Smooth tensor estimation with unknown permutations

Abstract

We consider the problem of structured tensor denoising in the presence of unknown permutations. Such data problems arise commonly in recommendation system, neuroimaging, community detection, and multiway comparison applications. Here, we develop a general family of smooth tensor models up to arbitrary index permutations; the model incorporates the popular tensor block models and Lipschitz hypergraphon models as special cases. We show that a constrained least-squares estimator in the block-wise polynomial family achieves the minimax error bound. A phase transition phenomenon is revealed with respect to the smoothness threshold needed for optimal recovery. In particular, we find that a polynomial of degree up to $(m-2)(m+1)/2$ is sufficient for accurate recovery of order-m tensors, whereas higher degree exhibits no further benefits. This phenomenon reveals the intrinsic distinction for smooth tensor estimation problems with and without unknown permutations. Furthermore, we provide an efficient polynomial-time Borda count algorithm that provably achieves optimal rate under monotonicity assumptions. The efficacy of our procedure is demonstrated through both simulations and Chicago crime data analysis.

Keywords: Tensor estimation, latent permutation, diverging dimensionality, phase transition, statistical-computational efficiency.

1 Introduction

Higher-order tensor datasets are rising ubiquitously in modern data science applications, for instance, recommendation systems [2, 3], social networks [4], genomics [13], and neuroimaging [36]. Tensor provides effective representation of data structure that classical vector- and matrix-based methods fail to capture. One example is music recommendation system [2] that records ratings of songs from users on various contexts. This three-way tensor of user $\times$ song $\times$ context allows us to investigate interactions of users and songs in a context-specific manner. Another example is network dataset that records the connections among a set of nodes. Pairwise interactions are often insufficient to capture the complex relationships, whereas multi-way interactions improve the understanding of networks in molecular system [33] and social networks [12]. In both examples, higher-order tensors represent multi-way interactions in an efficient way.

Tensor estimation problem cannot be solved without imposing structures. An appropriate reordering of tensor entries often provides effective representation of the hidden salient structure. In the music recommendation example, suppose that we have certain criteria available (such as, similarities of music genres, ages of users, and importance of contexts) to reorder the songs, users, and contexts. Then, the sorted tensor will exhibit smooth structure, because entries from similar groups tend to have similar values. Similar observation applies to network examples. An m-uniform hypergraph network can be represented by an order-m adjacency tensor, with entries indicating the presence and absence of m-way interactions among a set of nodes. Suppose the characteristics of individual nodes are available so that one can rearrange nodes based on their similarities. Then, the sorted adjacency tensor will exhibit smooth structure by the same reason.

In this article, we develop a *permuted* smooth tensor model based on the aforementioned

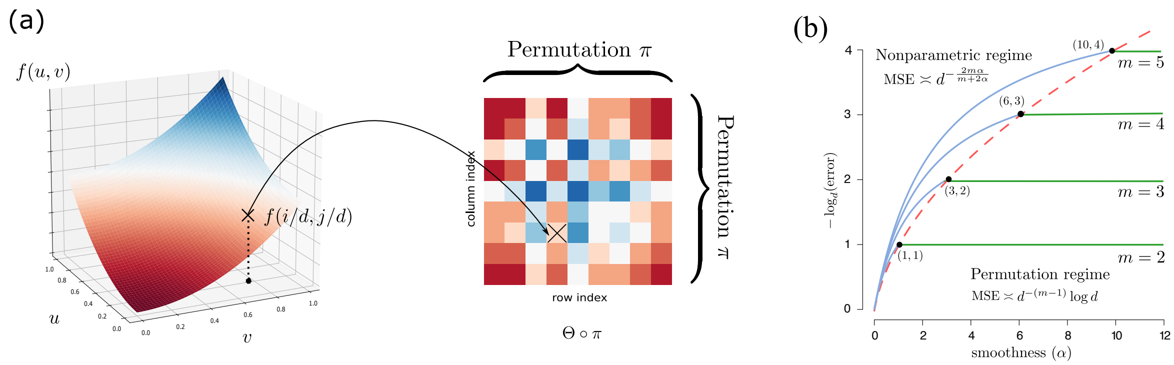


Figure 1: (a): Illustration of order-m d-dimensional permuted smooth tensor models with $m = 2$. (b): Phase transition of mean squared error (MSE) (on $-\log_d$ scale) as a function of smoothness α and tensor order m. Bold dots correspond to the critical smoothness level above which higher smoothness exhibits no further benefits to tensor estimation.

motivation. We study a class of structured tensors, called *permuted smooth tensor model*, of the following form:

$$\mathcal{Y} = \Theta \circ \pi + \text{noise}, \quad \text{where} \quad \Theta(i_1, \ldots, i_m) = f\left(\frac{i_1}{d}, \ldots, \frac{i_m}{d}\right), \tag{1}$$

where $\pi \colon [d] \to [d]$ is an *unknown* latent permutation, Θ is an *unknown* order-m d-dimensional signal tensor, and f is an *unknown* multivariate function with certain notion of smoothness, and $\Theta \circ \pi$ denotes the permuted tensor after reordering the indices along each of the m modes. Figure 1(a) shows an example of this generative model for the matrix case $m = 2$. Our primary goal is to estimate the permuted smooth signal tensor $\Theta \circ \pi$ from the noisy tensor observation $\mathcal{Y}$ of arbitrary order m.

1.1 Our contributions

We develops a suite of statistical theory, efficient algorithms, and related applications for permuted smooth tensor models (1). Our contributions are summarized below.

First, we develop a general permuted α-smooth tensor model of arbitrary smoothness

	Pananjady et al [22]	Balasubramanian [1]	Li et al. [18]	**Ours**
Model structure	monotonic	Lipschitz	Lipschitz	α-smoothness
Error rate for order-m tensor*	d^{-1}	$d^{-2m/(m+2)}$	$d^{-\lfloor m/3 \rfloor}$	$d^{-(m-1)}$
(e.g., when $m = 3$)		$(d^{-6/5})$	(d^{-1})	(d^{-2})
Minimax optimality	$\surd$	$\times$	$\times$	$\surd$
Polynomial algorithm	$\surd$	$\times$	$\surd$	$\surd$

Table 1: Comparison of our results with previous work. *For simplicity, we list here the error rate (omitting the log term) for ∞-smooth tensors. Our results allow general tensors of arbitrary smoothness level $\alpha \geq 0$; See Theorems 1-3 in Sections 4-5.

level $\alpha \geq 0$. We establish the statistically optimal error rate and its dependence on model complexity, including tensor order, tensor dimension, smoothness level, signal-to-noise ratio, and unknown permutations. Table 1 summarizes the comparison of our work with previous results. Our framework substantially generalizes earlier works which focus on only matrices with $m = 2$ [10, 16] or Lipschitz function with $\alpha = 1$ [1, 18]. The generalization enables us to obtain results previously impossible: i) As tensor order m increases, we demonstrate the failure of pervious clustering-based algorithms [1, 10], and we develop a new block-wise polynomial algorithm for tensors of order $m \geq 3$; ii) As smoothness α increases, we demonstrate that the error rate converges to a fast rate $\mathcal{O}(d^{-(m-1)})$, thereby disproving the conjectured lower bound $\mathcal{O}(d^{-2m/(m+2)})$ posed by earlier work [1]. The results showcase the accuracy gain of our new approach, as well as the intrinsic distinction between matrices and higher-order tensors.

Second, we discover a phase transition phenomenon with respect to the smoothness needed for optimal recovery in model (1). Figure 1(b) plots the dependence of estimation error in terms of smoothness level α for tensors of order m. We characterize two distinct error behaviors determined by a critical smoothness threshold; see Theorems 1-2 in Section 4. Specifically, the accuracy improves with α in the regime $\alpha \leq m(m-1)/2$, but then it becomes a constant of α in the regime $\alpha > m(m-1)/2$. The results imply a polynomial of degree $(m-2)(m+1)/2 = [m(m-1)/2 - 1]$ is sufficient for accurate recovery of order-m tensors of arbitrary smoothness in model (1), whereas higher degree brings no further benefits. The phenomenon is distinctive from matrix problems [16, 10] and classical *non-permuted* smooth function estimation [28], thereby highlighting the fundamental challenges in our new setting. These statistical contributions, to our best knowledge, are new to the literature of general permuted smooth tensor problems.

Third, we propose two estimation algorithms with accuracy guarantees: the least-squares estimation and Borda count estimation. The least-squares estimation, although being computationally hard, reveals the fundamental model complexity in the problem. The result serves as the benchmark and a useful guide to the algorithm design. Furthermore, we develop an efficient polynomial-time Borda count algorithm that provably achieves optimal rate under monotonicity assumptions. The algorithm handles a broad range of data types, including continuous and binary observations.

Lastly, we illustrate the efficacy of our method through both simulations and data applications. A range of practical settings are investigated in simulations, and we show the outperformance of our method compared to alternative approaches. Application to Chicago crime data is presented to showcase the usefulness of our method. We identify the key global pattern and pinpoint local smooth structure in the denoised tensor. Our method will help practitioners efficiently analyze tensor datasets in various areas. Toward this end, the package and all data used are released at CRAN.

1.2 Related work

Our work is closely related to but also clearly distinctive from several lines of existing research. We review related literature for comparison.

Structure learning with latent permutations. The estimation problem of (1) falls into the general category of structured learning with *latent permutation*. Models involving latent permutations have recently received a surge of interest, include graphons [5, 16], stochastic transitivity models [6, 24], statistical seriation [9, 14], and graph matching [8, 19]. These methods, however, are developed for matrices; the tensor counterparts are far less well understood. Table 1 summarizes the most related works to ours. Pananjady and Samworth [22] studied the permuted tensor estimation under isotonic constraints. We find that our smooth model results in a much faster rate $\mathcal{O}(d^{-(m-1)})$ than the rate $\mathcal{O}(d^{-1})$ for isotonic models. The works [1, 18] studied similar smooth models as ours, but we gain substantial improvement in both statistics and computations. Balasubramanian [1] developed a (non-polynomial-time) clustering-based algorithm with a rate $\mathcal{O}(d^{-2m/(m+2)})$. Li et al. [18] developed a (polynomial-time) nearest neighbor estimation with a rate $\mathcal{O}(d^{-\lfloor m/3 \rfloor})$. Neither approach investigates the minimax optimality. By contrast, we develop a polynomial-time algorithm with a fast rate $\mathcal{O}(d^{-(m-1)})$ under mild conditions. The optimality of our estimator is safeguarded by matching a minimax lower bound.

Low-rank tensor models. There is a huge literature on structured tensor estimation under low-rank models, including CP models [17, 26], Tucker models [34], and block models [30]. These models belong to parametric approaches, because they aim to explain the data with a finite number of parameters (i.e., decomposed factors). Our permuted smooth tensor model utilizes a different measure of model complexity than the usual low-rankness. We use *infinite* number of parameters (i.e., smooth functions) to allow growing model complexity. In this sense, our method belongs to nonparametric approaches. The comparison and benefits of nonparametric methods over parametric ones were discussed previously [22, 18, 10, 4, 24].

Nonparametric regression. Our model is also related to nonparametric regression [28]. One may view the problem (1) as a nonparametric regression, where the goal is to learn the function f based on scalar response $\mathcal{Y}(i_1,\ldots,i_m)$ and design points $(\pi(i_1),\ldots,\pi(i_m))$ in $\mathbb{R}^m$; see Figure 1(a). However, the *unknown* permutation π significantly influences the statistical and computational hardness of the problem. This latent π leads to a phase transition behavior in the estimation error; see Figure 1(b) and Sections 4. We reveal two components of error for the problem, one for nonparametric error and the other for permutation error. The impact of unknown permutation hinges on tensor order and smoothness in an intriguing way (see Theorems 1-3). This is clearly contrary to classical nonparametric regression.

Graphon and hypergraphon. Our work is also connected to graphons and hypergraphons. Graphon is a measurable function representing the limit of a sequence of exchangeable random graphs (matrices) [16, 10, 5]. Similarly, hypergraphon [35, 20] is introduced as a limiting function of m-uniform hypergraphs, i.e., a generalization of graphs in which edges can join m vertices with $m \geq 3$. While both our model (1) and hypergraphon focus on function representations, there are two remarkable differences. First, unlike the matrix case where graphon is represented as bivariate functions [20], hypergraphons for m-uniform hypergraphs should be represented as (2^m-2)-multivariate functions; see Zhao [35, Section 1.2]. Our framework (1) represents the function using m coordinates only, and in this sense, the model shares the common ground as *simple hypergraphons* [1]. We compare our method to earlier work in theory (Table 1 and Sections 4-5) and in numerical studies (Section 6). Second, unlike typical simple hypergraphons where the design points are random, our generative model uses deterministic design points. These two choices lead to different analysis in the same spirit as random- vs. fixed-designs in nonparametric regression [31, 28].

1.3 Notation and organization

We use $\mathbb{N}_+$ to denote the set of positive integers, and $\mathbb{N}_{\geq 0} = \mathbb{N}_+ \cup \{0\}$. We use $[d] = \{1, \ldots, d\}$ for d-set with $d \in \mathbb{N}_+$. For a set S, $|S|$ denotes its cardinality and $\mathbb{1}_S$ denotes the indicator function. For two positive sequences $\{a_d\}, \{b_d\}$, we denote $a_d \lesssim b_d$ if $\lim_{d\to\infty} a_d/b_d \leq c$ for some constant $c > 0$, and $a_d \asymp b_d$ if $c_1 \leq \lim_{d\to\infty} a_d/b_d \leq c_2$ for some constants $c_1, c_2 > 0$. Given a number $a \in \mathbb{R}$, the floor function $\lfloor a \rfloor$ is the largest integer no greater than a, and the ceiling function $\lceil a \rceil$ is the smallest integer no less than a. We use $\mathcal{O}(\cdot)$ to denote big-O notation hiding logarithmic factors, and $\circ$ the function composition. Let $\Theta \in \mathbb{R}^{d\times\cdots\times d}$ be an order-m d-dimensional tensor, $\pi\colon [d] \to [d]$ be an index permutation, and $\Theta(i_1, \ldots, i_m)$ the tensor entry indexed by $(i_1, \ldots, i_m)$. We use $\Theta \circ \pi$ to denote the permuted tensor such that $(\Theta \circ \pi)(i_1, \ldots, i_m) = \Theta(\pi(i_1), \ldots, \pi(i_m))$ for all $(i_1, \ldots, i_m) \in [d]^m$. We sometimes also use shorthand notation $\Theta(\omega)$ for tensor entries with indices $\omega = (i_1, \ldots, i_w)$. We call a tensor *a binary-valued tensor* if its entries take value on $\{0, 1\}$-labels, and *a continues-valued tensor* if its entries take values on a continuous scale. We define the Frobounis norm $\|\Theta\|_F^2 = \sum_{\omega\in[d]^m} |\Theta(\omega)|^2$ for a tensor Θ, and the ∞-norm $\|\boldsymbol{x}\|_\infty = \max_i |x_i|$ for a vector $\boldsymbol{x} = (x_1, \ldots, x_d)^T$. We use $\Pi(d, d) = \{\pi\colon [d] \to [d]\}$ to denote all permutations on $[d]$, while $\Pi(d, k) = \{\pi\colon [d] \to [k]\}$ the collection of all onto mappings from $[d]$ to $[k]$. An event A is said to occur *with high probability* if $\mathbb{P}(A)$ tends to 1 as the tensor dimension $d \to \infty$.

The rest of the paper is organized as follows. Section 2 presents the permuted smooth tensor model and its connection to smooth function representation. In Section 3, we establish the approximation error based on block-wise polynomial approximation. Then, we develop two estimation algorithms with accuracy guarantees: the least-squares estimation and Borda count estimation. Section 4 presents a statistically optimal but computationally challenging least-squares estimator. Section 5 presents a polynomial-time Borda count algorithm with a provably same optimal rate under monotonicity assumptions. Simulations and data analyses are presented in Section 6. We conclude the paper with a discussion in Section 7. All proofs and extensions are deferred to Supplementary Materials.

2 Smooth tensor model with unknown permutation

Suppose we observe an order-m d-dimensional data tensor from the following model,

$$\mathcal{Y} = \Theta \circ \pi + \mathcal{E}, \tag{2}$$

where $\pi\colon [d] \to [d]$ is an unknown latent permutation, $\Theta \in \mathbb{R}^{d\times\cdots\times d}$ is an unknown signal tensor under certain smoothness (to be specified in next paragraph), and $\mathcal{E}$ is a noise tensor consisting of zero-mean, independent sub-Gaussian entries with variance bounded by σ^2. We allow heterogeneous and non-identically distributed entries in noise $\mathcal{E}$. For instance, we allow binary tensor problem where entries in $\mathcal{Y}$ are $\{0,1\}$-labels from Bernoulli distribution, in which case, the noise variance depends on the mean. Our model (2) is applicable to a wide range of data types including continuous and binary tensors.

We now describe the smooth model on the signal Θ. Suppose that there exists a multivariate function $f\colon [0,1]^m \to \mathbb{R}$ underlying the signal tensor, such that

$$\Theta(i_1,\ldots,i_m) = f\left(\frac{i_1}{d},\ldots,\frac{i_m}{d}\right), \quad \text{for all } (i_1,\ldots,i_m) \in [d]^m. \tag{3}$$

Assume the generative function f is in the α-Hölder smooth family [31, 28].

Definition 1 (α-Hölder smooth)**.** Let $\alpha \geq 0$. A function $f\colon [0,1]^m \to \mathbb{R}$ is α-Hölder smooth, denoted as $f \in \mathcal{H}(\alpha, L)$, if there exists a polynomial function $\text{Poly}_{\lfloor \alpha \rfloor}(\boldsymbol{x} - \boldsymbol{x}_0)$ of degree $\lfloor \alpha \rfloor$, such that

$$|f(\boldsymbol{x}) - \text{Poly}_{\lfloor \alpha \rfloor}(\boldsymbol{x} - \boldsymbol{x}_0)| \leq L \|\boldsymbol{x} - \boldsymbol{x}_0\|_\infty^\alpha, \tag{4}$$

for all $\boldsymbol{x}, \boldsymbol{x}_0 \in [0,1]^m$ and a universal constant $L > 0$.

Hölder smooth function class is one of the most popular function classes considered in the nonparametric regression literature [16, 10]. In addition to the function class $\mathcal{H}(\alpha, L)$, we also define the smooth tensor class based on discretization (3),

$$\mathcal{P}(\alpha, L) = \left\{ \Theta \in \mathbb{R}^{d \times \cdots \times d} \colon \Theta(\omega) = f\left(\frac{\omega}{d}\right) \text{ for all } \omega = (i_1, \ldots, i_m) \in [d]^m \text{ and } f \in \mathcal{H}(\alpha, L) \right\}.$$

Combining (2) and (3) yields our proposed *permuted smooth tensor model*. The unknown parameters are the smooth tensor $\Theta \in \mathcal{P}(\alpha, L)$ and latent permutation $\pi \in \Pi(d, d)$. The generative model is visualized in Figure 1(a) for the case $m = 2$ (matrices). For ease of presentation, we mainly consider the tensor model of equal dimension and same permutations along m modes. The results for non-symmetric tensors with m distinct permutations are similar but require extra notations; we assess this general case in Section 6.

We give two examples to show the applicability of our permuted smooth tensor model.

Example 1 (Four-player game tensors)**.** Consider the tournament of a four-player board game. Suppose there are in total d players, among which all combinations of four have played with each other. The tournament results are summarized as an order-4 (non-symmetric) tensor, with entries encoding the winner out of the four. Our model is then given by

$$\begin{aligned} \mathbb{E}\mathcal{Y}(i_1, \ldots, i_4) &= \mathbb{P}(\text{player } i_1 \text{ wins over } (i_2, i_3, i_4)) \\ &= f\left(\frac{\pi(i_1)}{d}, \cdots, \frac{\pi(i_4)}{d}\right). \end{aligned}$$

In this setting, we can interpret the permutation π as the unknown ranking among d players, and the function f as the unknown four-player interaction. Players with similar ranking may have similar performance reflected by the smoothness of f. For example, a variant of popular Plackett-Luce model [7] considers the parametric form $f(x_1, x_2, x_3, x_4) = \exp(\beta x_1)/\sum_{i=1}^{4}\exp(\beta x_i)$. By contrast, our model leaves the form of f unspecified, and we learn the function from the data in a nonparametric approach.

Example 2 (Co-authorship networks)**.** Consider a co-authorship network consisting of d nodes (authors) in total. We say there exists a hyperedge of size m between nodes $(i_1, \ldots, i_m)$ if the authors $i_1, \ldots, i_m$ have co-authored at least one paper. The resulting m-uniform hypergraph is represented as an order-m (symmetric) adjacency tensor. Our model is then expressed as

$$\begin{aligned}\mathbb{E}\mathcal{Y}(i_1, \ldots, i_m) &= \mathbb{P}(\text{authors } i_1, \ldots, i_m \text{ co-authored})\\ &= f\left(\frac{\pi(i_1)}{d}, \cdots, \frac{\pi(i_m)}{d}\right).\end{aligned}$$

In this setting, we can interpret the permutation π as the affinity measures of authors, and the function f represents the m-way interaction among authors. The parametric model [29] imposes logistic function $f(x_1, \ldots, x_m) = (1 + \exp(-\beta x_1 x_2 \cdots x_m))^{-1}$. By contrast, our nonparametric model allows unknown f and learns the function directly from data.

3 Block-wise tensor estimation

Our general strategy for estimating the permuted smooth tensor is based on the block-wise tensor approximation. In this section, we first introduce the tensor block model [30, 12]. Then, we extend this model to block-wise polynomial approximation.

3.1 Tensor block model

Tensor block models describe a checkerboard pattern in a signal tensor. The block model provides a meta structure to many popular models including the low-rankness [33], latent space models [29], and isotonic tensors [22]. Here, we use tensor block models as a building block for estimating permutated smooth models.

Specifically, suppose that there are k clusters among d entities, and the cluster assignment is represented by a clustering function $z\colon [d] \to [k]$. Then, the tensor block model assumes that the entries of signal tensor $\Theta \in \mathbb{R}^{d\times\cdots\times d}$ take values from a core tensor $\mathcal{S} \in \mathbb{R}^{k\times\cdots\times k}$ according to the clustering function z; that is,

$$\Theta(i_1,\ldots,i_m) = \mathcal{S}(z(i_1),\ldots,z(i_m)), \quad \text{for all } (i_1,\ldots,i_m) \in [d]^m. \tag{5}$$

Here, the core tensor $\mathcal{S}$ collects the entry values of m-way blocks; the core tensor $\mathcal{S}$ and clustering function $z \in \Pi(d,k)$ are parameters of interest. A tensor Θ satisfying (5) is called a block-k tensor, where k is often assumed much smaller than d. Tensor block models allow various data types, as shown below.

Example 3 (Gaussian tensor block model)**.** Let $\mathcal{Y}$ be a continuous-valued tensor. The Gaussian tensor block model draws independent normal entries according to $\mathcal{Y}(i_1,\ldots,i_m) \overset{\text{ind}}{\sim} N(\mathcal{S}(z(i_1),\ldots,z(i_m)),\sigma^2)$. The mean model belongs to (5), and the noise has subGaussian parameter σ^2. The Gaussian tensor block model has served as the statistical foundation for many tensor clustering algorithms [30, 12].

Example 4 (Stochastic tensor block model)**.** Let $\mathcal{Y}$ be a binary-valued tensor. The stochastic tensor block model draws independent Bernoulli entries according to $\mathbb{P}(\mathcal{Y}(i_1,\ldots,i_m) = 1) = \mathcal{S}(z(i_1),\ldots,z(i_m))$. The mean model also belongs to (5), and the noise has sub-Gaussianity parameter σ bounded by $1/4$. The stochastic tensor block model is useful for community detection in multi-relational networks [4, 10].

Tensor block models have shown great success in discovering hidden group structures for many applications [30, 12]. Despite the popularity, the constant block assumption is insufficient to capture delicate structure when the signal tensor is complicated. This parametric model aims to explain data with a finite number of blocks; such an approach is useful when the sample outsizes the parameters. Our nonparametric model (3), by contrast, uses infinite number of parameters (i.e., smooth functions) to allow growing model complexity. Our next section will shift the goal of tensor block model from discovering hidden group structures to approximating the generative function f in (3). In our setting, the number of blocks k should be interpreted as a resolution parameter (i.e., a bandwidth) of the approximation, similar to the notion of number of bins in histogram and polynomial regression [31].

3.2 Block-wise polynomial approximation

The block tensor (5) can be viewed as a discrete version of piece-wise *constant* function with $\alpha = 0$ in (4). This connection motivates us to use block-wise *polynomial* tensors to approximate α-Hölder functions. Now we extend (5) to block-wise polynomial models.

We introduce some additional notations. For a given block number k, we use $z \in \Pi(d, k)$ to denote the canonical clustering function that partitions $[d]$ into k equally-sized clusters such that $z(i) = \lceil ki/d \rceil$, for all $i \in [d]$. The collection of inverse images $\{z^{-1}(j) \colon j \in [k]\}$ is a partition of $[d]$ into k disjoint and equal-sized subsets. We use $\mathcal{E}_k$ to denote the m-way partition, i.e., a collection of k^m disjoint and equal-sized subsets in $[d]^m$, such that

$$\mathcal{E}_k = \{z^{-1}(j_1) \times \cdots \times z^{-1}(j_m) \colon (j_1, \ldots, j_m) \in [k]^m\}.$$

Let $\Delta \in \mathcal{E}_k$ denote the element in $\mathcal{E}_k$. We propose to approximate the signal tensor Θ in (3) by degree-ℓ polynomial tensors within each block $\Delta \in \mathcal{E}_k$. Specifically, let $\mathscr{B}(k,\ell)$ denote the class of block-k degree-ℓ polynomial tensors,

$$\mathscr{B}(k,\ell) = \left\{ \mathcal{B} \in \mathbb{R}^{d\times\cdots\times d} \colon \mathcal{B}(\omega) = \sum_{\Delta \in \mathcal{E}_k} \mathrm{Poly}_{\ell,\Delta}(\omega) \mathbb{1}\{\omega \in \Delta\} \text{ for all } \omega \in [d]^m \right\}, \tag{6}$$

where $\mathrm{Poly}_{\ell,\Delta}(\cdot)$ denotes a degree-ℓ polynomial function in $\mathbb{R}^m$, with coefficients depending on block Δ; that is, a constant function $\mathrm{Poly}_{0,\Delta}(\omega) = \beta^0_\Delta$ for $\ell = 0$, a linear function $\mathrm{Poly}_{1,\Delta}(\omega) = \langle \boldsymbol{\beta}_\Delta, \omega \rangle + \beta^0_\Delta$ for $\ell = 1$, and so on so forth. Here β^0_Δ and $\boldsymbol{\beta}_\Delta$ denote unknown coefficients in polynomial function. Note that the degree-0 polynomial block tensor reduces to the constant block model (5). We generalize the constant block model to degree-ℓ polynomial block tensor (6), in a way that is analogous to the generalization from k-bin histogram to k-piece-wise polynomial regression in nonparametric statistics [31].

Smoothness of the function f in (3) plays an important role in the block-wise polynomial approximation. The following lemma explains the role of smoothness in the approximation.

Lemma 1 (Block-wise polynomial tensor approximation)**.** *Suppose* $\Theta \in \mathcal{P}(\alpha, L)$. *Then, for every block number* $k \leq d$, *and degree* $\ell \in \mathbb{N}_{\geq 0}$, *we have the approximation error*

$$\inf_{\mathcal{B} \in \mathscr{B}(k,\ell)} \frac{1}{d^m} \|\Theta - \mathcal{B}\|_F^2 \lesssim \frac{L^2}{k^{2\min(\alpha,\ell+1)}}.$$

Lemma 1 implies that we can always find a block-wise polynomial tensor close to the signal tensor generated from α-Hölder smooth function f. The approximation error decays with block number k and degree $\min(\alpha, \ell+1)$.

4 Fundamental limits via least-squares estimation

We develop two estimation methods based on the block-wise polynomial approximation. We first introduce a statistically optimal but computationally inefficient least-squares estimator.

The least-squares estimation serves as a statistical benchmark because of its minimax optimality. In Section 5, we will present a polynomial-time algorithm with a provably same optimal rate under monotonicity assumptions.

We propose the least-squares estimation for model (2) by minimizing the Frobounis loss over the block-k degree-ℓ polynomial tensor family $\mathscr{B}(k,\ell)$ up to permutations,

$$(\hat{\Theta}^{\mathrm{LSE}}, \hat{\pi}^{\mathrm{LSE}}) = \underset{\Theta\in\mathscr{B}(k,\ell),\pi\in\Pi(d,d)}{\arg\min} \|\mathcal{Y} - \Theta \circ \pi\|_F. \tag{7}$$

The least-squares estimator $(\hat{\Theta}^{\mathrm{LSE}}, \hat{\pi}^{\mathrm{LSE}})$ depends on two tuning parameters: the number of blocks k and the polynomial degree ℓ. The optimal choice (k^*, ℓ^*) will be provided below.

Our next Theorem 1 establishes the error bound for the least-squares estimator (7). Note that Θ and π are in general not separably identifiable; for example, when the true signal is a constant tensor, then every permutation $\pi \in \Pi(d,d)$ gives equally good fit in statistics. We assess the estimation error on the composition $\Theta \circ \pi$ to avoid this identifiability issue. For two order-m d-dimensional tensors Θ_1, Θ_2, define the mean squared error (MSE) by $\mathrm{MSE}(\Theta_1, \Theta_2) = d^{-m}\|\Theta_1 - \Theta_2\|_F^2$.

Theorem 1 (Least-squares estimation error)**.** *Consider the order-m ($m \geq 2$) permuted smooth tensor model* (2) *with $\Theta \in \mathcal{P}(\alpha, L)$. Let $(\hat{\Theta}^{\mathrm{LSE}}, \hat{\pi}^{\mathrm{LSE}})$ denote the least-squares estimator in* (7) *with a given (k,ℓ). Then, for every $k \leq d$ and degree $\ell \in \mathbb{N}_{\geq 0}$, we have*

$$\mathrm{MSE}(\hat{\Theta}^{\mathrm{LSE}} \circ \hat{\pi}^{\mathrm{LSE}},\ \Theta\circ\pi) \lesssim \underbrace{\frac{L^2}{k^{2\min(\alpha,\ell+1)}}}_{\text{approximation error}} + \sigma^2 \left(\underbrace{\frac{k^m(\ell+m)^\ell}{d^m}}_{\text{nonparametric error}} + \underbrace{\frac{\log d}{d^{m-1}}}_{\text{permutation error}} \right), \tag{8}$$

with high probability. In particular, setting $\ell^* = \min(\lfloor\alpha\rfloor, (m-2)(m+1)/2)$ and $k^* = c_1 d^{m/(m+2\min(\alpha,\ell^*+1))}$ yields the optimized error rate

$$(8) \lesssim \begin{cases} c_2 d^{-\frac{2m\alpha}{m+2\alpha}}, & \textit{when } \alpha < m(m-1)/2, \\ c_3 d^{-(m-1)}\log d, & \textit{when } \alpha \geq m(m-1)/2. \end{cases} \tag{9}$$

Here, the constants $c_1, c_2, c_3 > 0$ depend on the model configuration (m, σ, L, α) but not on the tensor dimension d. The closed-form expressions are provided in Appendix B.2.

We discuss the asymptotic error rate as $d \to \infty$ while treating other model configurations fixed. The final least-squares estimation rate (9) has two sources of error: the nonparametric error $d^{-\frac{2m\alpha}{m+2\alpha}}$ and the permutation error $d^{-(m-1)} \log d$. Intuitively, in the tensor data analysis problem, we can view each tensor entry as a data point, so sample size is the total number of entries, d^m. The unknown permutation results in $\log(d!) \approx d \log d$ complexity, whereas the unknown generative function results in $d^{-2m\alpha/(m+2\alpha)}$ nonparametric complexity. When the function f is smooth enough, estimating the function f becomes relatively easier compared to estimating the permutation π. This intuition coincides with the fact that the permutation error dominates the nonparametric error when $\alpha \geq m(m-1)/2$.

We now compare our results with existing work in the literature.

Remark 1 (Comparison to non-parametric regression)**.** In the vector case with $m = 1$, our model reduces to the one-dimensional regression problem such that

$$y_i = \theta_{\pi(i)} + \epsilon_i, \qquad \text{for all } i \in [d],$$

where $\theta_i = f(i/d)$ and unknown $\pi \in \Pi(d, d)$. A similar analysis of our Theorem 1 shows the error rate

$$\frac{1}{d} \sum_{i \in [d]} (\hat{\theta}_i^{\text{LSE}} - \theta_i)^2 \lesssim \left(d^{-\frac{2\alpha}{2\alpha+1}} + \log d \right), \tag{10}$$

under the choice of $\ell^* = 0$ and $k^* \asymp d^{\frac{1}{1+2\min(\alpha,1)}}$. Notice that $d^{-2\alpha/(2\alpha+1)}$ is the classical nonparametric minimax rate for α-Hölder smooth functions [28] with *known* permuted design points $\{\pi(i)\}_{i=1}^d$. By contrast, our model involves *unknown* π, which results in the non-vanishing permutation rate $\log d$ in (10).

Remark 2 (Breaking previous limits on matrices/tensors)**.** In the matrix case with $m = 2$, Theorem 1 implies that the best rate is obtained under $\ell^* = 0$, i.e., the block-wise *constant* approximation. This result is consistent with existing literature on smooth graphons [4, 10, 16], where constant block model (see Section 3.1) has been develop for accurate estimation.

Earlier work [1] suggests that constant block approximation ($\ell^* = 0$) remains minimax optimal for tensors of order $m \geq 3$. Our Theorem 1 disproves this conjecture, and we reveal a much faster rate $d^{-(m-1)}$ compared to the conjectured lower bound $d^{-2m/(m+2)}$ [1] for sufficiently smooth tensors. We demonstrate that a polynomial up to degree $(m-2)(m+1)/2$ is sufficient (and necessary; see Theorem 2 below) for accurate estimation of order-m permuted smooth tensors. For example, permuted α-smooth tensors of order-3 require quadratic approximation ($\ell^* = 2$) with $k^* \asymp d^{1/3}$ blocks, for all $\alpha \geq 2$. The results show the clear difference from matrices and highlight the challenges with tensors.

We now show that the rate in (9) cannot be improved. The lower bound is obtained via information-theoretical analysis and thus applies to all estimators including, but not limited to, the least-squares estimator (7) and Borda count estimator introduced in next section.

Theorem 2 (Minimax lower bound)**.** *For any given $\alpha \geq 0$, the estimation problem based on model* (1) *obeys the minimax lower bound*

$$\inf_{(\hat{\Theta},\hat{\pi})} \sup_{\Theta\in\mathcal{P}(\alpha,L),\pi\in\Pi(d,d)} \mathbb{P}\left(\text{MSE}(\hat{\Theta}\circ\hat{\pi},\ \Theta\circ\pi) \gtrsim c_2 d^{-\frac{2m\alpha}{m+2\alpha}} + c_3 d^{-(m-1)}\log d\right) \geq 0.8, \qquad (11)$$

where $c_2, c_3 > 0$ are the same constants as in Theorem 1.

The lower bound in (11) matches the upper bound in (9), demonstrating the statistical optimality of least-squares estimator (7). The two-component error reveals the intrinsic model complexity: the permutation error $d^{-(m-1)}$ dominates nonparametric error $d^{-2m\alpha/(m+2\alpha)}$ for sufficiently smooth tensors. This is clearly contrary to classical nonparametric regression.

Remark 3 (Phase transition)**.** We conclude this section by presenting an interesting phase transition phenomenon. Figure 1(b) plots the convergence rate of estimation error based on Theorems 1-2. We find that the impact of unknown permutation hinges on the tensor order and smoothness. The accuracy improves with smoothness in the regime $\alpha \leq m(m-1)/2$, but then it becomes a constant of smoothness in the regime $\alpha > m(m-1)/2$. The result implies a polynomial of degree $\approx (m-2)(m+1)/2$ is sufficient for accurate recovery of order-m tensors, whereas higher degree brings no further benefits. This full picture of error dependence, to our best knowledge, is new to the literature of permuted smooth tensors.

5 An efficient and computationally feasible procedure

At this point, we should point out that computing the least-squares optimizer $(\hat{\Theta}^{\text{LSE}}, \hat{\pi}^{\text{LSE}})$ in (7) is generally computationally hard, even in the simple matrix case [10]. In this section, we propose an efficient polynomial-time *Borda count* algorithm. We show that Borda count estimator provably achieves the same convergence rate as the minimax lower bound (11) under monotonicity assumptions.

5.1 Borda count algorithm

We introduce a notion of β-monotonicity for the generative functions.

Definition 2 (Weakly β-monotonicity)**.** A function $f\colon [0,1]^m \to \mathbb{R}$ is called β-monotonic ($\beta \geq 0$), denoted as $f \in \mathcal{M}(\beta)$, if there exists a small tolerance $\delta \lesssim d^{-(m-1)/2}$ such that

$$\left(\frac{i-j}{d}\right)^{1/\beta} \lesssim g(i) - g(j) + \delta, \qquad \text{for all } j < i \in [d], \tag{12}$$

where we define *score function* $g(i) = d^{-(m-1)} \sum_{(i_2,\ldots,i_m)\in[d]^{m-1}} f\left(\frac{i}{d}, \frac{i_2}{d}, \ldots, \frac{i_m}{d}\right)$ for all $i \in [d]$.

Our β-monotonicity condition extends the strictly monotonic degree condition in the graphon literature [5]; the latter is a special case of our definition with $\beta = 1, m = 2$ and $\delta = 0$. A large value of β in (12) implies the steepness of g. The introduction of tolerance δ relaxes the condition by allowing for small fluctuations. Our β-monotonicity condition is also related to isotonic functions [22] which consider more restricted coordinate-wise monotonicity; i.e., $f(x_1, \ldots, x_d) \leq f(x'_1, \ldots, x'_d)$ whenever $x_i \leq x'_i$ for $i \in [d]$.

The β-monotonicity condition allows us to efficiently estimate the permutation π. Before presenting the theoretical guarantees, we provide the intuition here. The exponent β measures the difficulty for estimating the permutation π. Consider the noisy observation $\mathcal{Y}$ from model (1). We define the empirical score function $\tau \colon [d] \to \mathbb{R}$ as

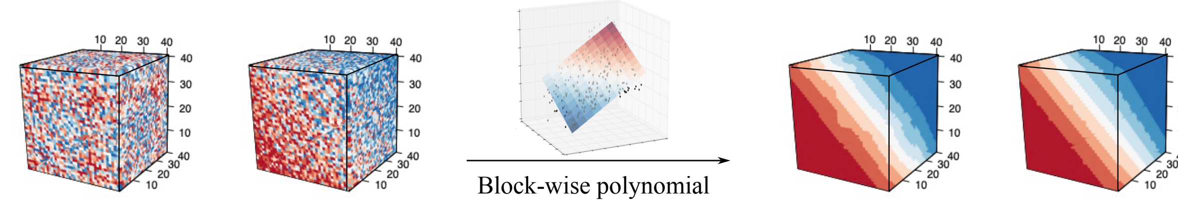
$$\tau(i) = \frac{1}{d^{m-1}} \sum_{(i_2,\ldots,i_m)\in[d]^m} \mathcal{Y}(i, i_2, \ldots, i_m).$$

The permuted score function $\tau \circ \pi^{-1}$ reduces to the function g in (12) under the noiseless setting. Therefore, a good estimate $\hat{\pi}$ should make the permuted score function $\tau \circ \hat{\pi}^{-1}$ monotonically increasing. Notice that the estimated permutation $\hat{\pi}$ could be different from the oracle permutation π due to the noise. We find that a larger β guarantees a faster consistency rate of $\hat{\pi}$. A large β implies large gaps of $|g(i) - g(j)|$ for $i \neq j$. Therefore, we obtain similar orderings of $\{\tau(i)\}_{i=1}^d$ before and after the addition of noise. This intuition is well represented by the following lemma.

Lemma 2 (Permutation error)**.** *Consider the permuted smooth tensor model with $f \in \mathcal{M}(\beta)$. Let $\hat{\pi}$ be the permutation such that the permuted empirical score function $\tau \circ \hat{\pi}^{-1}$ is monotonically increasing. Then, with high probability,*

$$\text{Loss}(\pi, \hat{\pi}) := \frac{1}{d} \max_{i\in[d]} |\pi(i) - \hat{\pi}(i)| \lesssim \left(\sigma d^{-(m-1)/2}\sqrt{\log d}\right)^{\beta}.$$

Now we introduce the *Borda count* estimation that consists of two stages. The full estimation procedure is illustrated in Figure 2.



(a) Observation (b) Sorted observation (c) Estimated signal (d) True signal

Figure 2: Illustration of Borda count estimation. We first sort tensor entries using the proposed procedure, and then estimate the signal by block-wise polynomial approximation.

1. **Sorting stage:** The purpose of the sorting is to rearrange the observed tensor $\mathcal{Y}$ so that the score function τ of sorted tensor is monotonically increasing. We define a permutation $\hat{\pi}^{\mathrm{BC}}$ such that

$$\tau((\hat{\pi}^{\mathrm{BC}})^{-1}(1)) \leq \tau((\hat{\pi}^{\mathrm{BC}})^{-1}(2)) \leq \cdots \leq \tau((\hat{\pi}^{\mathrm{BC}})^{-1}(d)). \tag{13}$$

Then, we obtain sorted observation $\tilde{\mathcal{Y}} = \mathcal{Y} \circ (\hat{\pi}^{\mathrm{BC}})^{-1}$, illustrated in Figure 2(b).

2. **Block-wise polynomial approximation stage:** Given sorted observation $\tilde{\mathcal{Y}}$, we estimate the signal tensor by block-wise polynomial tensor based on the following optimization,

$$\hat{\Theta}^{\mathrm{BC}} = \operatorname*{arg\,min}_{\Theta \in \mathscr{B}(k,\ell)} \|\tilde{\mathcal{Y}} - \Theta\|_F, \tag{14}$$

where $\mathcal{B}(k,\ell)$ denotes the block-k degree-ℓ tensor class in (6). An example of this procedure is shown in Figure 2(c). The estimator $\hat{\Theta}^{\mathrm{BC}}$ depends on two tuning parameters: the number of blocks k and polynomial degree ℓ. The optimal choice of (k^*, ℓ^*) is provided in Theorem 3. Notice that the least-squares estimator in (7) requires a combinatoric search with exponential-time complexity for estimating the permutation. By contrast, the estimator (14) requires only the estimation of degree-ℓ polynomial within k canonical blocks. Therefore, the Borda count estimator is polynomial-time efficient. Our algorithm implementation is available in CRAN.

5.2 Computational and statistical complexities

In this section, we show that the Borda count estimation achieves both computational efficiency and satanical accuracy.

The computational complexity of Borda count estimation is polynomial in tensor dimension d. In the sorting stage, computing the empirical score function τ requires $\mathcal{O}(d^{m-1})$ operations while sorting the $\{\tau(i)\}_{i=1}^d$ requires $\mathcal{O}(d\log d)$ comparisons. In the block-wise polynomial approximation stage, we compute k^m many degree-ℓ polynomial tensors. Each polynomial tensor approximation requires $\mathcal{O}((d/k)^m\ell)$ arithmetic operations. Thus, the second step requires $\mathcal{O}(d^m\ell)$ operations. Combining the two steps yields the total complexity at most $\mathcal{O}(d^m\log d)$. This complexity is comparable with existing efficient tensor estimation algorithms in other settings [18, 34].

The following theorem ensures the statistical accuracy of the Borda count estimator.

Theorem 3 (Estimation error for Borda count algorithm)**.** *Consider the permuted smooth tensor model with $f\in\mathcal{H}(\alpha,L)\cap\mathcal{M}(\beta)$. Let $(\hat{\Theta}^{\rm BC},\hat{\pi}^{\rm BC})$ be the Borda count estimator in* (13)-(14) *with a given (k,ℓ). Then, for every $k\leq d$ and degree $\ell\in\mathbb{N}_{\geq 0}$, we have*

$$\mathrm{MSE}(\hat{\Theta}^{\rm BC}\circ\hat{\pi}^{\rm BC},\ \Theta\circ\pi)\lesssim\frac{L^2}{k^{2\min(\alpha,\ell+1)}}+\sigma^2\frac{k^m(\ell+m)^\ell}{d^m}+\left(\sigma^2\frac{\log d}{d^{m-1}}\right)^{\beta\min(\alpha,1)},\quad (15)$$

with high probability. Furthermore, denote a constant $c(\alpha,\beta,m):=\frac{m(m-1)\beta\min(\alpha,1)}{\max(0,\ 2(m-(m-1)\beta\min(\alpha,1)))}$. Then, setting $\ell^*=\min\left(\lfloor\alpha\rfloor,\lfloor c(\alpha,\beta,m)\rfloor\right)$ and $k^*=c_1d^{m/(m+2\min(\alpha,\ell^*+1))}$ yields

$$(15)\ \lesssim\begin{cases}c_2d^{-\frac{2m\alpha}{m+2\alpha}} & \text{when } \alpha<c(\alpha,\beta,m),\\ \left(\frac{c_3\log d}{d^{m-1}}\right)^{\beta\min(\alpha,1)} & \text{when } \alpha\geq c(\alpha,\beta,m),\end{cases}\quad (16)$$

where $c_1,c_2,c_3>0$ are the same constants as in Theorem 1.

Remark 4 (Sufficiently smooth tensors)**.** When the generative function is infinitely smooth

($\alpha = \infty$) with Lipschitz monotonic score ($\beta = 1$), our estimation error (16) becomes

$$\text{MSE}(\hat{\Theta}^{\text{BC}} \circ \hat{\pi}^{\text{BC}},\ \Theta \circ \pi) \lesssim d^{-(m-1)} \log d, \tag{17}$$

under the choice of degree and block number

$$\ell^* = (m-2)(m+1)/2 \quad \text{and} \quad k^* \asymp d^{\frac{m}{m+2\ell^*}}.$$

Now, we compare the rate (17) with the classical low-rank estimation [29, 34, 17]. The low-rank tensor model with a constant rank is known to have MSE rate $\mathcal{O}(d^{-(m-1)})$ [29]. Our infinitely smooth tensor model achieves the nearly same rate up to the negligible log term. Compared to low-rank models, we utilize a different measure of *model complexity*. When the underlying signal is precisely low-rank, then rank might be a reasonable measure for model complexity. However, if the underlying signal is high rank but has certain shape structure, then our nonparametric approach may better capture the intrinsic model complexity.

Remark 5 (Comparison with least-squares estimation)**.** The three terms in the estimation bound (15) correspond to approximation error (Lemma 1), nonparametric error (Theorem 1), and permutation error (Lemma 2), respectively. We find that the Borda count estimator achieves the same minimax-optimal rate as the least-squares estimator for sufficiently smooth tensors under Lipschitz score condition $\beta = 1$. The least-squares estimator requires a combinatoric search with exponential-time complexity. By contrast, the Borda count estimator is polynomial-time solvable. Therefore, Borda count algorithm enjoys both statistical accuracy and computational efficiency.

Hyperparameter tuning. Our algorithm has two tuning parameters (k, ℓ). The theoretically optimal choices of (k, ℓ) are given in Theorems 1 and 3. In practice, since model configuration is unknown, we search (k, ℓ) via cross-validation. Based on our theorems, a polynomial of degree $\ell^* = (m-2)(m+1)/2$ is sufficient for accurate recovery of order-m tensors, whereas higher degree brings no further benefit. The practical impacts of hyperparameter tuning are investigated in Section 6.

6 Numerical analysis

6.1 Synthetic data

We simulate order-3 d-dimensional tensors based on the permuted smooth tensor model (3). Both symmetric and non-symmetric tensors are investigated. The symmetric tensors are generated based on functions f in Table 2, and the non-symmetric set-up is described in Appendix A.2. The generative functions involve compositions of operations such as

Model ID	$f(x,y,z)$	CP rank	Tucker rank
1	xyz	1	$(1,1,1)$
2	$(x+y+z)/3$	3	$(2,2,2)$
3	$(1+\exp(-3x^2+3y^2+3z^2))^{-1}$	9	$(4,4,4)$
4	$\log(1+\max(x,y,z))$	≥ 100	$\geq (50,50,50)$
5	$\exp\left(-\max(x,y,z)-\sqrt{x}-\sqrt{y}-\sqrt{z}\right)$	≥ 100	$\geq (50,50,50)$

Table 2: Smooth functions in simulation. We define the numerical CP/Tucker rank as the minimal rank r for which the relative approximation error is below 10^{-4}. The reported rank in the table is estimated from a $100 \times 100 \times 100$ signal tensor generated by (3).

polynomial, logarithm, exponential, square roots, etc. Notice that considered functions cover a reasonable range of model complexities from low rank to high rank. Two types of noise are considered: Gaussian noise and Bernoulli noise. For the Gaussian model,

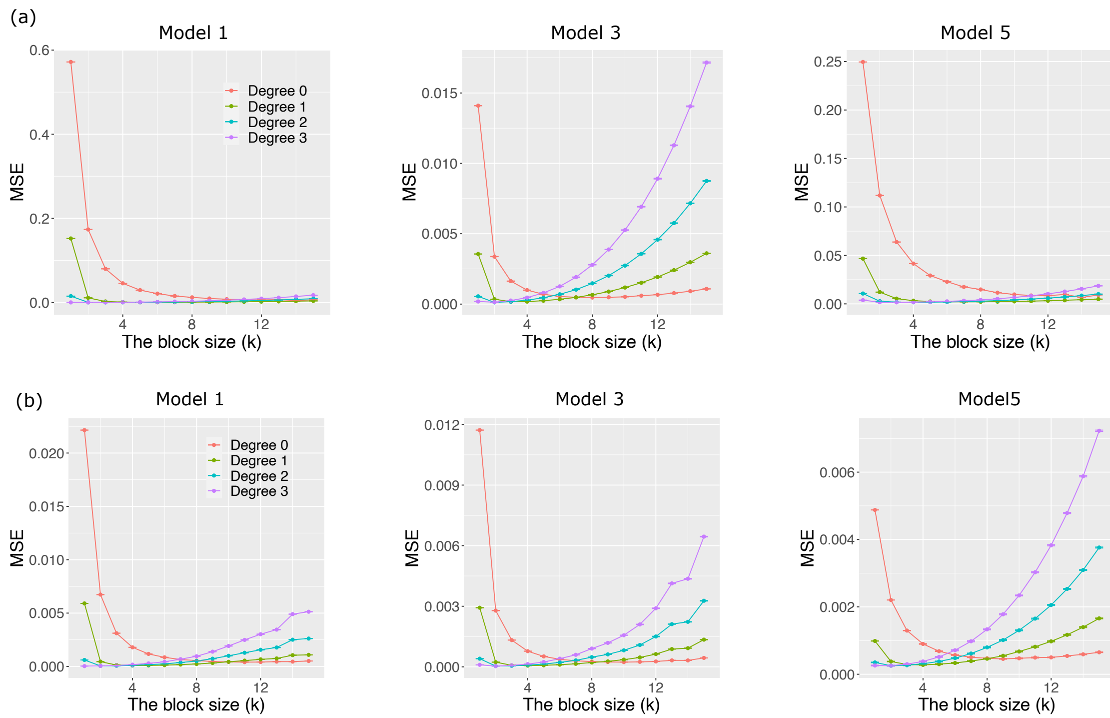


Figure 3: MSE versus the number of blocks based on different polynomial approximations. Columns 1-3 consider the Models 1, 3, and 5 respectively. Panel (a) is for continuous tensors, whereas (b) is for the binary tensors.

we simulate continuous-valued tensors with i.i.d. noises drawn from $N(0, 0.5^2)$. For the Bernoulli model, we generate binary tensors $\mathcal{Y}$ using the success probability tensor $\Theta \circ \pi$. The permutation π is randomly chosen. For space consideration, only results for Models 1, 3, and 5 are presented in the main paper. The rest is presented in Appendix A.1. We first examine impacts of model complexity to estimation accuracy. We then compare Borda count estimation with alternative methods under a range of scenarios. Extra simulation results and extensions are deferred to Appendix.

Impacts of the number of blocks, tensor dimension, and polynomial degree. The first experiment examines the impact of the block number k and degree of polynomial ℓ for the approximation. We fix the tensor dimension $d = 100$, and vary the number of blocks $k \in \{1, \ldots, 15\}$ and polynomial degree $\ell \in \{0, 1, 2, 3\}$. Figure 3 demonstrates the trade-off in accuracy determined by the number of groups for each polynomial degree. The results confirm our bias-variance analysis in Theorem 1. While a large block number k provides less biased approximation, this large k renders the signal tensor estimation difficult within each block due to small sample size. In addition, we find that degree-2 polynomial approximation with the optimal k gives the smallest MSE among all considered polynomial approximations. These observations are consistent with our theoretical results that the optimal number of blocks and polynomial degree are $(k^*, \ell^*) = (\mathcal{O}(d^{3/7}), 2)$.

The second experiment investigates the impact of the tensor dimension d for various polynomial degrees. We vary the tensor dimension $d \in \{10, \ldots, 100\}$ and polynomial degree $\ell \in \{0, 1, 2, 3\}$ in each model configuration. We set optimal number of blocks as the one that gives the best accuracy. Figure S1 compares the estimation errors among different polynomial approximations. The result verifies that the degree-2 polynomial approximation performs the best under the sufficient tensor dimension, which is consistent with our theoretical results. We emphasize that this phenomenon is different from the matrix case where the degree-0 polynomial approximation gives the best results [10, 16].

Comparison with alternative methods. We compare our method (**Borda Count**) with several popular alternative methods.

- Spectral method (**Spectral**) [32] that performs universal singular value thresholding [6] on the unfolded tensor.
- Least-squares estimation (**LSE**) [10] which solves the optimization problem (7) with constant block approximation ($\ell = 0$) based on spectral k-means. We extend the matrix-based biclustering algorithm to higher-order tensors [12].
- Least-squares estimation (**BAL**) [1] which solves the optimization problem (7) with

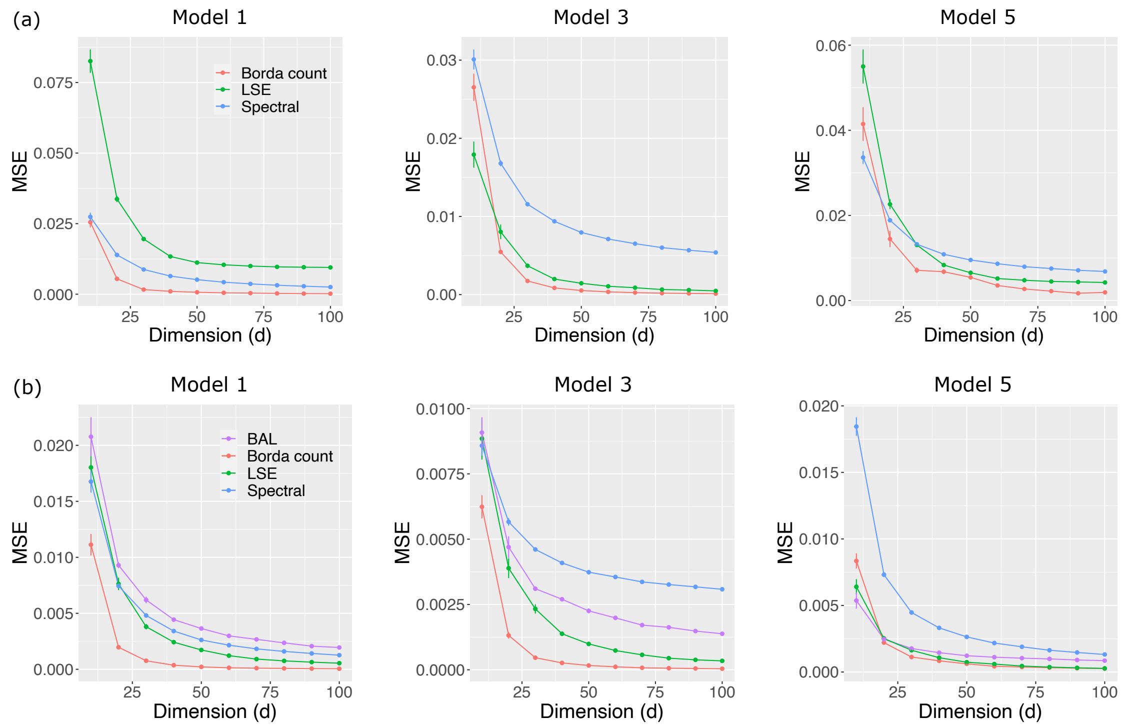


Figure 4: MSE versus the tensor dimension based on different estimation methods. Columns 1-3 consider the Models 1, 3, and 5 in Table 2 respectively. Panel (a) is for continuous tensors, whereas (b) is for the binary tensors.

constant block approximation ($\ell = 0$). This tensor-based algorithm is only available for binary observations because it uses count-based statistics. Therefore, we only use this algorithm for the Bernoulli model.

We choose degree-2 polynomial approximation as our theorems suggested, and vary tensor dimension $d \in \{10, \dots, 100\}$ under each model configuration. For **Borda Count** and **LSE**, we choose the block numbers that achieve the best performance in the corresponding outputs. For **Spectral** method, we set the hyperparameter (singular-value threshold) that gives the best performance.

Figure 4 shows that our algorithm **Borda Count** achieves the best performance in all scenarios as the tensor dimension increases. The poor performance of **Spectral** can be explained by the loss of multilinear structure in the tensor unfolding procedure. The

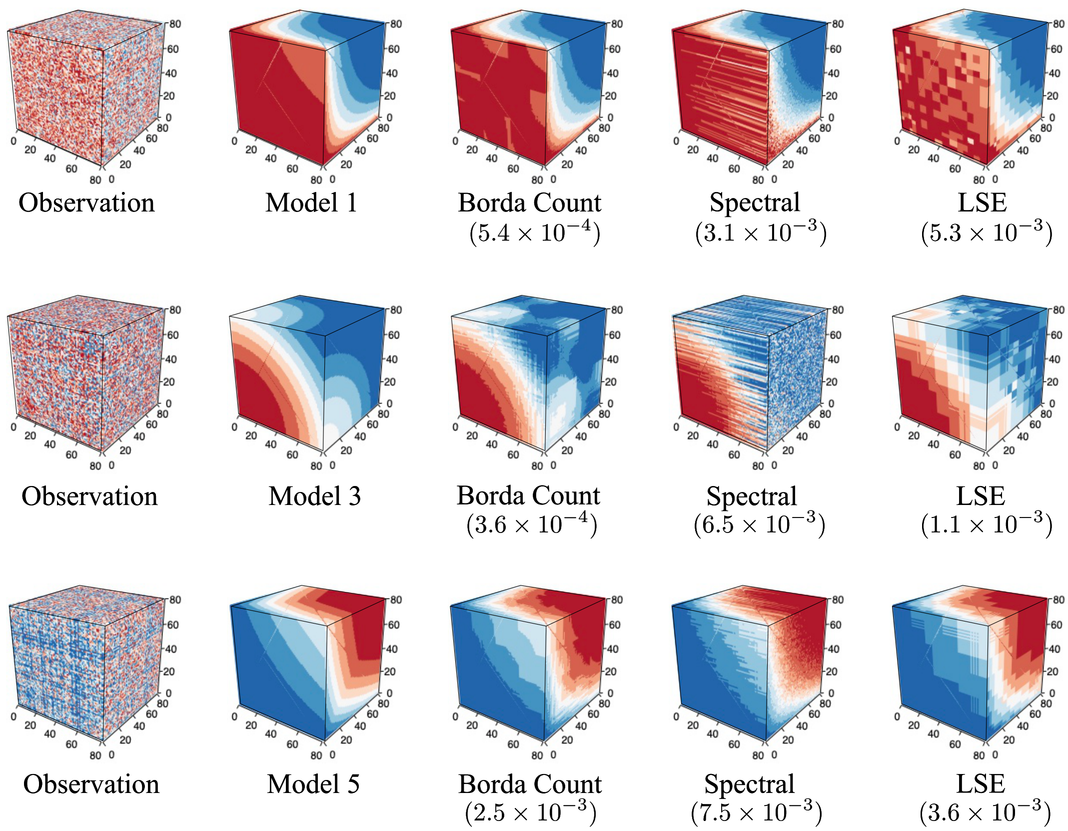


Figure 5: Performance comparison among different methods. The observed data tensors, true signal tensors, and estimated signal tensors are plotted for Models 1, 3 and 5 in Table 2 with fixed dimension $d = 80$. Numbers in parenthesis indicate the mean squared error.

sub-optimality of **LSE** is possibly due to its limits in both statistics and computations. Statistically, our theorems have shown that constant block approximation results in sub-optimal rates compared to polynomial approximation. Computationally, the least-squares optimization (7) is highly non-convex and computationally unstable. Figure 5 displays true signal tensors of three models and corresponding observed tensors of dimension $d = 80$ with Gaussian noise. We use oracle permutation π to obtain the estimated signal tensor from the estimated permuted signal tensor $\hat{\Theta} \circ \hat{\pi}$ for the better visualization and comparisons. As shown in the figure, we see clearly that our method achieves the best signal recovery, thereby supporting the numerical results in Figure 4. The outperformance of **Borda count** demonstrates the efficacy of our method.

Investigation of non-symmetric tensors. Our models and techniques easily extend to non-symmetric tensors. We use non-symmetric functions to generate order-3 signal tensors; see detailed setup in Appendix A.2. We choose hyperparameters that give the best accuracy for each method (see Table S2). Table 3 compares the MSEs from repeated simulations based on different methods under Models 1-5 (see Table S1). We find that Borda count estimation outperforms all alternative methods for non-symmetric tensors. The results demonstrate the applicability of our method to general tensors.

Method	Model 1	Model 2	Model 3	Model 4	Model 5
Borda count	**0.57 (0.01)**	**0.51 (0.02)**	**0.87 (0.02)**	**1.02 (0.02)**	**2.56 (0.21)**
LSE	23.58 (0.03)	7.70 (0.04)	9.45 (0.05)	3.29 (0.05)	9.93 (0.03)
Spectral	10.76 (0.06)	10.64 (0.05)	6.27 (0.05)	10.90 (0.06)	5.24 (0.04)

Table 3: MSEs from 20 repeated simulations based on different methods. All numbers are displayed on the scales 10^{-3}. Standard errors are reported in parenthesis.

6.2 Applications to Chicago crime data

Chicago crime dataset consists of crime counts reported in the city of Chicago, ranging from January 1st, 2001 to December 11th, 2017. The observed tensor is an order-3 tensor with entries representing the log counts of crimes from 24 hours, 77 community areas, and 32 crime types. We apply our Borda count method to Chicago crime dataset. Because the data tensor is non-symmetric, we allow different number of blocks across the three modes. Cross validation result suggests the $(k_1, k_2, k_3) = (6, 4, 10)$, representing the block number for crime hours, community areas, and crime types, respectively.

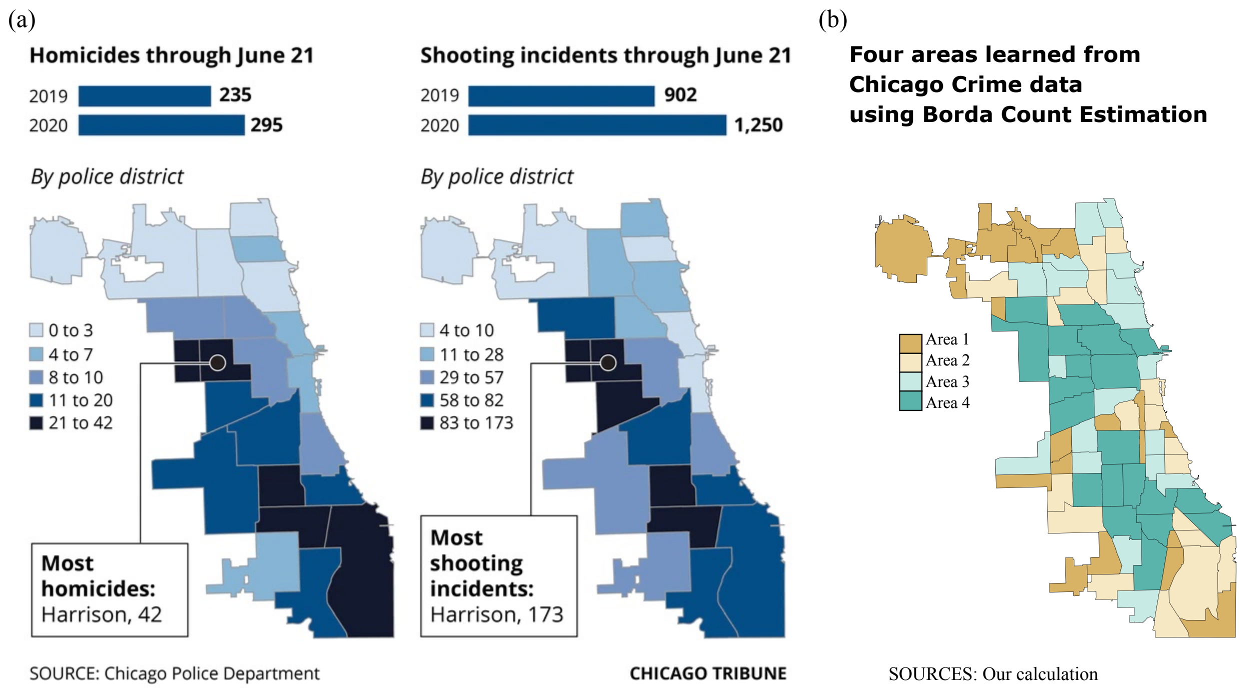


Figure 6: Chicago crime maps. Figure(a) is the benchmark map based on homicides and shooting incidents in community areas in Chicago [15]. Figure(b) shows the four clustered areas learned from 32 crime types using our method.

We first investigate the four community areas obtained from our Borda count algorithm. Figure 6(b) shows the four areas overlaid on the Chicago map. Interestingly, we find that the clusters are consistent with actual locations, even though our algorithm did not take any geographic information such as longitude or latitude as inputs. In addition, we compare the cluster patterns with benchmark maps based on homicides and shooting incidents in Chicago shown in Figure 6(a). We find that our clusters share similar geographical patterns with Figure 6(a). The results demonstrate the power of our approach in detecting meaningful pattern from tensor data.

Then, we examine the denoised signal tensor obtained from our method and analyze the trends between crime types and crime hours by the four community areas in Figure 6(b). Figure 7 shows the averaged log counts of crimes according to crime types and crime hours

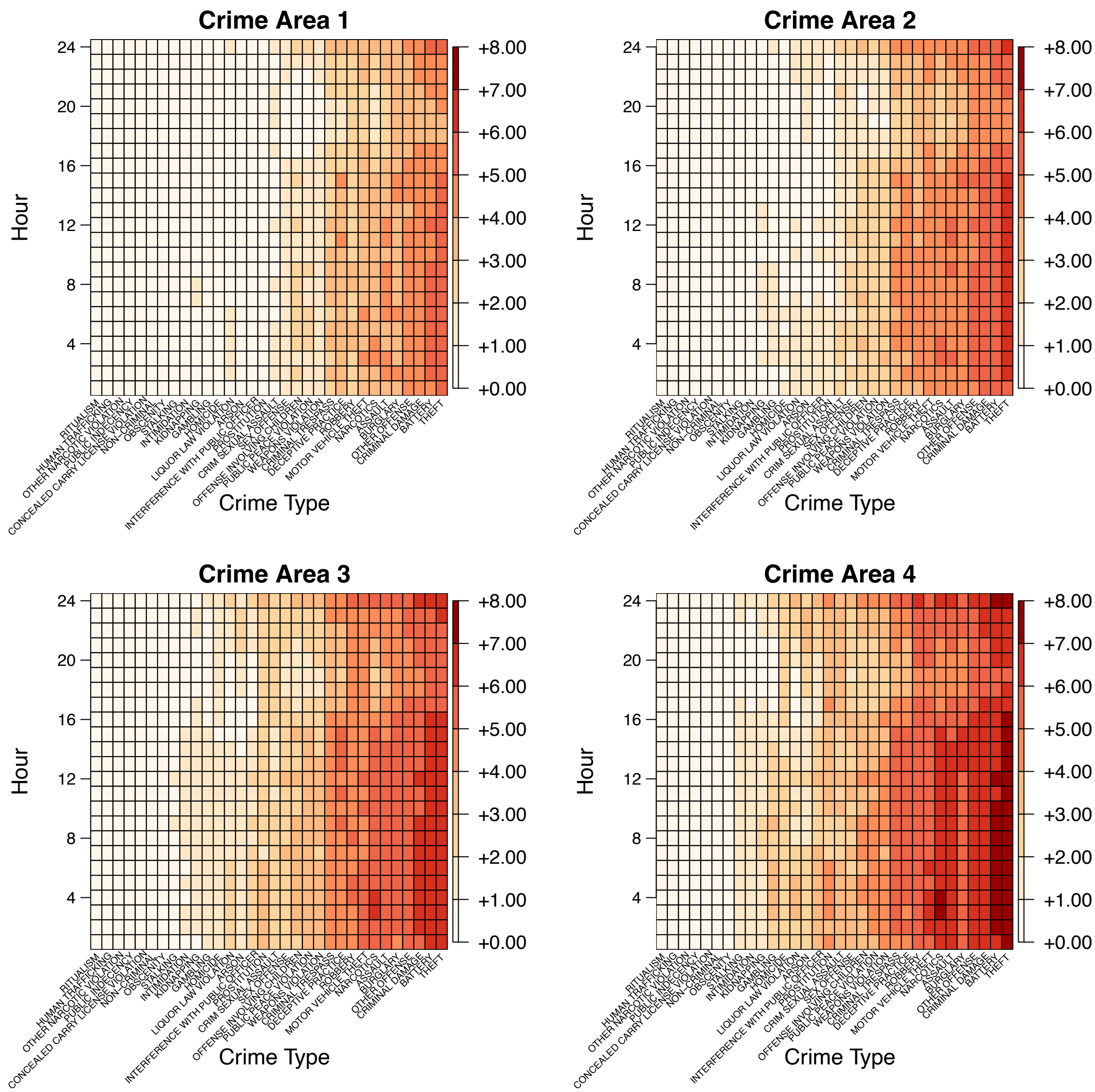


Figure 7: Averaged log counts of crimes according to crime types, hours, and the four areas estimated by our Borda count algorithm. We plot the estimated signal tensor entries averaged within four areas in the heatmap.

by four areas. We find that the major difference among four areas is the crime rates. Area 4 has the highest crime rates, and the crime rates monotonically decrease from Area 4 to Area 1. The variation in crime rates across hour and type, nevertheless, exhibits similarity among the four areas. For example, Figure 7 shows that the number of crimes increases hourly from 8 p.m., peaks at night hours, and then drops to the lowest at 6 p.m. The identified similarities and differences among the four community areas highlight the interpretability of our method in real data.

	Constant block model	Permuted smooth tensor model
MSE	0.399 (0.009)	0.283 (0.006)
Block number	$(7, 11, 10)$	$(6, 4, 10)$

Table 4: Performance comparison in Chicago data analysis. Reported MSEs are averaged over five runs of cross-validation, with 20% entries for testing and 80% for training, with standard errors in parentheses. Block number is set to achieve the best prediction performance.

Finally, we compare the prediction performance based on constant block model and our permuted smooth tensor model. Notice that constant block model uses $\ell = 0$ approximation, whereas our permuted smooth tensor model uses $\ell = 2$ approximation. Table 4 shows the mean squared error over five runs of cross-validation, with 20% entries for testing and 80% for training. We find that the permuted smooth tensor model substantially outperforms the classical constant block models. We emphasize that our method does not necessarily assume the block structure. The comparison supports our premises that permuted smooth tensor model with polynomial approximation performs better than common constant block models in this application.

7 Conclusion and Discussions

We have presented a suite of statistical theory, estimation methods, and data applications for permuted smooth tensor models. Two estimation algorithms are proposed with accuracy guarantees: the (statistically optimal) least-squares estimation and the (computationally tractable) Borda count estimation. In particular, we establish an interesting phase transition phenomenon with respect to the critical smoothness level. We demonstrate that a block-wise polynomial of order $(m-2)(m+1)/2$ is sufficient and necessary for accurate recovery of order-m tensors, in contrast to earlier beliefs on constant block approximation. Experiments demonstrate the effectiveness of both theoretical findings and algorithms.

There are several possible extensions from our work. The theory in this paper assumes symmetry on the signal tensor Θ for simplicity of exposition. In fact, all our results naturally extend to non-symmetric signal tensors. A non-symmetric tensor $\Theta \in \mathbb{R}^{d_1 \times \cdots \times d_m}$ can be represented by $\Theta(i_1, \ldots, i_m) = f\left(\frac{\pi_1(i_1)}{d_1}, \ldots, \frac{\pi_m(i_m)}{d_m}\right)$, where $\pi_\ell \colon [d_\ell] \to [d_\ell]$ is the latent permutation for each mode $\ell \in [m]$, and the function f is a smooth but non-symmetric function. Under the condition that $d_1, \ldots, d_m$ are asymptotically of the same order, similar estimation algorithms and theoretical accuracy guarantees still hold true.

Our framework of block-wise polynomial approximation can be extended to allow other nonparametric techniques, including B splines, smoothing splines, kernel regression, wavelets, etc. We choose to use polynomial basis because of its simplicity. The parsimony allows us to establish the insights on critical smoothness level $(m-2)(m+1)/2$. For example, our result suggests that quadratic splines are enough for accurate estimation of order-3 tensors. One can combine our approach with the modern trend filtering techniques [27, 21],

$$\hat{f} = \arg\min_{f} \sum_{(i_1,i_2,i_3)} \left(\mathcal{Y}(\hat{\pi}(i_1), \hat{\pi}(i_2), \hat{\pi}(i_3)) - f\left(\frac{i_1}{d}, \frac{i_2}{d}, \frac{i_3}{d}\right)\right)^2 + \lambda \left\|\boldsymbol{\nabla}_2 f\right\|,$$

where $\|\boldsymbol{\nabla}_2 f\|$ represents total variation of second order difference. The case of $m = 2$ (matrix) reduces to the total variation smoothing in graphons [5]. For general order-m tensors, our theory provides a principle of guidance for the order of smoothness needed. Exploiting the benefits and properties of various nonparametric fitting techniques for general tensor models warrants future research.

Finally, our current approach assumes no randomness in the signal tensor Θ. One can also extend the generative model to allow random designs, where the signal tensor is represented by $\Theta(i_1, \ldots, i_m) = f(x_{i_1}, \ldots, x_{i_m})$ with $(x_i)_{i=1}^d$ i.i.d. randomly drawn from certain distribution. Similar techniques have been developed for graphons and hypergraphons [5, 10, 16, 1]. The two choices of designs lead to different analysis in the same spirit as random- vs. fixed-designs in nonparametric regression. Extending our theory to random design is an interesting question for future research.

References

[1] Balasubramanian, K. (2021). Nonparametric modeling of higher-order interactions via hypergraphons. *Journal of Machine Learning Research 22*, 1–25.

[2] Baltrunas, L., M. Kaminskas, B. Ludwig, O. Moling, F. Ricci, A. Aydin, K.-H. Lüke, and R. Schwaiger (2011). InCarMusic: Context-aware music recommendations in a car. In *International Conference on Electronic Commerce and Web Technologies*, pp. 89–100. Springer.

[3] Bi, X., A. Qu, and X. Shen (2018). Multilayer tensor factorization with applications to recommender systems. *The Annals of Statistics 46*(6B), 3308–3333.

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

[5] Chan, S. and E. Airoldi (2014). A consistent histogram estimator for exchangeable graph models. In *International Conference on Machine Learning*, pp. 208–216.

[6] Chatterjee, S. (2015). Matrix estimation by universal singular value thresholding. *The Annals of Statistics 43*(1), 177–214.

[7] Chen, P., C. Gao, and A. Y. Zhang (2021). Optimal full ranking from pairwise comparisons. *arXiv preprint arXiv:2101.08421*.

[8] Ding, J., Z. Ma, Y. Wu, and J. Xu (2021). Efficient random graph matching via degree profiles. *Probability Theory and Related Fields 179*(1), 29–115.

[9] Flammarion, N., C. Mao, and P. Rigollet (2019). Optimal rates of statistical seriation. *Bernoulli 25*(1), 623–653.

[10] Gao, C., Y. Lu, and H. H. Zhou (2015). Rate-optimal graphon estimation. *The Annals of Statistics 43*(6), 2624–2652.

[11] Györfi, L., M. Kohler, A. Krzyżak, and H. Walk (2002). *A distribution-free theory of nonparametric regression*, Volume 1. Springer.

[12] Han, R., Y. Luo, M. Wang, and A. R. Zhang (2020). Exact clustering in tensor block model: Statistical optimality and computational limit. *arXiv preprint arXiv:2012.09996*.

[13] Hore, V., A. Viñuela, A. Buil, J. Knight, M. I. McCarthy, K. Small, and J. Marchini (2016). Tensor decomposition for multiple-tissue gene expression experiments. *Nature genetics 48*(9), 1094.

[14] Hütter, J.-C., C. Mao, P. Rigollet, and E. Robeva (2020). Estimation of monge matrices. *Bernoulli 26*(4), 3051–3080.

[15] Jeremy, G. A trying first half of 2020 included spike in shootings and homicides in chicago. *Chicago Tribune*.

[16] Klopp, O., A. B. Tsybakov, and N. Verzelen (2017). Oracle inequalities for network models and sparse graphon estimation. *The Annals of Statistics 45*(1), 316–354.

[17] Kolda, T. G. and B. W. Bader (2009). Tensor decompositions and applications. *SIAM Review 51*(3), 455–500.

[18] Li, Y., D. Shah, D. Song, and C. L. Yu (2019). Nearest neighbors for matrix estimation interpreted as blind regression for latent variable model. *IEEE Transactions on Information Theory 66*(3), 1760–1784.

[19] Livi, L. and A. Rizzi (2013). The graph matching problem. *Pattern Analysis and Applications 16*(3), 253–283.

[20] Lovász, L. (2012). *Large networks and graph limits*, Volume 60. American Mathematical Soc.

[21] Ortelli, F. and S. van de Geer (2021). Prediction bounds for higher order total variation regularized least squares. *Annals of Statistics, in press*.

[22] Pananjady, A. and R. J. Samworth (2021). Isotonic regression with unknown permutations: Statistics, computation, and adaptation. *The Annals of Statistics, in press*.

[23] Phillippe Rigollet, J.-C. H. (2015). High dimensional statistics. *Lecture notes for course 18S997*.

[24] Shah, N., S. Balakrishnan, and M. Wainwright (2019). Low permutation-rank matrices: Structural properties and noisy completion. *Journal of Machine Learning Research 20*, 1–43.

[25] Stone, C. J. (1982). Optimal global rates of convergence for nonparametric regression. *The Annals of Statistics 10*(4), 1040–1053.

[26] Sun, W. W., J. Lu, H. Liu, and G. Cheng (2017). Provable sparse tensor decomposition. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 79*(3), 899–916.

[27] Tibshirani, R. J. (2014). Adaptive piecewise polynomial estimation via trend filtering. *The Annals of Statistics 42*(1), 285–323.

[28] Tsybakov, A. B. (2009). *Introduction to nonparametric estimation*. Springer Science & Business Media.

[29] Wang, M. and L. Li (2020). Learning from binary multiway data: Probabilistic tensor decomposition and its statistical optimality. *Journal of Machine Learning Research 21*(154), 1–38.

[30] Wang, M. and Y. Zeng (2019). Multiway clustering via tensor block models. In *Advances in Neural Information Processing Systems*, pp. 713–723.

[31] Wasserman, L. (2006). *All of nonparametric statistics*. Springer Science & Business Media.

[32] Xu, J. (2018). Rates of convergence of spectral methods for graphon estimation. In *International Conference on Machine Learning*, pp. 5433–5442.

[33] Young, J.-G., G. St-Onge, P. Desrosiers, and L. J. Dubé (2018). Universality of the stochastic block model. *Physical Review E 98*(3), 032309.

[34] Zhang, A. and D. Xia (2018). Tensor SVD: Statistical and computational limits. *IEEE Transactions on Information Theory 64*(11), 7311 – 7338.

[35] Zhao, Y. (2015). Hypergraph limits: a regularity approach. *Random Structures & Algorithms 47*(2), 205–226.

[36] Zhou, H., L. Li, and H. Zhu (2013). Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association 108*(502), 540–552.