbol{C}' \in \mathbb{R}^R} \left\langle \frac{\Theta(\boldsymbol{M}, \boldsymbol{C}) - \Theta'(\boldsymbol{M}', \boldsymbol{C}')}{\|\Theta(\boldsymbol{M}, \boldsymbol{C}) - \Theta'(\boldsymbol{M}', \boldsymbol{C}')\|_F}, \mathcal{E} \right\rangle.\end{aligned}$$

By union bound, we have, for any $t > 0$,

$$\begin{aligned}
\mathbb{P}\left(\|\hat{\Theta} - \Theta_{\text{true}}\|_F > t\right) &\le \mathbb{P}\left(\sup_{\boldsymbol{M},\boldsymbol{M}'\in\mathcal{M}} \sup_{\boldsymbol{C},\boldsymbol{C}'\in\mathbb{R}^R} \left|\left\langle \frac{\Theta(\boldsymbol{M},\boldsymbol{C}) - \Theta'(\boldsymbol{M}',\boldsymbol{C}')}{\|\Theta(\boldsymbol{M},\boldsymbol{C}) - \Theta'(\boldsymbol{M}',\boldsymbol{C}')\|_F}, \mathcal{E}\right\rangle\right| > \frac{t}{2}\right) \\
&\le \sum_{\boldsymbol{M},\boldsymbol{M}'\in\mathcal{M}} \mathbb{P}\left(\sup_{\boldsymbol{C}'\in\mathbb{R}^R} \sup_{\boldsymbol{C}\in\mathbb{R}^R} \left|\left\langle \frac{\Theta(\boldsymbol{M},\mathcal{C}) - \Theta'(\boldsymbol{M}',\mathcal{C})}{\|\Theta(\boldsymbol{M},\mathcal{C}) - \Theta'(\boldsymbol{M}',\mathcal{C})\|_F}, \mathcal{E}\right\rangle\right| \ge \frac{t}{2}\right) \\
&\le |\mathcal{M}|^2 C_1^R \exp\left(-\frac{C_2 t^2}{32\sigma^2}\right) \\
&= \exp\left(2\sum_k d_k \log R_k + C_1 \prod_k R_k - \frac{C_2 t^2}{32\sigma^2}\right),
\end{aligned}$$

for two universal constants $C_1, C_2 > 0$. Here the third line follows from [1] (Theorem 1.19) and the fact that $\Theta = \Theta(\boldsymbol{M}, \cdot)$ lies in a linear space of dimension R. The last line uses $|\mathcal{M}| \le \prod_k R_k^{d_k}$ and $R = \prod_k R_k$. Choosing $t = C\sigma\sqrt{\prod_k R_k + \sum_k d_k \log R_k}$ yields the desired bound. □

A.4 Proof of Theorem 2

First we give a list of notation used in the proof. For ease of notation, we allow the basic arithmetic operators ($+, -, \ge$, etc) to be applied to pairs of vectors in an element-wise manner.

A.4.1 Notations

$\boldsymbol{M}_k = [\![m_{ir}^{(k)}]\!] \in \{0,1\}^{d_k \times R_k}$: the mode-$k$ membership matrix. The element $m_{ir}^{(k)} = 1$ if and only if the ith slide in mode k belongs to the rth cluster.

$\boldsymbol{M}_{k,\text{true}}$, $\hat{\boldsymbol{M}}_k \in \{0,1\}^{d_k \times R_k}$: the true and estimated mode-k cluster membership matrices, respectively.

$\boldsymbol{p}^{(k)} = [\![p_r^{(k)}]\!] \in [0,1]^{R_k}$: the marginal cluster proportion vector listing the relative cluster sizes along the mode k. The element $p_r^{(k)} = \frac{1}{d_k}\sum_{i=1}^{d_k} \mathbb{1}\{m_{ir}^{(k)} = 1\}$ denotes the proportion of the rth cluster. The cluster proportion vector $\boldsymbol{p}^{(k)} = \boldsymbol{p}^{(k)}(\boldsymbol{M}_k)$ can be viewed as a function of $\boldsymbol{M}_k$.

$\boldsymbol{p}_{\text{true}}^{(k)}$, $\hat{\boldsymbol{p}}^{(k)} \in [0,1]^{R_k}$: the true and estimated mode-k cluster proportion vectors, respectively.

$\boldsymbol{D}^{(k)} = [\![D_{rr'}^{(k)}]\!] \in [0,1]^{R_k \times R_k}$: the mode-$k$ confusion matrix between clustering $\boldsymbol{M}_{k,\text{true}}$ and $\hat{\boldsymbol{M}}_k$. The entries in the confusion matrix is $D_{rr'}^{(k)} = \frac{1}{d_k}\sum_{i=1}^{d_k} \mathbb{I}\{m_{ir,\text{true}}^{(k)} = \hat{m}_{ir'}^{(k)} = 1\}$. The confusion matrix $\boldsymbol{D}^{(k)} = \boldsymbol{M}_{k,\text{true}}^T \hat{\boldsymbol{M}}_k$ is a function of $\boldsymbol{M}_{k,\text{true}}$ and $\hat{\boldsymbol{M}}_k$.

$\mathcal{J}_\tau = \{(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K) : \boldsymbol{p}^{(k)}(\boldsymbol{M}_k) \ge \tau \text{ for all } k \in [K]\}$: the set of all possible partitions that satisfy the marginal non-degenerating assumption.

$\mathcal{I} \subset 2^{[d_1]} \times \cdots \times 2^{[d_K]}$: the set of blocks that satisfy the marginal non-degenerating assumption for all $k \in [K]$;

$L = \inf\{|I| : I \in \mathcal{I}\}$: the minimum block size in $\mathcal{I}$.

$\|\mathcal{A}\|_\infty = \max_{r_1,\ldots,r_K} |a_{r_1,\ldots,r_K}|$ for any tensor $\mathcal{A} = [\![a_{i_1,\ldots,i_K}]\!] \in \mathbb{R}^{R_1 \times \ldots \times R_K}$.

$f(x) = x^2$: the quadratic objective function.

Remark 1. *By definition, the confusion matrix $\boldsymbol{D}^{(k)}$ satisfies the following two properties:*

1. $\boldsymbol{D}^{(k)}\mathbf{1} = \boldsymbol{p}_{true}^{(k)}$, $(\boldsymbol{D}^{(k)})^T\mathbf{1} = \hat{\boldsymbol{p}}^{(k)}$.
2. *The estimated clustering matches the true clustering if and only if $\boldsymbol{D}^{(k)}$ equals to the diagonal matrix up to permutation.*

We view the membership matrix $\boldsymbol{M}_k$ as an onto function $\boldsymbol{M}_k : [d_k] \mapsto [R_k]$. With a little abuse of notation, we still use $\boldsymbol{M}_k$ to denote the mapping function. In this context, $\boldsymbol{M}_k(i)$ represents the cluster that the index i belongs to, and $\boldsymbol{M}_k^{-1}(r)$ represents the set of indices that belong to cluster r.

A.4.2 Auxiliary Results

Recall that the objective function in our tensor block model is

$$f(\mathcal{C}, \{\boldsymbol{M}_k\}) = \langle \mathcal{Y}, \Theta \rangle - \frac{\|\Theta\|_F^2}{2}, \quad \text{where } \Theta = \mathcal{C} \times_1 \boldsymbol{M}_1 \times_2 \cdots \times_K \boldsymbol{M}_K, \tag{7}$$

where $\mathcal{Y} \in \mathbb{R}^{d_1 \times \cdots \times d_K}$ is the data, $\mathcal{C}$ is the core tensor of interest, and $\{\boldsymbol{M}_k\}$ is the membership matrices of interest. Without loss of generality, we will work with the scaled objective $\frac{1}{2\prod_k d_k} f(\mathcal{C}, \{\boldsymbol{M}_k\})$. With a little abuse of notation, we still denote the scaled function as $f(C, \{\boldsymbol{M}_k\})$.

We will prove that, if there is non-negligible mismatch between $\{\hat{\boldsymbol{M}}_k\}$ and $\{\boldsymbol{M}_{k,\text{true}}\}$, then $\{\hat{\boldsymbol{M}}_k\}$ cannot be the optimizer to (7). To show this, we investigate the objective values at the global optimizer vs. at the true parameter. The deviation between these two values comes from two aspects: the label assignments (i.e., the estimation of $\{\boldsymbol{M}_k\}$) and the estimation of the core tensor. In what follows, we tease apart these two aspects.

1. First, suppose the partitions $\{\boldsymbol{M}_k\}$ are given, which are not necessarily equal to $\{\boldsymbol{M}_{k,\text{true}}\}$. We now assess the stochastic error due to estimation of $\mathcal{C}$, conditional on $\{\boldsymbol{M}_k\}$. In such a case, the core $\hat{\mathcal{C}} = \arg\min_{\mathcal{C}} f(\mathcal{C}, \{\boldsymbol{M}_k\})$ can be solved explicitly. Specifically, the optimizer $\hat{\mathcal{C}} = [\![\hat{c}_{r_1,\ldots,r_K}]\!]$ consists of the sample averages of each tensor block, where

$$\begin{aligned} \hat{c}_{r_1,\ldots,r_K} &= \hat{c}_{r_1,\ldots,r_K}(\{\boldsymbol{M}_k\}) \\ &= \frac{1}{p_{r_1}^{(1)} \cdots p_{r_K}^{(K)}} \left[\mathcal{Y} \times_1 \boldsymbol{M}_1^T \times_2 \cdots \times_K \boldsymbol{M}_K^T\right]_{r_1,\ldots,r_K} \end{aligned} \tag{8}$$

   where the marginal cluster proportion $p_{r_k}^{(k)}$ is induced by the clustering $\boldsymbol{M}_k$.
   Define a new cost function $F(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K) = f(\hat{\mathcal{C}}, \boldsymbol{M}_1, \ldots, \boldsymbol{M}_k)$, where $\hat{\mathcal{C}} = [\![\hat{c}_{i_1,\ldots,i_K}]\!]$ is expressed in (8). A straightforward calculation shows that the function $F(\cdot)$ has the form

$$F(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K) = \sum_{r_1,\ldots,r_K} \left(\prod_k p_{r_k}^{(k)}\right) \hat{c}_{r_1,\ldots,r_K}^2. \tag{9}$$

   Let $G(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_k) = \mathbb{E}(F(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K))$, where the expectation is taken with respect to the $\mathcal{C} = [\![\hat{c}_{r_1,\ldots,r_K}]\!]$. We have that

$$G(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K) = \sum_{r_1,\ldots,r_K} \left(\prod_k p_{r_k}^{(k)}\right) \mu_{r_1,\ldots,r_K}^2, \tag{10}$$

   where

$$\mu_{r_1,\ldots,r_K} = \mathbb{E}(\hat{c}_{r_1,\ldots,r_K}) = \frac{1}{\prod_k p_{r_k}^{(k)}} \left[\mathcal{C} \times_1 \boldsymbol{D}^{(1)^T} \times_2 \cdots \times_K \boldsymbol{D}^{(K)^T}\right]_{r_1,\ldots,r_K}$$

   is the expectation of the average of $y_{i_1,\ldots,i_K}$ over the tensor block indexed by $(r_1, \ldots, r_K)$, and $\boldsymbol{D}^{(k)} = [\![D_{i_k j_k}^{(k)}]\!]$ is the confusion matrix between $\boldsymbol{M}_{k,\text{true}}$ and $\boldsymbol{M}_k$.
   The deviation $F(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K) - G(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K)$ quantifies the stochastic error caused by the core tensor estimation. We sometimes use $G(\boldsymbol{D}^{(1)}, \ldots, \boldsymbol{D}^{(K)})$ to denote $G(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_k)$ if we want to emphasize the error caused by mismatch in label assignments. Based on (9) and (10), we define a residual tensor

$$\begin{aligned} &\mathcal{R}(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K) = [\![R_{r_1,\ldots,r_K}]\!], \text{where} \\ &R_{r_1,\ldots,r_K} = \hat{c}_{r_1,\ldots,r_K} - \mu_{r_1,\ldots,r_K}, \quad \text{for all } (r_1, \ldots, r_K) \in [R_1] \times \cdots \times [R_K]. \end{aligned} \tag{11}$$

   Note that the entries in the residual tensor are i.i.d. σ-sub-Gaussian, conditional on the partition $\{\boldsymbol{M}_k\}$.

2. Next, we free $\{\boldsymbol{M}_k\}$ and quantify the stochastic error caused by clustering. Note that optimizing (7) is equivalent to optimizing (9) with respect to $\{\boldsymbol{M}_k\}$. So the least-square estimator of $\{\boldsymbol{M}_k\}$ can be expressed as

$$(\widehat{\boldsymbol{M}}_1, \ldots, \widehat{\boldsymbol{M}}_K) = \underset{(\boldsymbol{M}_1,\ldots,\boldsymbol{M}_K)\in\mathcal{J}_\tau}{\arg\max} F(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K).$$

The expectation (with respect to $\hat{\mathcal{C}}$) of the objective value at the true parameter is

$$G(\boldsymbol{M}_{1,\text{true}}, \ldots, \boldsymbol{M}_{K,\text{true}}) = \sum_{r_1,\ldots,r_K} p^{(1)}_{r_1,\text{true}} \cdots p^{(K)}_{r_K,\text{true}} c^2_{r_1,\ldots,r_K,\text{true}}$$

We use $G(\boldsymbol{D}^{(1)}, \ldots, \boldsymbol{D}^{(K)}) - G(\boldsymbol{M}_{1,\text{true}}, \ldots, \boldsymbol{M}_{K,\text{true}})$ to measure the stochastic error caused by mismatch in label assignments; and use $F(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K) - G(\boldsymbol{D}^{(1)}, \ldots, \boldsymbol{D}^{(K)})$ to measure stochastic error caused estimation of core tensors.

The following lemma shows that, if there is non-negligible mismatch between $\boldsymbol{M}_{k,\text{true}}$ and $\hat{\boldsymbol{M}}_k$, then $\hat{\boldsymbol{M}}_k$ cannot be the global optimizer to the objective function (7).

Lemma 2. *Consider partitions that satisfying $(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K) \in \mathcal{J}_\tau$, for some $\tau > 0$. Define the minimal gap between block means $\delta^{(k)} = \min_{r_k \neq r'_k} \max_{r_1,\ldots,r_{k-1},r_{k+1},\ldots,r_K} (c_{r_1,\ldots,r_k,\ldots,r_K} - c_{r_1,\ldots,r'_k,\ldots,r_K})^2 > 0$ and assume $\delta_{\min} = \min_k \delta^{(k)} > 0$. For any fixed $\varepsilon > 0$, suppose MCR$(\boldsymbol{M}_{k,true}, \hat{\boldsymbol{M}}_k) \geq \varepsilon$ for some $k \in [K]$. Then, we have*

$$G(\boldsymbol{D}^{(1)}, \ldots, \boldsymbol{D}^{(K)}) - G(\boldsymbol{M}_{1,true}, \ldots, \boldsymbol{M}_{K,true}) \leq -\frac{1}{2}\varepsilon\tau^{K-1}\delta_{\min},$$

where $\boldsymbol{D}^{(k)}$ is the confusion matrix between $\boldsymbol{M}_{k,true}$ and $\hat{\boldsymbol{M}}_k$.

Proof of Lemma 2. For ease of notation, we drop the subscript "true" and simply write $p^{(k)}_{r_k}$, $\boldsymbol{M}_k$, $\mathcal{C}$, etc. as the true parameters. The corresponding estimators are denoted as $\hat{p}^{(k)}_{r_k}$, $\hat{\boldsymbol{M}}_k$, etc. Recall that

$$G(\boldsymbol{D}^{(1)}, \ldots, \boldsymbol{D}^{(K)}) = \sum_{r_1,\ldots,r_K} \hat{p}^{(1)}_{r_1} \cdots \hat{p}^{(K)}_{r_K} \mu^2_{r_1,\ldots,r_K},$$

where $\hat{p}^{(k)}_{r_k}$ is the marginal cluster proportion induced by $\hat{\boldsymbol{M}}_k$, and $\mu_{r_1,\ldots,r_K}$ is the expected block mean induced by $\hat{\boldsymbol{M}}_k$:

$$\mu_{r_1,\ldots,r_K} = \mu_{r_1,\ldots,r_K}(\widehat{\boldsymbol{M}}_1, \ldots, \widehat{\boldsymbol{M}}_K) = \frac{1}{\prod_k \hat{p}^{(k)}_{r_k}} \left[\mathcal{C} \times_1 \boldsymbol{D}^{(1)^T} \times_2 \cdots \times_K \boldsymbol{D}^{(K)^T}\right]_{r_1,\ldots,r_K}.$$

We provide the proof for $k = 1$. The proof for other $k \in [K]$ is similar. The condition on MCR implies that, there exist some $r_1 \in [R_1]$ and some $a_1 \neq a'_1 \in [R_1]$, such that $\min\{D^{(1)}_{a_1 r_1}, D^{(1)}_{a'_1 r_1}\} \geq \varepsilon$. Because the minimal gap between tensor block means are non-zero, we choose $(a_2, \ldots, a_K)$ such that $(c_{a_1,a_2,\ldots,a_K} - c_{a'_1,a_2,\ldots,a_K})^2 = \max_{a_2,\ldots,a_K} (c_{a_1,a_2,\ldots,a_K} - c_{a'_1,a_2,\ldots,a_K})^2 > 0$.

Let $\mathcal{N} = [\![c^2_{a_1,\ldots,a_K}]\!] \in \mathbb{R}^{R_1\times\cdots\times R_K}$ be the quadratic loss evaluated at block, $W_{r_1,\ldots,r_K} = \prod_k \hat{p}^{(k)}_{r_k} > 0$ the size for the block indexed by $(r_1, \ldots, r_K)$. For ease of notation, we drop the subscript $(r_1, \ldots, r_K)$ and simply write W.

Based on the convexity of quadratic loss, there exists $c_* \in \mathbb{R}$ such that the weighted quadratic loss can be expressed as

$$\begin{aligned}
&[\mathcal{N} \times_1 \boldsymbol{D}^{(1)^T} \times_2 \cdots \times_K \boldsymbol{D}^{(K)^T}]_{r_1,\ldots,r_K} \\
&= D^{(1)}_{a_1 r_1} D^{(2)}_{a_2 r_2} \cdots D^{(K)}_{a_K r_K} c^2_{a_1,a_2,\ldots,a_K} + D^{(1)}_{a'_1 r_1} D^{(2)}_{a_2 r_2} \cdots D^{(K)}_{a_K r_K} c^2_{a'_1,a_2,\ldots,a_K} + \\
&(W - D^{(1)}_{a_1 r_1} D^{(1)}_{a_2 r_2} \cdots D^{(K)}_{a_K r_K} - D^{(1)}_{a'_1 r_1} D^{(1)}_{a_2 r_2} \cdots D^{(K)}_{a_K r_K}) c^2_*.
\end{aligned}$$

Recall that $\mu_{r_1,\ldots,r_K} = \frac{1}{W}[\mathcal{C} \times_1 \boldsymbol{D}^{(1)^T} \times_2 \cdots \times_K \boldsymbol{D}^{(K)^T}]_{r_1,\ldots,r_K}$ is the $(r_1,\ldots,r_k)$-th weighted entry of the block means. By the Taylor expansion of quadratic loss function at $\mu_{r_1,\ldots,r_K}$, we have

$$\begin{aligned}
&\frac{1}{W}[\mathcal{N} \times_1 \boldsymbol{D}^{(1)^T} \times_2 \cdots \times_K \boldsymbol{D}^{(K)^T}]_{r_1,\ldots,r_K} - \mu^2_{r_1,\ldots,r_K} \\
\geq &\frac{1}{2W} D^{(1)}_{a_1 r_1} D^{(2)}_{a_2 r_2} \cdots D^{(K)}_{a_K r_K} (c_{a_1,a_2,\ldots,a_K} - \mu_{r_1,\ldots,r_K})^2 + \\
&\frac{1}{2W} D^{(1)}_{a_1',r_1} D^{(2)}_{a_2 r_2} \cdots D^{(K)}_{a_K r_K} (c_{a_1',a_2,\ldots,a_K} - \mu_{r_1,\ldots,r_K})^2 + \\
&\frac{1}{2W} \left(W - D^{(1)}_{a_1 r_1} D^{(2)}_{a_2,r_2} \cdots D^{(K)}_{a_K r_K} - D^{(1)}_{a_1',r_1} D^{(2)}_{a_2,r_2} \cdots D^{(K)}_{a_K r_K} \right) (c_* - \mu_{r_1,\ldots,r_K})^2. \qquad (12)
\end{aligned}$$

Combining (12) and basic inequality $(a^2+b^2) \geq \frac{1}{2}(a+b)^2$ gives

$$\begin{aligned}
&\frac{1}{W}[\mathcal{N} \times_1 \boldsymbol{D}^{(1)^T} \times_2 \cdots \times_K \boldsymbol{D}^{(K)^T}]_{r_1,\ldots,r_K} - \mu^2_{r_1,\ldots,r_K} \\
\geq &\frac{1}{4W} \min \left\{ D^{(1)}_{a_1 r_1},\ D^{(1)}_{a_1' r_1} \right\} D^{(2)}_{a_2 r_2} \cdots D^{(K)}_{a_K r_K} (c_{a_1,\ldots,a_K} - c_{a_1',\ldots,a_K})^2 \\
\geq &\frac{\varepsilon D^{(2)}_{a_2 r_2} \cdots D^{(K)}_{a_K r_K}}{4W} (c_{a_1,a_2,\ldots,a_K} - c_{a_1',a_2,\ldots,a_K})^2. \qquad (13)
\end{aligned}$$

The inequality (13) only holds for a certain $r_1 \in [R_1]$. For any other $r_1' \in [R_1]/\{r_1\}$, by Jensen's inequality we have

$$\frac{1}{W}[\mathcal{N} \times_1 \boldsymbol{D}^{(1)^T} \times_2 \cdots \times_K \boldsymbol{D}^{(K)^T}]_{r_1',\ldots,r_K} - \mu^2_{r_1',\ldots,r_K} \geq 0. \qquad (14)$$

Combining the sum of (13) and (14) over $(r_2,\ldots,r_K)$ gives

$$\begin{aligned}
&G(\boldsymbol{D}^{(1)},\ldots,\boldsymbol{D}^{(K)}) - \sum_{r_1,\ldots,r_K} p^{(1)}_{r_1} \cdots p^{(K)}_{r_K} c^2_{r_1,\ldots,r_K} \\
\leq\ & -\varepsilon \sum_{r_2,\ldots,r_K} \frac{D^{(2)}_{a_2 r_2} \cdots D^{(K)}_{a_K r_K}}{4} (c_{a_1,a_2,\ldots,a_K} - c_{a_1',a_2,\ldots,a_K})^2 \\
\leq\ & -\frac{1}{4}\varepsilon \tau^{K-1} \delta_{\min},
\end{aligned}$$

where the last line uses the fact that $\sum_{r_k} D^{(k)}_{a_k r_k} = p^{(k)}_{a_k} \geq \tau$. □

A.4.3 Proof

Proof of Theorem 2. The notations we use here are inherited from Lemma 2. By lemma 2, we obtain that

$$\begin{aligned}
&\mathbb{P}\left(\mathrm{MCR}(\widehat{\boldsymbol{M}}_k, \boldsymbol{M}_{k,\text{true}}) \geq \varepsilon \right) \\
\leq &\mathbb{P}\left(G(\boldsymbol{D}^{(1)},\ldots,\boldsymbol{D}^{(K)}) - G(\boldsymbol{M}_{1,\text{true}},\ldots,\boldsymbol{M}_{K,\text{true}}) \leq -\frac{1}{2}\varepsilon\tau^{K-1}\delta_{\min} \right). \qquad (15)
\end{aligned}$$

Define $r = \sup_{\mathcal{J}_\tau} |F(\boldsymbol{M}_1,\ldots,\boldsymbol{M}_K) - G(\boldsymbol{D}^{(1)},\ldots,\boldsymbol{D}^{(K)})|$ as the stochastic deviation caused by the label assignment. When the event $G(\boldsymbol{D}^{(1)},\ldots,\boldsymbol{D}^{(K)}) - G(\boldsymbol{M}_{1,\text{true}},\ldots,\boldsymbol{M}_{K,\text{true}}) \leq -\frac{1}{2}\varepsilon\tau^{K-1}\delta_{\min}$ holds, by triangle inequality, we have

$$F(\widehat{\boldsymbol{M}}_1,\ldots,\widehat{\boldsymbol{M}}_K) - F(\boldsymbol{M}_{1,\text{true}},\ldots,\boldsymbol{M}_{K,\text{true}}) \leq 2r - \frac{1}{2}\varepsilon\tau^{K-1}\delta_{\min}. \qquad (16)$$

Plugging the event (16) back into inequality (15), we obtain

$$\begin{aligned}
&\mathbb{P}\left(\mathrm{MCR}(\widehat{\boldsymbol{M}}_k, \boldsymbol{M}_{k,\text{true}}) \geq \varepsilon\right) \\
\leq &\mathbb{P}\left(F(\widehat{\boldsymbol{M}}_1, \ldots, \widehat{\boldsymbol{M}}_K) - F(\boldsymbol{M}_{1,\text{true}}, \ldots, \boldsymbol{M}_{K,\text{true}}) \leq 2r - \frac{1}{2}\varepsilon\tau^{K-1}\delta_{\min}\right) \\
= &\mathbb{P}\left(r \geq \frac{\varepsilon\tau^{K-1}\delta_{\min}}{4}\right).
\end{aligned} \tag{17}$$

Now we aim to bound $r = \sup_{\mathcal{J}_\tau} |F(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K) - G(\boldsymbol{D}^{(1)}, \ldots, \boldsymbol{D}^{(K)})|$. Note that r involves the quadratic objective $f(x) = x^2$. The quadratic function $f(x)$ is locally lipschitz continuous with lipschitz constant $b = \sup_x |f'(x)|$, where x is in the closure of the convex hull of the entries of $\mathcal{C}$. Therefore, for any partitions $\{\boldsymbol{M}_k\}$ (which are necessarily equal to $\{\hat{\boldsymbol{M}}_k\}$ or $\{\boldsymbol{M}_{k,\text{true}}\}$):

$$\begin{aligned}
&\left|F(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K) - G(\boldsymbol{D}^{(1)}, \ldots, \boldsymbol{D}^{(K)})\right| \\
\leq &\sum_{r_1,\ldots,r_K} p^{(1)}_{r_1} p^{(2)}_{r_2} \cdots p^{(K)}_{r_K} \left|f(\hat{c}_{r_1,\ldots,r_K}) - f(\mu_{r_1,\ldots,r_K})\right| \\
\leq &\, b\|\mathcal{R}(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K)\|_\infty,
\end{aligned} \tag{18}$$

where

$$\hat{c}_{r_1,\ldots,r_K} = \frac{1}{\prod_k p^{(k)}_{r_k}} (\mathcal{Y} \times_1 \boldsymbol{M}_1^T \times_2 \cdots \times_K \boldsymbol{M}_K^T)_{r_1,\ldots,r_K},$$

and

$$\mu_{r_1,\ldots,r_K} = \frac{1}{\prod_k p^{(k)}_{r_k}} \left[\mathcal{C} \times_1 \boldsymbol{D}^{(1)^T} \times_2 \cdots \times_K \boldsymbol{D}^{(K)^T}\right]_{r_1,\ldots,r_K}.$$

are, respectively, sample average and expected sample average, conditional on the partitions $\boldsymbol{M}_k$, and $\mathcal{R}(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K)$ is the residual tensor defined in (11).

Combining (17), (18) and Hoeffding's inequality, we have

$$\begin{aligned}
\mathbb{P}\left(\mathrm{MCR}(\widehat{\boldsymbol{M}}_k, \boldsymbol{M}_{k,\text{true}}) \geq \varepsilon\right) &\leq \mathbb{P}\left(\sup_{\mathcal{J}_\tau} \|\mathcal{R}(\boldsymbol{M}_1, \ldots, \boldsymbol{M}_K)\|_\infty \geq \frac{\varepsilon\tau^{K-1}\delta_{\min}}{4b}\right) \\
&\leq \mathbb{P}\left(\sup_{I \in \mathcal{I}} \frac{\sum_{(i_1,\ldots,i_K)\in I} (Y_{i_1,\ldots,i_K} - \mathbb{E}(Y_{i_1,\ldots,i_K}))}{|I|} \geq \frac{\varepsilon\tau^{K-1}\delta_{\min}}{4b}\right) \\
&\leq 2^{1+\sum_k d_k} \exp\left(-\frac{\varepsilon^2\tau^{2(K-1)}\delta^2_{\min} L}{64\sigma^2 b^2}\right),
\end{aligned} \tag{19}$$

where the last line comes from the fact that the residuals are i.i.d. sub-Gaussian (conditional on partition $\{\boldsymbol{M}_k\}$), and $L = \inf\{|I| : I \subset \mathcal{I}\} \geq \tau^K \prod_{k=1}^K d_k$ is introduced in Section A.4.1. Defining $C = \frac{1}{64b^2}$ in (19) yields the desired conclusion.

□

A.5 Sparse estimator

Lemma 3. *Consider the regularized least-square estimation,*

$$\hat{\Theta}^{sparse} = \arg\min_{\Theta \in \mathcal{P}} \left\{\|\mathcal{Y} - \Theta\|_F^2 + \lambda\|\mathcal{C}\|_\rho\right\}, \tag{20}$$

where $\mathcal{C} = [\![c_{r_1,\ldots,r_K}]\!] \in \mathbb{R}^{R_1 \times \cdots \times R_K}$ 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. We have

$$\hat{c}^{sparse}_{r_1,\ldots,r_K} = \begin{cases} \hat{c}^{ols}_{r_1,\ldots,r_K} \mathbb{1}\left\{|\hat{c}^{ols}_{r_1,\ldots,r_K}| \geq \sqrt{\frac{\lambda}{n_{r_1,\ldots,r_K}}}\right\} & \text{if } \rho = 0, \\ sign(\hat{c}^{ols}_{r_1,\ldots,r_K}) \left(|\hat{c}^{ols}_{r_1,\ldots,r_K}| - \frac{\lambda}{2n_{r_1,\ldots,r_K}}\right)_+ & \text{if } \rho = 1, \end{cases} \tag{21}$$

where $a_+ = \max(a, 0)$ and $\hat{c}^{ols}_{r_1,\ldots,r_K}$ denotes the ordinary least-square estimate as in Algorithm 1.

Proof. We formulate the estimation of $\mathcal{C}$ as a regularized least-square regression. Note that $\Theta \in \mathcal{P}$ implies that

$$\Theta = \mathcal{C} \times_1 \boldsymbol{M}_1 \times \cdots \times_K \boldsymbol{M}_K.$$

Define $\boldsymbol{X} = \boldsymbol{M}_1 \otimes \ldots \otimes \boldsymbol{M}_K \in \mathbb{R}^{d \times R}$, where $d = \prod_k d_k$ and $R = \prod_k R_k$, and $\boldsymbol{\beta} = \text{vec}(\mathcal{C}) \in \mathbb{R}^R$. Here $\boldsymbol{X}$ is a membership matrix that indicates the block allocation among tensor entries. Specifically, $\boldsymbol{X}$ consists of orthogonal columns with $\boldsymbol{X}^T\boldsymbol{X} = \text{diag}(n_1, \ldots, n_R)$, where n_r is the number of entries in the tensor block that corresponds to the r-th column of $\boldsymbol{X}$.

For a given set of $\boldsymbol{M}_k's$, the optimization (22) with respect to $\mathcal{C}$ is equivalent to a regularized linear regression with $\boldsymbol{Y} = \text{vec}(\mathcal{Y})$ as the response and $\boldsymbol{X}$ as the design matrix:

$$L(\boldsymbol{\beta}) = \|\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{\beta}\|_2^2 + \lambda\|\boldsymbol{\beta}\|_\rho. \tag{22}$$

When $\lambda = 0$ (no penalty), the minimizer is $\hat{\boldsymbol{\beta}}^{\text{ols}} = (\hat{\beta}_1^{\text{ols}}, \ldots, \hat{\beta}_R^{\text{ols}}) = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{Y}$, where $\hat{\beta}_r^{\text{ols}} = \frac{1}{n_r}\boldsymbol{y_r}\mathbf{1}_{n_r}^T$ for all $r \in [R]$.

Case 1: $\rho = 0$.

Note that $\boldsymbol{X}$ induces a partition of indices $[d]$ into R blocks. With a little abuse of notation, we use $\mathcal{R} = \{i \in [d] : \boldsymbol{X}(i) = r\}$ to denote the collection of tensor indices that belong to the rth block, and use $\boldsymbol{Y}_\mathcal{R} \in \mathbb{R}^{n_r}$ to denote the corresponding tensor entries. By the orthogonality of $\boldsymbol{X}$, we have

$$\begin{aligned} L(\boldsymbol{\beta}) &= \sum_{r=1}^R \|\boldsymbol{Y}_\mathcal{R} - \beta_r \mathbf{1}_{n_r}\|_2^2 + \lambda \sum_{r=1}^R \mathbb{1}\{\beta_r \neq 0\} \\ &= \sum_{r=1}^R \underbrace{\left(\|\boldsymbol{Y}_\mathcal{R} - \beta_r \mathbf{1}_{n_r}\|_2^2 + \lambda \mathbb{1}\{\beta_r \neq 0\}\right)}_{:= L_r(\beta_r)} \end{aligned}$$

The optimization can be separated into each of β_r's. For any $r \in [R]$, the sub-optimization $\min_{\beta_r} L_r(\beta_r)$ has a closed-form solution

$$\min_{\beta_r} L_r(\beta_r) = \begin{cases} \boldsymbol{Y}_\mathcal{R}^T\boldsymbol{Y}_\mathcal{R} - n_r\left(\hat{\beta}_r^{\text{ols}}\right)^2 + \lambda & \text{if } \hat{\beta}_r^{\text{ols}} \neq 0, \\ \boldsymbol{Y}_\mathcal{R}^T\boldsymbol{Y}_\mathcal{R} & \text{if } \hat{\beta}_r^{\text{ols}} = 0, \end{cases}$$

with

$$\arg\min_{\beta_r} L_r(\beta_r) = \begin{cases} 0 & \text{if } n_r\left(\hat{\beta}_r^{\text{ols}}\right)^2 \leq \lambda, \\ \hat{\beta}_r^{\text{ols}} & \text{otherwise.} \end{cases} \tag{23}$$

Solution (23) can be simplified as $\hat{\beta}_r^{\text{sparse}} = \hat{\beta}_r^{\text{ols}}\mathbb{1}\{|\hat{\beta}_r^{\text{ols}}| \leq \sqrt{\frac{\lambda}{n_r}}\}$. The proof is complete by noting that $\hat{c}_{r_1,\ldots,r_R}^{\text{sparse}} = \hat{\beta}_r^{\text{sparse}}$ and $n_{r_1,\ldots,r_K} = n_r$ for all $(r_1, \ldots, r_K) \in [R_1] \times \cdots \times [R_K]$.

Case 2: $\rho = 1$.

Similar as in Case 1, we write the optimization (22) as

$$L(\boldsymbol{\beta}) = \sum_{r=1}^R \underbrace{\left(\|\boldsymbol{Y}_\mathcal{R} - \beta_r \mathbf{1}_{n_r}\|_2^2 + \lambda|\beta_r|\right)}_{:= L_r(\beta_r)},$$

where, with a little abuse of notation, we still use $L_r(\beta_r)$ to denote the sub-optimization. To solve $\arg\min_{\beta_r} L_r(\beta_r)$, we use the properties of subderivative. Taking the subderivative with respect to β_r, we obtain

$$\frac{\partial L_r(\beta_r)}{\partial \beta_r} = \begin{cases} 2n_r\beta_r - 2n_r\hat{\beta}_r^{\text{ols}} + \lambda & \text{if } \beta_r > 0, \\ [2n_r\beta_r - 2\hat{\beta}_r^{\text{ols}} - \lambda,\ 2n_r\beta_r - \hat{\beta}^{\text{ols}} + \lambda] & \text{if } \beta_r = 0, \\ 2n_r\beta_r - 2n_r\hat{\beta}_r^{\text{ols}} + \lambda & \text{if } \beta_r < 0. \end{cases}$$

Because $\hat{\beta}_r^{\text{sparse}}$ minimizes $L_r(\beta_r)$ if and only if $0 \in \frac{\partial L_r(\beta_r)}{\partial \beta_j}$, we have:

$$\hat{\beta}_r^{\text{sparse}} = \begin{cases} \hat{\beta}_r^{\text{ols}} + \frac{\lambda}{2n_r} & \text{if } \hat{\beta}_r^{\text{ols}} < -\frac{\lambda}{2n_r}, \\ 0 & \text{if } \hat{\beta}_r^{\text{ols}} \in [-\frac{\lambda}{2n_r}, \frac{\lambda}{2n_r}], \\ \hat{\beta}_r^{\text{ols}} - \frac{\lambda}{2n_r} & \text{if } \hat{\beta}_r^{\text{ols}} > \frac{\lambda}{2n_r}. \end{cases} \tag{24}$$

The solution (24) can be simplified as

$$\hat{\beta}_r^{\text{sparse}} = \text{sign}(\hat{\beta}_r^{\text{ols}}) \left(|\hat{\beta}_r^{\text{ols}}| - \frac{\lambda}{2n_r} \right)_+, \quad \text{for all } r \in [R].$$

□

B Supplementary Figures and Tables

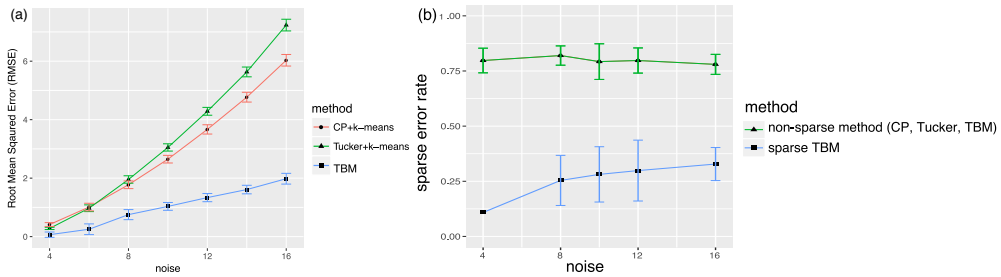

Figure S1: (a) estimation error and (b) sparse error rate against noise for sparse tensors of dimension $(40, 40, 40)$ when $p = 0.8$.

Dimensions (d_1, d_2, d_3)	True clustering sizes (R_1, R_2, R_3)	Noise (σ)	Estimated clustering sizes $(\hat{R}_1, \hat{R}_2, \hat{R}_3)$
$(40, 40, 40)$	$(4, 4, 4)$	4	$(\mathbf{4},\ \mathbf{4},\ \mathbf{4}) \pm (0,\ 0,\ 0)$
$(40, 40, 40)$	$(4, 4, 4)$	8	$(\mathbf{3.94},\ \mathbf{3.96},\ \mathbf{3.96}) \pm (0.03,\ 0.03,\ 0.03)$
$(40, 40, 40)$	$(4, 4, 4)$	12	$(3.08,\ 3.12,\ 3.12) \pm (0.10, 0.10, 0.10)$
$(40, 40, 80)$	$(4, 4, 4)$	4	$(\mathbf{4},\ \mathbf{4},\ \mathbf{4}) \pm (0,\ 0,\ 0)$
$(40, 40, 80)$	$(4, 4, 4)$	8	$(\mathbf{4},\ \mathbf{4},\ \mathbf{4}) \pm (0,\ 0,\ 0)$
$(40, 40, 80)$	$(4, 4, 4)$	12	$(\mathbf{3.96},\ \mathbf{3.96},\ 3.92) \pm (0.04, 0.04, 0.04)$
$(40, 40, 40)$	$(2, 3, 4)$	4	$(\mathbf{2},\ \mathbf{3},\ \mathbf{4}) \pm (0,\ 0,\ 0)$
$(40, 40, 40)$	$(2, 3, 4)$	8	$(\mathbf{2},\ \mathbf{3},\ \mathbf{3.96}) \pm (0,\ 0,\ 0.03)$
$(40, 40, 40)$	$(2, 3, 4)$	12	$(\mathbf{2},\ \mathbf{2.96},\ 3.60) \pm (0,\ 0.05,\ 0.09)$

Table S1: The simulation results for estimating $\boldsymbol{R} = (R_1, R_2, R_3)$. Bold number indicates no significant difference between the estimate and the ground truth, based on a z-test with a level 0.05.

Tissues	Over-expressed genes	Block-means	Under-expressed genes	Block-means
Cluster 1	GFAP, MBP	10.88	GPR6 , DLX5 , DLX6 , NKX2-1	-8.40
Cluster 2	GFAP, MBP	5.98	CDH9, RXFP1, CRH, ARX, CARTPT, DLX1,FEZF2	-9.49
Cluster 3	GFAP, MBP	8.34	AVPR1A, CCKAR, CHRNB4, CYP19A1, HOXA4 , LBX1, SLC6A3 TBR1, SLC17A6, SLC30A3	-8.45 -8.17
Cluster 4	GFAP, MBP	8.83	AVPR1A, CCKAR, CHRNB4, CYP19A1, HOXA4 , LBX1, SLC6A3 DAO EN2 EOMES	-8.40 -6.57

Table S2: Top expression blocks from the multi-tissue gene expression analysis. The tissue clusters are described in Supplementary Section D.

Countries	Countries	Relation types
Cluster 1	Clusters 4 and 5	reltreaties, booktranslations, relbooktranslations, relexports, exports3
Clusters 1 and 4	Cluster 5	relintergovorgs, relngo, intergovorgs3, ngoorgs3
Cluster 3	Clusters 1, 4, and 5	commonbloc0, blockpositionindex
Clusters 1 and 3 Cluster 1 Cluster 4	Clusters 4 and 5 Cluster 3 Cluster 5	timesinceally, independence
Cluster 4	Cluster 5	treaties, conferences, weightedunvote, unweightedunvote, intergovorgs, ngo, officialvisits, exportbooks, relexportbooks, tourism, reltourism, tourism3, exports, militaryalliance, commonbloc2

Table S3: Top blocks from the *Nations* data analysis. The countries clusters are described in Supplementary Section D.

C Time complexity

The total cost of our Algorithm 1 is $\mathcal{O}(d)$ per iteration, where $d = \prod_k d_k$ denotes the total number of tensor entries. The per-iteration computational cost scales linearly with the sample size, and this complexity is comparable to the classical tensor methods such as CP and Tucker decomposition. More specifically, each iteration of Algorithm 1 consists of updating the core tensor $\mathcal{C}$ and K membership matrices $\boldsymbol{M}_k$'s. The update of $\mathcal{C}$ requires $\mathcal{O}(d)$ operations and the update of $\boldsymbol{M}_k$ requires $\mathcal{O}(R_k \frac{d}{d_k})$ operations. Therefore the total cost is $\mathcal{O}(d + d\sum_k \frac{R_k}{d_k})$.

D Additional information for real data analysis

Multi-tissue gene expression. The gene expression data we analyzed is part of the GTEx v6 datasets (`https://www.gtexportal.org/home/datasets`). We cleaned and preprocessed the data following the steps in [2]. We focused on the 13 brain tissues, 193 individuals, and 362 annotated genes provided by Atlax of the Developing Human Brain (`http://www.brainspan.org/ish`). After applying the ℓ-0 penalized TBM to the mean-centered data tensor, we identified the following four clusters of tissues:

- Cluster 1: Substantia nigra, Spinal cord (cervical c-1)
- Cluster 2: Cerebellum, Cerebellar Hemisphere
- Cluster 3: Caudate (basal ganglia), Nucleus accumbens (basal ganglia), Putamen (basal ganglia)
- Cluster 4: Cortex, Hippocampus, Anterior cingulate cortex (BA24), Frontal Cortex (BA9), Hypothalamus, Amygdala

We found that most tissue clusters are spatially restricted to specific brain regions, such as the two cerebellum tissues (cluster 2), three basal ganglia tissues (cluster 3), and the cortex tissues (cluster 4). Supplementary Table S2 reports the associated gene cluster for each tissue cluster. Because our method attaches importance to blocks by the absolute mean estimates, our method is able to detect both over- and under-expression patterns. Blocks with highly positive means correspond to over-expressed genes, whereas blocks with highly negative means correspond to under-expressed genes.

Nations dataset. This is a $14 \times 14 \times 56$ binary tensor consisting of 56 political relations of 14 countries between 1950 and 1965 [3]. The tensor entry indicates the presence or absence of a political action, such as "treaties", "sends tourists to", between the nations. We applied the ℓ-0 penalized TBM to the binary-valued data tensor, and we identified the following five clusters of countries:

- Cluster 1: Brazil, Egypt, India, Israel, Netherlands
- Cluster 2: Burma, Indonesia, Jordan
- Cluster 3: China, Cuba, Poland, USSA
- Cluster 4: USA
- Cluster 5: UK

Supplementary Table S3 reports the cluster constitutions for top blocks. Because the tensor entries take value on either 0 or 1, the top blocks mostly have mean one.

References

[1] Phillippe Rigollet and Jan-Christian Hütter. High dimensional statistics. *Lecture notes for course 18S997*, 2015.

[2] 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.

[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.