A Bayesian Covariance Based Clustering for High Dimensional Tensors

Rene Gutierrez Marquez, Aaron Wolfe Scheffler and Rajarshi Guhaniyogi

January 2021

Abstract

Clustering of high dimensional tensors with limited sample size has become prevalent in a variety of application areas. Existing Bayesian model based clustering of tensors yields less accurate clusters when the tensor dimensions are sufficiently large, sample size is low and clusters of tensors mainly reveal difference in their variability. This article develops a novel clustering technique for high dimensional tensors with limited sample size when the clusters show difference in their covariances, rather than in their means. The proposed approach constructs several matrices from a tensor to adequately estimate its variability along different modes and implements a model-based approximate Bayesian clustering algorithm with the matrices thus constructed, in place with the original tensor data. Although some information in the data is discarded, we gain substantial computational efficiency and accuracy in clustering. Simulation study assesses the proposed approach along with its competitors in terms of estimating the number of clusters, identification of the modal cluster membership along with the probability of mis-classification in clustering (a measure of uncertainty in clustering). We further establish the effectiveness of our algorithm through applications to a real data set from a biomedical context.

Keywords: Bayesian statistics; Brain genome expression; Clustering; Chinese restaurant process; Tensor normal distribution.

1 Introduction

In recent times, multidimensional arrays or tensors, which are higher order extensions of two dimensional matrices, are being encountered in datasets emerging from different disciplines including datasets from different brain imaging modalities, multi-omics studies, chemometrics and psychometrics. Statistical analysis of tensor data presents several challenges over and above multivariate vector-based methods. First of all, due to the high dimensional nature of tensor data, inference from tensors often require a large parameter space. Also, extra care needs to be exercised to exploit structural information in a tensor object. To address such challenges for tensor data, a plethora of literature has emerged on tensor decomposition (Chi and Kolda, 2012; Dunson and Xing, 2009; Sun and Li, 2019a) and regressions with general and symmetric tensors (Zhou et al., 2013; Guhaniyogi et al., 2017; Lock, 2018; Guhaniyogi and Spencer, 2018; Guha and Guhaniyogi, 2020; Spencer et al., 2020). Most of these approaches employ low-rank and sparse approximations in the tensor structure to reduce the number of parameters considerably, and propose novel estimation tools to draw adequate inference.

This article focuses on clustering of high dimensional tensors into subgroups when tensors in different subgroups are barely distinguishable in terms of locations (e.g. mean), but exhibit difference in their correlation structures/variability. Examples of such datasets can be found in image analysis, financial, and biological processes. FLoss-based algorithmic approaches for clustering of vectors (Hartigan and Wong, 1979; Banerjee et al., 2004) can be extended to the clustering of tensors (Huang et al., 2008), offering a simple approach that is computationally efficient. However, loss-based approaches focuses on the aggregation and separation of a sample into groups depending on similarities in locations of data, and hence is not useful in applications of our interest. Moreover, there is no way to account for clustering uncertainty in these methods. In contrast with algorithmic clustering, model-based clustering exploits the entire data distribution for clustering, hence is relatively less affected by the fact that locations of the tensors are similar. For more background, see Fraley and Raftery (2002); Müller et al. (2015) for overviews of model-based clustering. In clustering the tensor observations under the model-based clustering framework, one simple

solution would be to vectorize the tensor object followed by unsupervised clustering of these vectors. Such an approach can make use of the wide literature on clustering high dimensional vector observations (Medvedovic and Sivaganesan, 2002; Zhong and Ghosh, 2003; Raftery and Dean, 2006; Fröhwirth-Schnatter and Kaufmann, 2008; Pan and Shen, 2007; Wang and Zhu, 2008; Lee et al., 2013; Oh and Raftery, 2007). However, vectorization ignores the crucial neighborhood structure of tensor objects. Additionally, vectorization of a K-mode tensor of dimensions $p_1 \times \cdots \times p_K$ results in a $\prod_{k=1}^{K} p_k$ dimensional vector. Model-based clustering of such long vectors often results in inaccurate clustering with each subject assigned to its own singleton cluster (Celeux et al., 2019). Frühwirth-Schnatter (2006) proposes a specific prior elicitation criterion to overcome this issue for moderate dimensions. However, calibration of hyper-parameters may appear to be difficult for large dimensions that we focus in this article.

The model-based clustering typically assumes each observation to follow a finite/infinite mixture of distributions. In particular, Gaussian mixture model (GMM) is widely deployed for clustering of scalar- or vector-valued observations. In the context of clustering higher order tensors, an ordinary GMM can be extended to mixture of tensor normal distributions, referred to as tensor normal mixtures (TNM) hereon. The tensor normal distribution expresses the covariance structure of a tensor in terms of covariance structure in every mode of the tensor, i.e., the covariance of a K-mode tensor is expressed with covariance matrices of the order $p_1 \times p_1$,..., $p_K \times p_K$. This eliminates the need to model an unstructured covariance matrix of the order of $p \times p$, where $p = \prod_{k=1}^{K} p_k$ for a tensor observation, and instead expresses covariance structure with only $\sum_{k=1}^{K} p_k(p_k + 1)/2$ elements, leading to a substantial reduction in the number of parameters required for covariance modeling. Further, the tensor covariance structure can be suitably exploited to simultaneously cluster observations and estimate parameters using either expectation maximization (EM) algorithm, its variants (in the frequentist framework) or Gibbs sampling (in the Bayesian framework) (Viroli, 2011; Anderlucci et al., 2015; Gao et al., 2020; Mai et al., 2021a). However, a standard Gibbs sampling algorithm applied to the clustering of high-dimensional tensors presents the arduous task of sampling the covariance structure in each mode of the high-dimensional tensors at every iteration. Besides being computationally inefficient, this often results in inaccurate

estimation of true clusters.

This article tackles the problem from a different point of view. In particular, we focus on a set of observations from multiple populations all of which follow tensor normal distributions with the same mean but different covariances. Rather than directly clustering these observations using model-based clustering that presents challenges described earlier, we adopt a two-step approach. As a first step, we construct a set of matrices, referred to as the "transformed features," from each tensor. These transformed features are designed to estimate variability of a tensor along different modes. We show that when $p_1, ..., p_K$ are large, the transformed features provide abundant information on the mode-specific covariance matrices of a TN distribution, thereby turning curse of dimensionality into a blessing. In the second step, a Bayesian mixture model on transformed features is employed to cluster observations. The proposal makes use of difference between clusters in their covariance structure, and at the same time avoids drawing Markov Chain Monte Carlo (MCMC) samples for high dimensional covariance parameters from tensor normal distributions, resulting in straightforward computation even with large tensor dimensions. Moreover, we provide clustering uncertainty in terms of mis-classification probabilities.

In the similar spirit as ours, Ieva et al. (2016) developed a novel covariance-based clustering algorithm exploiting the distance between covariances for multi-variate and functional data. Their approach is based on the crucial assumption that there are two groups/clusters, while we do not need to specify the number of clusters. Hallac et al. (2018) proposed a method for multivariate time-series data to segment and cluster. While this approach can be used for the tensor clustering, they assume a Toeplitz structure for the covariance matrix. In contrast, our proposed approach is applicable to the general structure of the tensor covariance matrix induced by the tensor normal distribution.

Rather than clustering tensors using the mixture of tensor normal distributions, there is a literature regarding K-means clustering on low-rank approximation of tensors. For example, a class of methods assume tensor decomposition of the mean of the tensor normal distribution, followed by minimization of the total squared Euclidean distance of each observation mean to its cluster centroid (Sun and Li, 2019a). While the low-rank approximation is widely adopted in tensor data analysis, this approach typically work on identifying clusters through

centers of their distributions, and is thus less suitable for our purpose. Our goal is also very different from the literature on bi-clustering and co-clustering methods. Lee et al. (2010); Tan and Witten (2014) develop bi-clustering methods that simultaneously group features and observations into clusters. Extensions of the feature-sample bi-clustering for vector observations are known as the co-clustering or multiway clustering problems (Jegelka et al., 2009; Chi et al., 2020; Wang and Zeng, 2019), where each mode of the tensor is clustered into groups. Our problem is different from these works in that our sole goal is to cluster the observations.

Rest of the article evolves as follows. In section 2 we provide a brief introduction of model based clustering and describe our approach for clustering tensors with covariance estimators. Posterior computation from the model is described in Section 3. Empirical evaluations with simulation studies and a real data analysis are presented in Sections 4 and 5, respectively. Finally, we conclude in Section **??** with an eye towards the future work.

2 Covariance-Based Bayesian Tensor Clustering

This section begins with defining notations related to tensors. The Bayesian model-based clustering approach is then briefly discussed in its full generality in the context of tensor observations. We then describe the covariance-based two-step clustering approach in the context of high dimensional tensor observations.

2.1 Notations

We begin with a quick review of some tensor notations and operations which will be subsequently used. A more detailed review can be found in Kolda and Bader (2009).

Consider the K-way tensor (also known as K-mode or K-th order tensor) $\mathbf{T} \in \mathbb{R}^{p_1 \times \ldots \times p_K}$ with its $(i_1, ..., i_K)$-th element denoted by $T_{i_1,\ldots,i_K}$. When $K = 1$, the tensor reduces to a vector and when $K = 2$, the tensor is a matrix. The vec($\mathbf{T}$) operator applied to a tensor $\mathbf{T}$ stacks elements into a column vector of dimension $p = \prod_{k=1}^{K} p_k$ with $T_{i_1,\ldots,i_K}$ mapped to the j-th entry of $vec(\mathbf{T})$, for $j = 1 + \sum_{k=1}^{K}(i_k - 1)\prod_{k'=1}^{k-1} p_{k'}$.

A fiber is the higher order analogue of a matrix row and column, and is defined by fixing every index of the tensor but one. A k-mode fiber is a p_k-dimensional vector obtained by

fixing all other modes except the k-th mode. For example, a matrix column is a mode-1 fiber and a row is a mode-2 fiber. There are p/p_k such k-mode fibers for $\mathbf{T}$ each with dimension $p_k \times 1$. The k-mode matricization of a tensor transforms a tensor into a matrix $\mathbf{T}_{(k)} \in \mathbb{R}^{p_k \times \frac{p}{p_k}}$, where $T_{(i_1,...,i_K)}$ mapping to (i_k, j)-th element of the matrix, where $j = \sum_{k' \neq k}(i_{k'} - 1) \prod_{k'' < k', k'' \neq k} p_{k''}$. The k-mode product of a tensor $\mathbf{T} \in \mathbb{R}^{p_1 \times ... \times p_K}$ and a compatible matrix $\mathbf{A} \in \mathbb{R}^{J \times p_k}$, will result in a tensor $\mathbf{T} \times_k \mathbf{A} \in \mathbb{R}^{p_1 \times ... \times p_{k-1} \times J \times p_{k+1} \times ... p_K}$, where each element is the product of mode-k fiber of $\mathbf{T}$ multiplied by $\mathbf{A}$. Notice that this operation reduces to the usual matrix product for a 2-way tensor and to the inner product for a 1-way tensor. Finally, for a list of matrices $\mathbf{A}_1, \ldots, \mathbf{A}_K$ with compatible sizes $A_k \in \mathbb{R}^{J_k \times p_k}$ we define the product $\mathbf{T} \times [\mathbf{A}_1, \ldots, \mathbf{A}_K] = \mathbf{T} \times_1 \mathbf{A}_1 \times_2 \ldots \times_K \mathbf{A}_K \in \mathbb{R}^{J_1 \times ... \times J_K}$. Thus, when $\mathbf{A}_1, \ldots, \mathbf{A}_K$ are square matrices, the resulting tensor is of the same dimension as $\mathbf{T}$. In what follows, we will use $||\cdot||_F$ to denote the Frobenius norm of the tensor $\mathbf{T}$ given by $||\mathbf{T}||_F := \sqrt{\sum_{i_1,...,i_K} T^2_{i_1,...,i_K}}$.

2.2 Bayesian Model-based Tensor Clustering Approach

Let $\mathbf{T}_i$ be a tensor valued observation in $\mathcal{T}$, $\mathcal{T} \subseteq \mathbb{R}^{p_1 \times \cdots \times p_K}$, for $i = 1, ..., n$. Let $\mathcal{C} = \{\mathcal{C}_1, ..., \mathcal{C}_{n(\mathcal{C})}\}$ be a partition of n observations into $n(\mathcal{C})$ disjoint sets, i.e., $|\mathcal{C}| = n(\mathcal{C})$. Typical Bayesian models for clustering are based on posterior distributions of the form

$$\pi(\mathcal{C}|\mathbf{T}_1, .., \mathbf{T}_n) \propto \pi(\mathcal{C}) \prod_{h=1}^{n(\mathcal{C})} \left[\int \prod_{i \in \mathcal{C}_h} f(\mathbf{T}_i|\mathbf{\Theta}_h) \pi(\mathbf{\Theta}_h) d\mathbf{\Theta}_h \right] = \pi(\mathcal{C}) \prod_{h=1}^{n(\mathcal{C})} m(\{\mathbf{T}_i : i \in \mathcal{C}_h\}), \quad (1)$$

where $f(\mathbf{T}_i|\mathbf{\Theta}_h)$ denotes the likelihood for a tensor observation belonging to the h-th cluster with the cluster-specific model parameter $\mathbf{\Theta}_h$ and $\pi(\mathbf{\Theta}_h)$ corresponds to the prior distribution on the parameter $\mathbf{\Theta}_h$. The quantity $m(\{\mathbf{T}_i : i \in \mathcal{C}_h\}) = \int \prod_{i \in \mathcal{C}_h} f(\mathbf{T}_i|\mathbf{\Theta}_h) \pi(\mathbf{\Theta}_h) d\mathbf{\Theta}_h$ denotes the marginal distribution of tensors belonging to the h-th cluster which is typically not obtained in a closed form. Alternatively, the partition can be described through cluster labels for n observations given by $\mathbf{c} = (c_1, ..., c_n)'$, so that $c_i = h$, if and only if $i \in \mathcal{C}_h$, for $i = 1, ..., n$. Irrespective of the representation, our interest only lies in the induced partition $\mathcal{C}$ rather than the labels on the indicators $\mathbf{c} = (c_1, ..., c_n)'$.

A natural choice for the likelihood $f(\mathbf{T}_i|\mathbf{\Theta}_h)$ appears to be a tensor normal distribution,

denoted as $\text{TN}(\boldsymbol{M}_h, \boldsymbol{\Sigma}_{1,h}, ..., \boldsymbol{\Sigma}_{K,h})$, and is given by

$$f(\mathbf{T}_i|\boldsymbol{M}_h, \boldsymbol{\Sigma}_{1,h}, \ldots, \boldsymbol{\Sigma}_{K,h}) = (2\pi)^{-\frac{p}{2}} \left\{ \prod_{k=1}^{K} |\boldsymbol{\Sigma}_{k,h}|^{-\frac{p}{2p_k}} \right\} \exp\left(-\frac{1}{2} \left\| (\mathbf{T}_i - \boldsymbol{M}_h) \times [\boldsymbol{\Sigma}_{1,h}^{-\frac{1}{2}}, \ldots, \boldsymbol{\Sigma}_{K,h}^{-\frac{1}{2}}] \right\|_F^2 \right), \tag{2}$$

where $\boldsymbol{M}_h$ is the mean/center of the tensor normal distribution, and $\boldsymbol{\Sigma}_{k,h}$ is a $p_k \times p_k$ dimensional positive definite matrix, also referred to as the covariance matrix for the k-th mode. We consider a scenario where the observed tensors in the sample are barely distinguishable in terms of their means. Thus, we make the following crucial assumption:

Assumption A: *Different clusters of tensors only vary in terms of their covariance structure and not in their means. Thus, without loss of generality,* $\boldsymbol{M}_h = \mathbf{0}$ *for all* $h = 1, .., n(\mathcal{C})$.

According to the likelihood specification in (2) and Assumption A, $\boldsymbol{\Theta}_h$ corresponds to the collection of covariance matrices for all modes, i.e., $\boldsymbol{\Theta}_h = \{\boldsymbol{\Sigma}_{1,h}, \ldots, \boldsymbol{\Sigma}_{K,h}\}$.

Notably, the distributional form of $f(\mathbf{T}_i|\boldsymbol{\Theta}_h)$, as given in (2), does not yield a closed form integral for the marginal distribution in (1). The common practice is to begin with the distribution $(\mathbf{T}_i|\boldsymbol{\Theta}_h, c_i = h) \sim f(\mathbf{T}_i|\boldsymbol{\Theta}_h)$ and develop a Gibbs sampler to draw posterior samples of $\mathbf{c}$ along with $\boldsymbol{\Sigma}_{k,h}$'s, for all $k = 1, ..., K$ and $h = 1, ..., n(\mathcal{C})$. However, when $p_1, ..., p_K$ are large, Gibbs sampling of covariance matrices $\boldsymbol{\Sigma}_{k,h}$'s results in inferential inaccuracy related to clustering, as well as computational challenges, as demonstrated in our detailed empirical investigation in Section 4. Next section develops an approximate Bayesian clustering algorithm that offers remedies to both these challenges simultaneously.

2.3 A Covariance-Based Bayesian Tensor Clustering Approach

To avoid complications due to model based clustering of high-dimensional tensor observations, we propose a twos-step Bayesian clustering approach of tensors. In summary, our approach first extracts important features of high dimensional tensors to adequately estimate the covariance structure along different modes, followed by model-based clustering of these features. To elaborate on it, let $\mathcal{A}(\mathbf{T}_i)$ be the set of extracted features from tensor $\mathbf{T}_i$ which will be referred to as transformed features (TF) hereon. The transformed features are carefully chosen to estimate variability of the tensor normal distribution in each mode. Section

2.4 details out a specific choice of such transformed features. While the exact distribution of $\mathcal{A}(\mathbf{T}_i)$ is determined by the tensor normal specification given in (2), we focus on a reasonable approximation of the distribution for $\mathcal{A}(\mathbf{T}_i)$ in our goal to cluster these transformed features. Let $\tilde{f}(\mathcal{A}(\mathbf{T}_i)|\tilde{\mathbf{\Theta}}_h, \tilde{\mathbf{\Theta}}_a)$ be the approximated distribution of $\mathcal{A}(\mathbf{T}_i)$ in the h-th cluster, with $\tilde{\mathbf{\Theta}}_h$ as its h-th cluster-specific parameter and $\tilde{\mathbf{\Theta}}_a$ an auxiliary lower dimensional parameter common across all clusters. Let $\tilde{\pi}_h(\tilde{\mathbf{\Theta}}_h)$ and $\tilde{\pi}_a(\tilde{\mathbf{\Theta}}_a)$ denote the prior distribution of $\tilde{\mathbf{\Theta}}_h$ and $\tilde{\mathbf{\Theta}}_a$, respectively, for $h = 1, ..., H$. We choose $\tilde{f}(\cdot)$ and $\tilde{\pi}_h(\cdot)$ to ensure closed form marginal distribution of $\tilde{m}(\{\mathcal{A}(\mathbf{T}_i) : i \in \mathcal{C}_h\}|\tilde{\mathbf{\Theta}}_a) = \int \prod_{i\in\mathcal{C}_h} \tilde{f}(\mathcal{A}(\mathbf{T}_i)|\tilde{\mathbf{\Theta}}_h, \tilde{\mathbf{\Theta}}_a)\tilde{\pi}_h(\tilde{\mathbf{\Theta}}_h)d\tilde{\mathbf{\Theta}}_h$.

With closed form marginals for TFs in each cluster, the posterior distribution of clusters and the auxiliary parameters is given by,

$$\pi(\mathcal{C}, \tilde{\mathbf{\Theta}}_a \mid \mathcal{A}(\mathbf{T}_1), ..., \mathcal{A}(\mathbf{T}_n)) = \pi(\mathcal{C})\tilde{\pi}_a(\tilde{\mathbf{\Theta}}_a) \prod_{h=1}^{n(\mathcal{C})} \tilde{m}(\{\mathcal{A}(\mathbf{T}_i) : i \in \mathcal{C}_h\}|\tilde{\mathbf{\Theta}}_a), \tag{3}$$

where $\pi(\mathcal{C})$ denotes the prior on partitions. In the absence of real prior information about the items, we will assign positive prior probability to every possible partition. In the interests of computational convenience, we might be attracted to prior models on partitions for which posterior simulation methods are fully developed. While the nonzero prior on partitions can be induced by Dirichlet processes (Ferguson, 1973; Antoniak, 1974; Gopalan and Berry, 1998), an explicit prior on partitions can also be derived from an infinite or a finite mixture model representation of the distribution of $\mathcal{A}(\mathbf{T}_i)$ after integrating out the weights of the mixing components. With the posterior distribution of partitions given in (3), the computation proceeds through a Chinese restaurant sampler described below (Lau and Green, 2007).

1. *Initialize:* Choose an initial partition $\mathcal{C}^{(0)}$. Common options are either to set singleton clusters or to put all observations in the same cluster.

2. *Obtain s-th iterate of $\mathcal{C}$:* To obtain s-th iterate of the partition $\mathcal{C}^{(s)}$ do:

   (a) *Initialize the Partition:* Set $\mathcal{C} = \mathcal{C}^{(s-1)}$, and let $\mathcal{C} = \{\mathcal{C}_1, \ldots, \mathcal{C}_{n(\mathcal{C})}\}$.

   (b) *Loop through every observation:*

i. *Remove observation $\mathcal{A}(\mathbf{T}_i)$ from the partition:* Remove i-the observation from the partition $\mathcal{C}$ to obtain a new partition

$$\mathcal{C}_{-i} = \{\mathcal{C}_{1,-i}, ..., \mathcal{C}_{n(\mathcal{C}_{-i}),-i}\}.$$

ii. *Assign observation i:* Either assign the i-th observation to a new cluster, that is update $\mathcal{C}$ to $\mathcal{C} = \{\mathcal{C}_{1,-i}, ..., \mathcal{C}_{n(\mathcal{C}_{-i}),-i}, \{i\}\}$ with probability proportional to:

$$\tilde{m}(\mathcal{A}(\mathbf{T}_i)|\tilde{\mathbf{\Theta}}_a) \times \frac{\pi(\{\mathcal{C}_{1,-i}, ..., \mathcal{C}_{n(\mathcal{C}_{-i}),-i}, \{i\}\})}{\pi(\{\mathcal{C}_{1,-i}, ..., \mathcal{C}_{n(\mathcal{C}_{-i}),-i}\})}, \tag{4}$$

or, assign the i-th observation to the existing j-th cluster $\mathcal{C}_{j,-i}$, that is update $\mathcal{C}$ to
$\mathcal{C} = \{\mathcal{C}_{1,-i}, ..., \mathcal{C}_{j,-i} \cup \{i\}, \ldots, \mathcal{C}_{n(\mathcal{C}_{-i}),-i}\}$ with probability proportional to:

$$\frac{\tilde{m}(\{\mathcal{A}(\mathbf{T}_s) : s \in \{\{i\} \cup \mathcal{C}_{j,-i}\}\}|\tilde{\mathbf{\Theta}}_a)}{\tilde{m}(\{\mathcal{A}(\mathbf{T}_s) : s \in \mathcal{C}_{j,-i}\}|\tilde{\mathbf{\Theta}}_a)} \times \frac{\pi(\{\mathcal{C}_{1,-i}, ..., \mathcal{C}_{j,-i} \cup \{i\}, \ldots, \mathcal{C}_{n(\mathcal{C}_{-i}),-i}\})}{\pi(\{\mathcal{C}_{1,-i}, ..., \mathcal{C}_{n(\mathcal{C}_{-i}),-i}\})} \tag{5}$$

(c) *Set the partition $\mathcal{C}^{(s)}$:* After updating $\mathcal{C}$, going through every observation, set $\mathcal{C}^{(s)} = \mathcal{C}$.

3. *Sample the s-th iterate of $\tilde{\mathbf{\Theta}}_a$:* Draw s-th iterate of $\tilde{\mathbf{\Theta}}_a$ from its full conditional distribution derived from (3).

Notably, steps (a)-(c) involve marginal distribution of TFs which are available in closed form by our assumption. In fact, the algorithm bypasses updating high dimensional parameters at any step, which leads to rapid mixing of the Markov Chain. Since the algorithm uses transformed features $\mathcal{A}(\mathbf{T}_i)$ of the tensor $\mathbf{T}_i$, the clustering accuracy is naturally dependent on the choice of these features. Next section describes specific choice of TFs which leads to desirable clustering performance for tensors, as discussed in the simulation studies.

2.4 Transformed Features and Their Distributions

This section discusses the specific choice of transformed features $\mathcal{A}(\mathbf{T})$ and the approximate distribution $\tilde{f}(\mathcal{A}(\mathbf{T})|\tilde{\mathbf{\Theta}}_h, \tilde{\mathbf{\Theta}}_a)$ of the transformed features used in this article. For clustering of high dimensional tensors, we propose to work with the collection of transformed features given by $\mathcal{A}(\mathbf{T}_i) = \{\frac{p_k}{p}\mathbf{T}_{i,(k)}\mathbf{T}'_{i,(k)} : k = 1, ..., K\}$, where $\mathbf{T}_{i,(k)}$ is the k-th mode matrix of the tensor $\mathbf{T}_i$. Therefore, given a k-way tensor observation $\mathbf{T}_i$ of dimension $p = \prod_{i=1}^{K} p_i$, we extract a collection of K matrices of sizes $p_1 \times p_1, \ldots, p_K \times p_k$, which will suitably capture the covariance structure of the observed tensor, as described by the lemma below.

Lemma 2.1 *Let* $\mathbf{T}_i \sim TN(\mathbf{0}, \mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_K)$ *and* $\mathcal{A}(\mathbf{T}_i)^{(k)} = \frac{p_k}{p}\mathbf{T}_{i,(k)}\mathbf{T}'_{i,(k)}$. *Assume that for all* $k = 1, ..., K$, *(i)* $\frac{p_k}{p} \to 0$ *(ii)* $\frac{p_k}{p} tr(\otimes_{k' \neq k}\mathbf{\Sigma}_{k'}) \to w_k$ *and (iii)* $\frac{p_k^2}{p^2}\sum_{l,r}\{\otimes_{k' \neq k}\mathbf{\Sigma}_{k'}\}_{l,r} \to 0$, *for all* $l, r = 1, ..., p/p_k$, *where* $\{\otimes_{k' \neq k}\mathbf{\Sigma}_{k'}\}_{l,r}$ *denotes the* (l, r)*th entry of the matrix* $\otimes_{k' \neq k}\mathbf{\Sigma}_{k'}$. *(i)-(iii) together imply that* $\{\mathcal{A}(\mathbf{T}_i)^{(k)}\}_{l,r} \to \{\mathbf{\Sigma}_k\}_{l,r} w_k$, *where* w_k *is a constant.*

The proof of Lemma 2.1 is provided in the supplementary material. While high-dimensional tensors pose challenges in the ordinary clustering approaches due to the need to estimate high dimensional covariance matrices for different modes, higher tensor dimensions appear to be "blessings" for our approximate tensor clustering approach, as revealed in Lemma 2.1. In fact, the result implies that under regularity conditions, as the tensor dimensions grow, the transformed features converge to mode-specific covariance matrices upto a scale factor, recovering their shapes and orientations.

Some discussions on assumptions (i)-(iii) is warranted. Assumption (i) is a mild one only guaranteeing growth of tensor along every dimension. Assumptions (ii) and (iii) restrict the growth of the elements in the covariance matrices of the data generating tensor normal distribution. In particular, when $\mathbf{\Sigma}_k$ is an identity matrix of dimension $p_k \times p_k$, (ii) and (iii) are trivially satisfied with $w_k = 1$ for all $k = 1, ..., K$. Broadly, the conditions (ii) and (iii) assumes sparsity in the mode-specific covariance matricesnwhich turn out to be a crucial in dictating the clustering performance of the approach.

2.4.1 The TF Distribution and Prior On Parameters

To cluster tensors with the transformed features introduced in the previous section, we employ cluster-specific normal means model on the upper triangular entries of $\mathcal{A}(\mathbf{T}_i)^{(k)}$ in all clusters and for all modes $k = 1, ..., K$. More specifically, the (l, r)-th entry of $\mathcal{A}(\mathbf{T}_i)^{(k)}$ is modeled as

$$\{\mathcal{A}(\mathbf{T}_i)^{(k)}\}_{l,r} \overset{ind.}{\sim} N(\theta^{(k)}_{l,r,h}, \sigma^2), \text{ for } i \in \mathcal{C}_h, \ \theta^{(k)}_{l,r,h} \sim N(\theta_0, \sigma^2/\phi), \ l < r. \tag{6}$$

(6) appears to be an approximation to the actual distribution of TFs under the tensor normal specification of $\mathbf{T}_i$, when tensor dimensions are large. In fact, when $i \in \mathcal{C}_h$ and $\mathbf{T}_i \sim TN(\mathbf{0}, \mathbf{\Sigma}_{1,h}, .., \mathbf{\Sigma}_{K,h})$, $\{\mathcal{A}(\mathbf{T})^{(k)}\}_{l,r}$ is approximately distributed as normal by central limit theorem as $p_k/p \to 0$.

The specification of (6) leads to a close form marginal distribution of $\mathcal{A}(\mathbf{T}_i)$ in each cluster conditional on the auxiliary parameters $\tilde{\mathbf{\Theta}}_a = (\sigma^2, \phi)'$ by integrating out cluster specific parameters $\tilde{\mathbf{\Theta}}_h = (\theta^{(k)}_{l,r,h} : l < r)'$. More specifically,

$$\begin{aligned}\tilde{m}\left(\{\{\mathcal{A}(\mathbf{T}_i)^{(k)}\}_{l,r} : i \in \mathcal{C}_h\}|\phi, \sigma^2\right) &= \left(2\pi\sigma^2\right)^{\frac{-n_h}{2}} \left[\frac{\phi}{n_h + \phi}\right]^{\frac{1}{2}} \\ &\exp\left\{-\frac{1}{2\sigma^2}\left(\left[\sum_{i\in\mathcal{C}_h}\left(\{\mathcal{A}(\mathbf{T}_i)^{(k)}\}_{l,r} - \{\bar{\mathcal{A}}(\mathbf{T})^{(k)}_{\mathcal{C}_h}\}_{l,r}\right)^2\right] + \phi\left(\{\bar{\mathcal{A}(\mathbf{T})}^{(k)}_{\mathcal{C}_h}\}_{l,r} - \theta_0\right)^2\right)\right\},\end{aligned} \tag{7}$$

where $n_h = |\mathcal{C}_h|$ is the number of samples belonging to the h-th cluster $\mathcal{C}_h$ and $\{\bar{\mathcal{A}}(\mathbf{T})^{(k)}_{\mathcal{C}_h}\}_{l,r} = \frac{1}{n_h+\phi}\left(\sum_{i\in\mathcal{C}_h}\{\mathcal{A}(\mathbf{T}_i)^{(k)}\}_{l,r} + \phi\theta_0\right)$. The marginal distribution of $\mathcal{A}(\mathbf{T}_1), ..., \mathcal{A}(\mathbf{T}_n)$ conditional on the auxiliary parameters σ^2 and ϕ is of the form

$$\tilde{m}\left(\mathcal{A}(\mathbf{T}_1), ..., \mathcal{A}(\mathbf{T}_n)|\phi, \sigma^2\right) = \prod_{h=1}^{n(\mathcal{C})}\prod_{i\in\mathcal{C}_h}\prod_{k=1}^{K}\prod_{1\le l<r\le p_k}\tilde{m}\left(\{\{\mathcal{A}(\mathbf{T}_i)^{(k)}\}_{l,r} : i \in \mathcal{C}_h\}|\phi, \sigma^2\right), \tag{8}$$

where the form of $\tilde{m}\left(\{\{\mathcal{A}(\mathbf{T}_i)^{(k)}\}_{l,r} : i \in \mathcal{C}_h\}|\phi, \sigma^2\right)$ is obtained from (7).

While Section 2.3 outlines a number of possibilities for the choice of the prior distribution on partitions, we have adopted the prior on the partitions induced from the Dirichlet Process.

Following Lau and Green (2007), the prior distribution on the partition $\mathcal{C}$ under such a specification assumes the form,

$$\pi(\mathcal{C}|\phi) = \phi^{n(\mathcal{C})+1} \frac{\Gamma(\phi)}{\Gamma(n+\phi)} \prod_{h=1}^{n(\mathcal{C})} \Gamma(n_h), \tag{9}$$

with the prior being dependent on the auxiliary parameter ϕ. Following the Chinese Restaurant analogy, (9) implies that the probability of assigning a new customer to a new table is proportional to ϕ a priori. The prior specification is completed by setting an inverse-gamma prior on σ^2, $\sigma^2 \sim IG(a_\sigma, b_\sigma)$ and a discrete uniform prior on ϕ taking values $\phi_1, ..., \phi_F$ each with probability $1/F$.

2.5 Point Estimation and Uncertainty Quantification in Clustering

While we will explore the posterior distribution of partitions through MCMC-based sampling algorithms (see Section 3 for details of posterior computation), it is worth understanding the point estimate of partitions induced by our approach. Although several alternatives exist (e.g., Medvedovic et al. (2004); Lau and Green (2007); Fritsch et al. (2009)), maximum a posteriori (MAP) estimation provides a particularly natural and simple choice. Unfortunately, the maximum a posteriori clusters are not available in closed form from our approach; thus we study some profile properties of partitions by fixing the auxiliary parameters σ^2 and ϕ. In particular, from (8), the MAP estimate of clustering is obtained by minimizing the following objective function with respect to clusters $\mathcal{C}_1, ..., \mathcal{C}_{n(\mathcal{C})}$ and their centers.

$$\sum_{h=1}^{n(\mathcal{C})} \sum_{k=1}^{K} \sum_{1 \le l < r \le p_k} \left\{ \sum_{i \in \mathcal{C}_h} ||\{\mathcal{A}(\mathbf{T}_i)^{(k)}\}_{l,r} - \{\bar{\mathcal{A}}(\mathbf{T})^{(k)}_{\mathcal{C}_h}\}_{l,r}||^2 + \phi \left(\theta_0 - \{\bar{\mathcal{A}}(\mathbf{T})^{(k)}_{\mathcal{C}_h}\}_{l,r}\right)^2 \right\} \tag{10}$$

With little algebra, equation (10) can be rewritten as

$$\sum_{h=1}^{n(\mathcal{C})} \sum_{k=1}^{K} \sum_{1 \le l < r \le p_k} \left\{ \sum_{i \in \mathcal{C}_h} ||\{\mathcal{A}(\mathbf{T}_i)^{(k)}\}_{l,r} - \{\bar{\mathcal{A}}(\mathbf{T})^{(k)}_{\mathcal{C}_h,0}\}_{l,r}||^2 + \frac{n_h \phi}{n_h + \phi} \left(\theta_0 - \{\bar{\mathcal{A}}(\mathbf{T})^{(k)}_{\mathcal{C}_h,0}\}_{l,r}\right)^2 \right\}, \tag{11}$$

where $\{\bar{\mathcal{A}}(\mathbf{T})^{(k)}_{\mathcal{C}_h,0}\}_{l,r} = \frac{1}{|\mathcal{C}_h|} \sum_{i \in \mathcal{C}_h} \{\mathcal{A}(\mathbf{T}_i)^{(k)}\}_{l,r}$. Notably, the objective function in (11) bears close connection with the objective function of regularized k-means clustering for high dimensional objects (Sun et al., 2012). Since the upper triangular vectors of $\mathcal{A}(\mathbf{T}_i)$ are high-dimensional, regularized k-means clustering is more suitable for cluster analysis than the ordinary k-means clustering. In fact, in an ordinary k-means clustering, the observations from the same cluster tend to lie symmetrically at the vertices of a regular simplex, and the distance between observations from different clusters is determined by the cluster difference relative to the data dimension. Consequently, if the cluster difference is relatively small compared with the diverging data dimension, the ordinary k-means clustering based on the Euclidean distance will operate in a degenerate fashion, assigning all the observations to the same cluster. In contrast, a regularized k-means clustering shrinks high dimensional observations to a lower-dimensional subspace while simultaneously performing cluster analysis, which is more suitable in our context.

One of the advantages of probabilistic model-based clustering is that it offers uncertainty quantification along with point estimate of clusters. Recall that the partitioning set $\mathcal{C}$ can be equivalently expressed in terms of cluster membership indices $\mathbf{c} = (c_1, ..., c_n)'$ for the data points, where each $c_i = h \Leftrightarrow i \in \mathcal{C}_h$. In principle, the uncertainty of clustering can be expressed through posterior probabilities $P(c_i = h|\text{Data})$, but these are affected by the label-switching phenomenon (Stephens, 2000). For this reason, one typically focuses on the co-clustering matrix $\mathbf{G}$ (Fritsch et al., 2009), whose entries $G_{i,i'}$ are such that $G_{i,i'} = P(c_i = c_{i'}|\text{Data})$, for $i, i' \in \{1, ..., n\}$. The $\mathbf{G}$ matrix can be used to identify which pair of units are more certain/uncertain to belong to the same cluster.

3 Posterior Computation

With likelihood and prior distributions specified as in Section 2.4.1, the full posterior distribution of partitions and auxiliary variables is given by,

$$p(\mathcal{C}, \phi, \sigma^2 | \mathcal{A}(\mathbf{T}_1), ..., \mathcal{A}(\mathbf{T}_n)) \propto \tilde{m}\left(\mathcal{A}(\mathbf{T}_1), ..., \mathcal{A}(\mathbf{T}_n) | \phi, \sigma^2\right) \times \phi^{n(\mathcal{C})+1} \frac{\Gamma(\phi)}{\Gamma(n+\phi)} \prod_{h=1}^{n(\mathcal{C})} \Gamma(n_h)$$
$$\times \frac{\beta_\sigma^{\alpha_\sigma}}{\Gamma(\alpha_\sigma)} (\sigma^2)^{-\alpha_\sigma - 1} \exp\left(-\frac{\beta_\sigma}{\sigma^2}\right).$$

The posterior computation proceeds following the general algorithm described in Section 2.3 with simplifications due to the prior structure. Specifically, the probability of assigning the i-th observation to a new cluster, described in (4), reduces to

$$\tilde{m}(\mathcal{A}(\mathbf{T}_i) | \phi, \sigma^2) \times \phi.$$

On the other hand, the probability of being assigned to the existing j-th cluster $\mathcal{C}_{j,-i}$, described in (5), takes the form

$$\frac{\tilde{m}\left(\{\mathcal{A}(\mathbf{T}_s) : s \in \{i\} \cup \mathcal{C}_{j,-i}\} | \phi, \sigma^2\right)}{\tilde{m}\left(\{\mathcal{A}(\mathbf{T}_s) : s \in \mathcal{C}_{j,-i}\} | \phi, \sigma^2\right)} \times |\mathcal{C}_{j,-i}|.$$

Thus Chinese restaurant process assigns an observation into an existing cluster or to a new cluster depending on the size of the existing clusters, parameter ϕ and similarity of the customers (observations) already in a cluster with the new observation.

Finally, the full conditional distribution to sample σ^2 in step 3 of the algorithm is given by $\text{IG}(a_{\sigma|-}, b_{\sigma|-})$ distribution with the values of $a_{\sigma|-}$ and $b_{\sigma|-}$ are given by

$$a_{\sigma|-} = a_\sigma + \frac{n \sum_{k=1}^{K} p_k(p_k - 1)}{2}$$
$$b_{\sigma|-} = b_\sigma + \frac{\sum_{h=1}^{n(\mathcal{C})} \sum_{k=1}^{K} \sum_{1 \le l < r \le p_k} \left[\sum_{i \in \mathcal{C}_h} \left(\{\mathcal{A}(\mathbf{T}_i)^{(k)}\}_{l,r} - \{\bar{\mathcal{A}}(\mathbf{T})_{\mathcal{C}_h}^{(k)}\}_{l,r} \right)^2 + \phi \left(\{\bar{\mathcal{A}}(\mathbf{T})_{\mathcal{C}_h}^{(k)}\}_{l,r} - \theta_0 \right)^2 \right]}{2}.$$

ϕ is sampled in each iteration from a discrete uniform distribution taking values ϕ_f with probability proportional to $\tilde{m}\left(\mathcal{A}(\mathbf{T}_1), ..., \mathcal{A}(\mathbf{T}_n)|\phi_f, \sigma^2\right) \times \phi_f^{n(\mathcal{C})+1} \frac{\Gamma(\phi_f)}{\Gamma(n+\phi_f)}$, for $f = 1, ..., F$. We fix $F = 20$ throughout our empirical investigation.

4 Numerical Illustration

This section studies the clustering performance of our proposed Bayesian Tensor Clustering (BTC) approach vis-a-vis its competitors. To study all competitors under various data generation schemes, we simulate $n = 100$ tensors $\mathbf{T}_1$,...,$\mathbf{T}_n$ from a finite mixture of tensor normal models with H mixing components given by

$$\mathbf{T}_i \sim \sum_{h=1}^{H} \pi_h TN(\mathbf{0}, \mathbf{\Sigma}_{1,h}, ..., \mathbf{\Sigma}_{K,h}), \ \sum_{h=1}^{H} \pi_h = 1. \tag{12}$$

The data generation scheme ensures that the tensors in different cluster differ only in their variability. Further, each simulated tensor is assumed to have $K = 3$ modes of dimensions $p_1 = 10$, $p_2 = 20$ and $p_3 = 30$. While our approach is scalable for a much bigger tensor size, we kept the tensor dimensions moderate in simulations to aid its comparison with the full Bayesian model-based clustering approach, discussed later. The probability of inclusion in every mixture component is taken to be identical $\pi_h = 1/H$, resulting in clusters of similar size. The precision matrices $\mathbf{\Sigma}_{k,h}^{-1}$ for the covariance structure are generated as sparse matrices to introduce complex conditional independence structure between the tensor cells following the popular literature on graphical models (Rothman et al., 2008; Liu and Martin, 2019; Cai et al., 2011). More specifically, each sparse matrix of dimension $p_k \times p_k$, $k = 1, ..., K$, is generated following the steps described below.

1. A symmetric edge matrix $\mathbf{E}$ is generated. Where each of diagonal entry is equal to 1 with probability α and 0 otherwise. And all the diagonal elements are equal to 0.

2. A matrix $\mathbf{D} = \mathbf{E}/2 + \delta\mathbf{I}$ where $\mathbf{I}$ is the identity and δ is chosen so that $\mathbf{D}$ has a condition number of p_k. Note that α determines the sparsity level.

3. The final matrix is obtained sampling from a G-Wishart distribution with degrees of freedom equal to $p_k + 3$ and scale matrix equal to $\mathbf{D}$.

In generating the true covariance matrices for different modes, the parameter α is used to control sparsity of the covariance matrices. We consider seven simulation cases by varying the number of clusters H and the sparsity of random precision matrices α, given by,

(a) **Case 1:** $H = 3, \alpha = 0.1$, (b) **Case 2:** $H = 4, \alpha = 0.1$,

(c) **Case 3:** $H = 3, \alpha = 0.2$, (d) **Case 4:** $H = 4, \alpha = 0.2$,

(e) **Case 5:** $H = 3, \alpha = 0.3$, (f) **Case 6:** $H = 4, \alpha = 0.3$,

(g) **Case 7:** $H = 4, \alpha = 0.4$.

The simulation results will develop understanding of how the interplay between number of clusters and the sparsity in the covariance matrices affects performance of the competitors.

4.1 Competitors and Metrics of Evaluation

As a competitor to our approach, we employ a few popular frequentist tensor clustering approaches; a static version of the Dynamic Tensor Clustering algorithm (DTC) (Sun and Li, 2019b) and Doubly-Enhanced EM algorithm (DEEM) proposed for tensor mixture models (Mai et al., 2021b). While our Bayesian approach allows simultaneous model-based determination of cluster number and composition of each cluster, both of these frequentist clustering techniques fix the number of clusters before implementing the clustering. In the simulation studies, we implement both DTC and DEEM by fixing the number of clusters at the truth. Although this leads to somewhat unfair comparison for BTC, it is nonetheless instructive to investigate its performance vis-a-vis these competitors. Finally, we also employ (12) after fixing the true number of clusters and the true values of $\mathbf{\Sigma}_{k,h}$'s for each tensor normal mixture component. This competitor is referred to as the Oracle Bayesian tensor clustering approach, where the only parameters left to estimate are the weights of the mixture components. Oracle is generally expected to perform better than all the approaches and is used to assess the loss in performance due to various approximations in our approach. Notably, Oracle competitor is only available for simulation studies.

To assess inference on clusters from BTC, we look at (i) the point estimate of cluster membership indicators denoted by $\hat{\mathbf{c}}$, and (ii) a heatmap of the posterior probability of any two samples belonging to the same cluster, or the co-clustering matrix $\mathbf{G}$ with the (i, j)th entry $P(c_i = c_j|\text{Data})$ (which provides a measure of the uncertainty associated with the

clustering). An empirical estimate of the co-clustering matrix $\mathbf{G}$ can be obtained from the post burn-in MCMC samples of the cluster membership indices $\mathbf{c}$. With the information on true cluster configuration in simulation studies, we evaluate the quality of point estimate of clustering using the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) of the posterior cluster configurations with respect to the known cluster configuration. The ARI evaluates the agreement in cluster assignment between two cluster configurations. It ranges between -1 and 1, with larger values indicating more agreement between cluster configurations. Notably, ARI is only available for simulation studies where the true clusters are known.

4.2 Simulation Results

Table 1 provide insights into the point estimates of the cluster structure by displaying the discrepancy between the true and the estimated clusters. BTC shows excellent clustering accuracy under all cases with ARI being close to 1. However, as sparsity of tensors decreases BTC tends to mis-classify a fraction of the data points, leading to a drop of ARI to 0.67 for $\alpha = 0.4$ and further deteriorating with higher values of α. The deterioration in performance can be attributed to the fact that with decreasing sparsity, the transformed features may not be able to provide an accurate estimation of the tensor covariance structure, as noted in Lemma 2.1. Further, BTC essentially clusters high-dimensional transformed features and sparsity or any low-dimensional structure favors high-dimensional clustering (Sun et al., 2012). While DEEM is supplied with the true number of clusters, it often clubs multiple clusters to a single cluster which naturally yields an under-estimation in the number of clusters and consequently, a drop of ARI values. Table 1 shows that the clustering accuracy of DEEM plummets when true number of clusters in the data increases, though sparsity does not seem to have any major impact on the clustering performance of DEEM. Can you give an explanation?. Note that DTC clusters based on the low-rank decomposition of the mean structure of each tensor which is not conducive in capturing in the present scenario, since data generating clusters mainly differ in terms of their variability. In fact, the tensors simulated from (12) are not likely to be approximated well by a low-rank decomposition, which presumably leads to the less satisfactory performance of DTC. In contrast, the "gold standard" Oracle is provided with the true covariance structure of the tensors as well as the

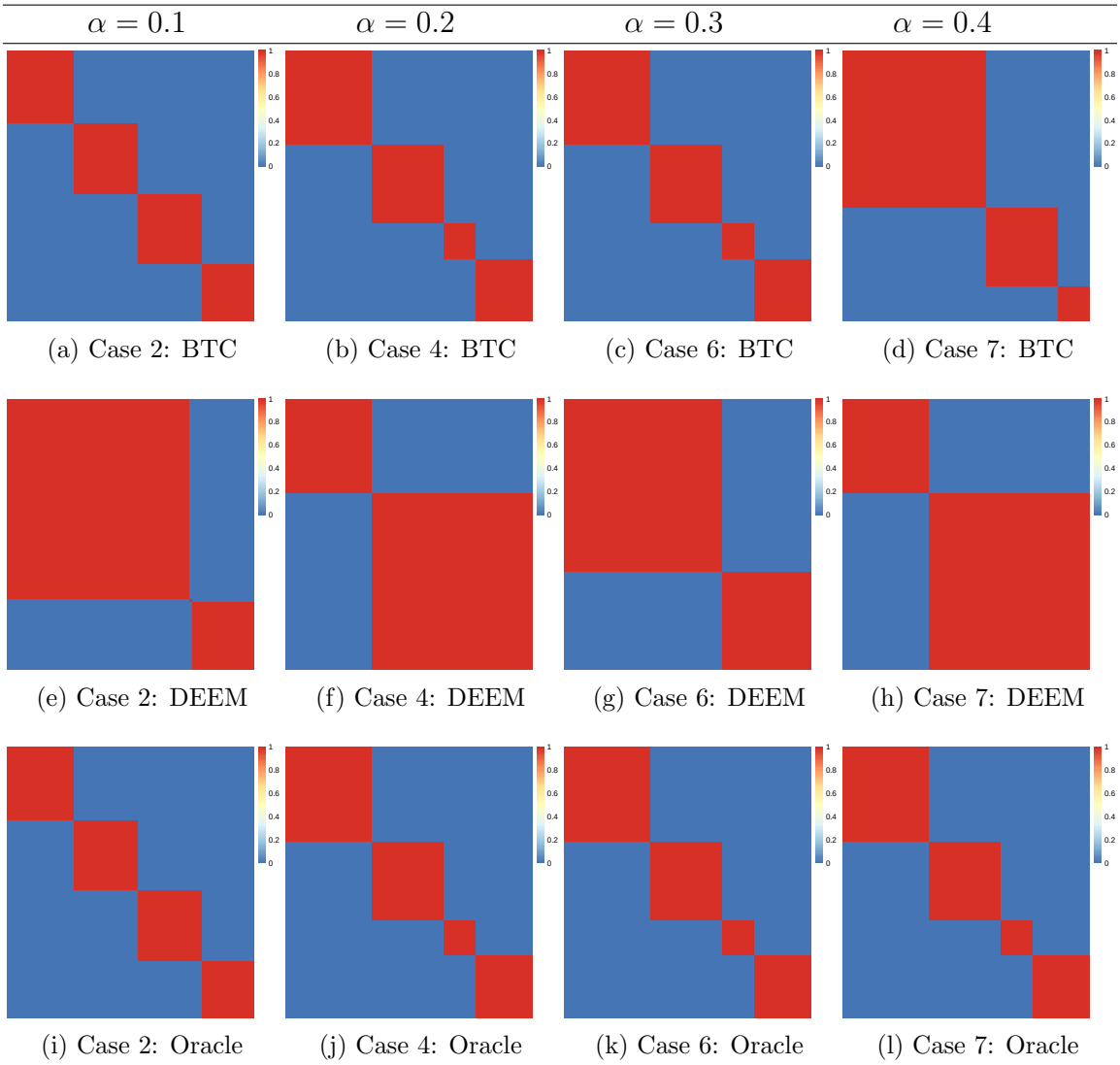
Table 1: Adjusted Rand Index (ARI) for competitors (BTC, DTC, DEEM, Oracle) for different simulation configurations.

Cases	α	H	*BTC*	*DEEM*	*DTC*	*Oracle*
1	0.1	3	0.94	0.53	0.05	0.98
2	0.1	4	1.00	0.32	0.37	1.00
3	0.2	3	0.96	0.65	0.13	0.97
4	0.2	4	1.00	0.39	0.32	1.00
5	0.3	3	0.99	0.79	0.11	0.99
6	0.3	4	1.00	0.62	0.30	1.00
7	0.4	4	0.67	0.38	0.31	0.94

true number of clusters; hence it demonstrates ARI close to 1 in every simulation. Interestingly, for higher degree of sparsity in the simulated tensors, the clustering performance of BTC and Oracle are practically indistinguishable.

The uncertainty in clustering is displayed using the heat maps of posterior probabilities of pairs of subjects belonging to the same cluster, or the co-clustering matrix. Figures 1 and 2 show co-clustering matrices for all competitors (except DTC) under all the simulation scenarios. Since DTC only offers point estimate of clusters, co-clustering matrix corresponding to DTC is not available. To facilitate visualization in Figures 1 and 2, subjects are ordered according to their true cluster configurations in the heatmap. In cases 1-6, BTC successfully recovers the true cluster structure, with little uncertainty associated with the estimator. With decreasing sparsity, the clustering performance deteriorates as demonstrated by case 7. However, even in case 7, where the BTC framework falls short of recovering the true cluster structure, we find less uncertainty in the cluster estimation. As discussed before, DEEM often produces less accurate clusters, though it does so with a very little uncertainty. Oracle also recovers true clusters with very little uncertainty. In general, BTC appears to be a competitive clustering approach when tensors are sparse. Importantly, unlike existing model-based tensor clustering approaches, high dimensionality of tensors is a blessing rather than a curse for BTC as with high dimensions, the transformed features can more accurately recover the true covariance matrices. This offers crucial advantage to BTC in neuroscietific applications where high resolution tensors are routinely collected.

Figure 1: Heatmap of the posterior probability of any two samples belonging to the same cluster. For the cases with $H = 4$.



(a) Case 2: BTC (b) Case 4: BTC (c) Case 6: BTC (d) Case 7: BTC

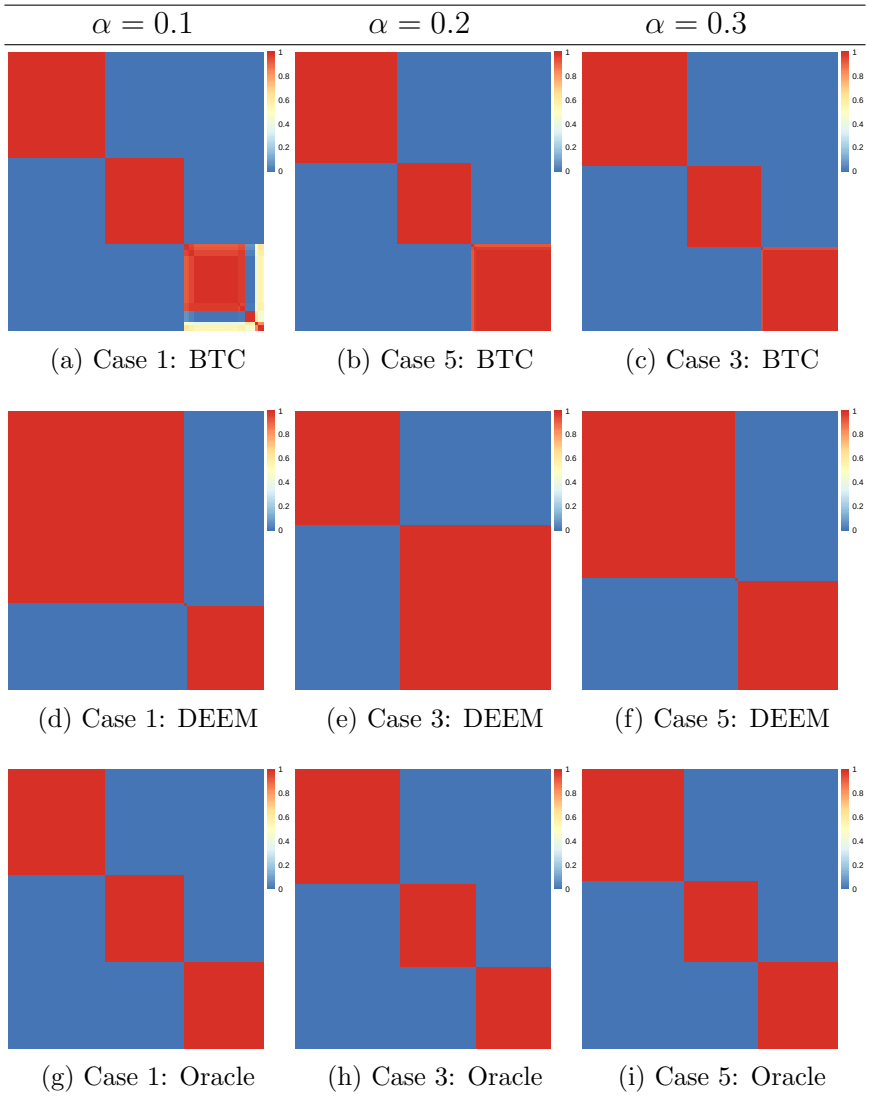
(e) Case 2: DEEM (f) Case 4: DEEM (g) Case 6: DEEM (h) Case 7: DEEM

(i) Case 2: Oracle (j) Case 4: Oracle (k) Case 6: Oracle (l) Case 7: Oracle

5 EEG Data Application

We illustrate performance of BTC using a dataset on EEG signals for 58 children aged 25 to 126 months with autism spectrum disorder (ASD). For each subject, EEG signals were sampled at 500 HZ for two minutes from a 128-channel HydroCel Geodesic sensor Net. EEG recordings were collected during an 'eyes-open' paradigm in which bubbles were displayed on a screen in a sound-attenuated room to subjects at rest. More details related to pre-processing and data acquisition can be found at Scheffler et al. (2019). The EEG data for

Figure 2: Heatmap of the posterior probability of any two samples belonging to the same cluster. For the cases with $H = 3$.



(a) Case 1: BTC (b) Case 5: BTC (c) Case 3: BTC

(d) Case 1: DEEM (e) Case 3: DEEM (f) Case 5: DEEM

(g) Case 1: Oracle (h) Case 3: Oracle (i) Case 5: Oracle

each subject is interpolated down to a standard $10 - 20$ system 25 electrode montage using interpolation as discussed in Perrin et al. (1989), producing 25 electrodes with continuous EEG signal. We obtained spectral density estimates on the first 38 seconds of artifact free EEG data, across subjects, using the Fast Fourier Transform described in Welch (1967) with two second Hanning windows and 50 percent overlap. We further restrict our data to the alpha spectral band ($\Omega = (6\text{Hz}, 14\text{Hz})$) which due to the sampling scheme has a frequency resolution of 0.25Hz resulting in 33 functional grid points. Finally, we normalize this band to a unit area to better facilitate comparisons across electrodes and subjects. As a result we

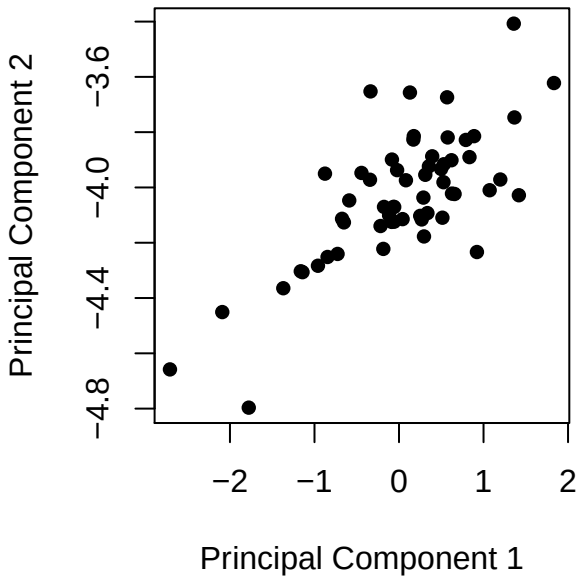
end up with 58 two-way tensors (or matrices) of dimensions 25×33.

Prior evidence suggests patients with autism spectrum disorder (ASD) can be clustered based on EEG recordings with substantial heterogeneity in cluster-specific mean and covariance structures **?**. In a previous analysis of our motivating alpha spectral density EEG data, **?** found a common alpha spectral mean structure across ASD patients 2-12 years old. However, patients exhibited substantial heterogeneity in terms of alpha spectral dynamics across the scalp. Thus, in this application, it is of interest to determine if ASD patients cluster in terms of patterns of variation rather than mean structure as most unsupervised approaches consider. Potential subgroups with cluster-specific covariances can be investigated for links to observed characteristics such as age, gender, or verbal and non-verbal intelligence quotients (VIQ and NVIQ, respectively).

We apply the approximate tensor clustering framework of BTC to the collection of this 58 tensors. Since the BTC approach is mainly designed to address clustering of tensors which are similar in their centers but show difference in variability, it is instructive to investigate if the EEG dataset exhibits such a structure. While it is hard to verify such an assumption in high dimensional objects, two separate exploratory analyses are presented to investigate this issue on this dataset. As part of our first exploratory analysis, we compute principal Components of the data matrices and present a plot for the first two principal components (see Figure 3), which account for 42.39% of the total variability in the observations. No clustering of the first two principle components is apparent here, with the first two principal components for observations are smoothly distributed instead of being clustered in groups. While this offers no guarantee, one might expect that a meaningful cluster difference in the location of the observations would become apparent here.

To investigate this issue further, we vectorize each 25×33 tensor to a long vector of 825 co-ordinates and perform k-means clustering separately on each of these co-ordinates. If several of the coordinates show similar clustering pattern, then one might intuitively expect that there is a meaningful difference in the cluster means. We compute the similarity of coordinate clustering by computing the ARI of every coordinate cluster against every other coordinate cluster, resulting in a possible $\binom{825}{2}$ ARI values. We perform this analysis for k-means with $k = 2, 3, 4$ and 5 clusters. Table 2 presents the 5th, 25th, 50th, 75th and

Figure 3: Observations visualizations: we present the first two principal components after performing Principal Component Analysis.



95th percentile values for ARI corresponding to $k = 2, 3, 4, 5$. The results demonstrate the distribution of the ARI is concentrated around 0 for all choices of k, offering no evidence that a significant number of coordinates results in similar clusters. $K-$means clustering with higher values of k leads to even lower degree of concordance between clustering of samples along different dimensions.

Table 2: Summary statistics of the coordinate clustering similarity computed by ARI.

Means	*5th percentile*	*25th percentile*	*Median*	*75th percentile*	*95th percentile*
$k = 2$	-0.06940	-0.023240	-0.003562	0.06126	0.2623
$k = 3$	-0.02938	-0.010196	0.015847	0.06482	0.1857
$k = 4$	-0.02837	-0.005866	0.019221	0.05750	0.1400
$k = 5$	-0.02692	-0.003716	0.018981	0.05024	0.1162

With the preliminary exploration indicating no difference in clusters in terms of mean, we proceed to identify clusters with differences in their variability using BTC. BTC shows rapid convergence and is run for 400 iterations, out of which first 100 is used as burn-in and inference is based on post burn-in iterates. The posterior distribution of the number

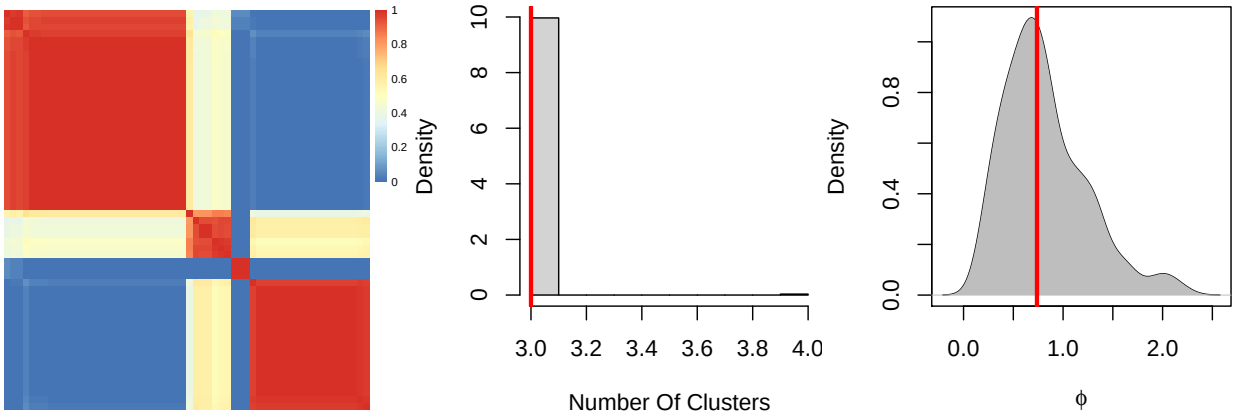
of clusters in Figure 4b shows a clear mode at 3, indicating three clusters among subjects. The co-clustering matrix shown in Figure 4a indicates four clusters with a high degree of uncertainty in the cluster membership for elements in the first two clusters. Indeed, the result indicates that the elements in the second cluster are often included as part of the first cluster in post burn-in iterates, which is consistent with the posterior mode of the number of clusters being identified as three. In Figure 4c we observe that the posterior distribution of ϕ in our approximate Bayesian clustering approach concentrates around 1, which is equivalent to using a Chinese restaurant approach with a person already seated in each table.

To demonstrate the stability of clusters in the post burn-in iterations, we plot (Figure 5) ARI of clusters in any two successive post burn-in iterations. The plot indicates that most of the partitions in successive iterations are identical or have high overlaps. The nominal degree of fluctuations in the ARI stems mainly from the fact that elements in the second cluster are entirely part of the first cluster in many of the iterations.

We further investigate the three clusters identified by BTC. The three clusters include 30, 25 and 3 subjects. The groups are contrasted across four covariates measured on the sample: gender, age, VIQ, and NVIQ. The three clusters varied significantly with respect to NVIQ (p-value = 0.021) and borderline significance with respect to VIQ (p-value = 0.065). These results seem driven largely by the third cluster which only contains three subjects. If we remove this cluster, there are no more significant contrasts. Ultimately, an unsupervised tensor clustering analysis is inherently exploratory, and the identified clusters form the basis of identifying ASD phenotypes of interest by fitting a sophisticated cluster specific model.

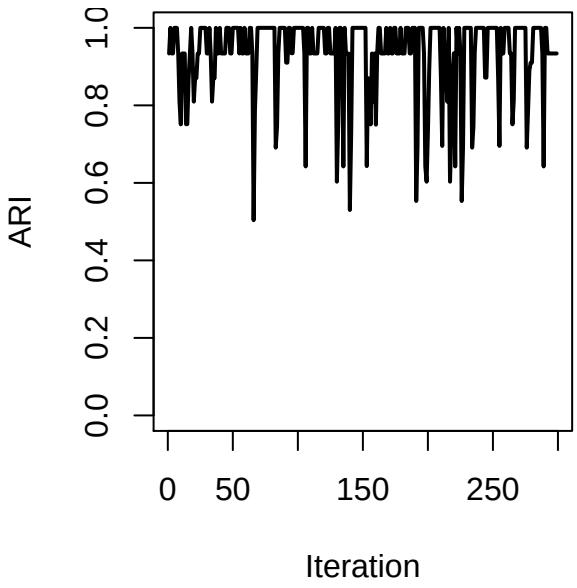
Since the size of the tensors in the EEG data application is smaller than the simulation studies, they allow fitting a full Bayesian mixture model analysis of the data using matrix normal distributions with zero mean as mixture components. This approach also clusters tensors based on their variability, but without any approximation as in BTC. The Bayesian mixture modeling approach should ideally offer better clustering performance than DTC, since DTC is essentially a clustering technique that clusters tensors based on the difference in their centers. As the true model parameters are not available for the real data, we are unable to present the Oracle. Figure 6 presents co-clustering matrices for the full Bayesian implementation for a mixture of $k = 3, 4, 5$ matrix normal distributions. The figure demon-

Figure 4: Cluster structure for EEG data on 58 ASD children.



(a) Cluster Probability Matrix (b) Histogram Number of Clusters (c) Density ϕ

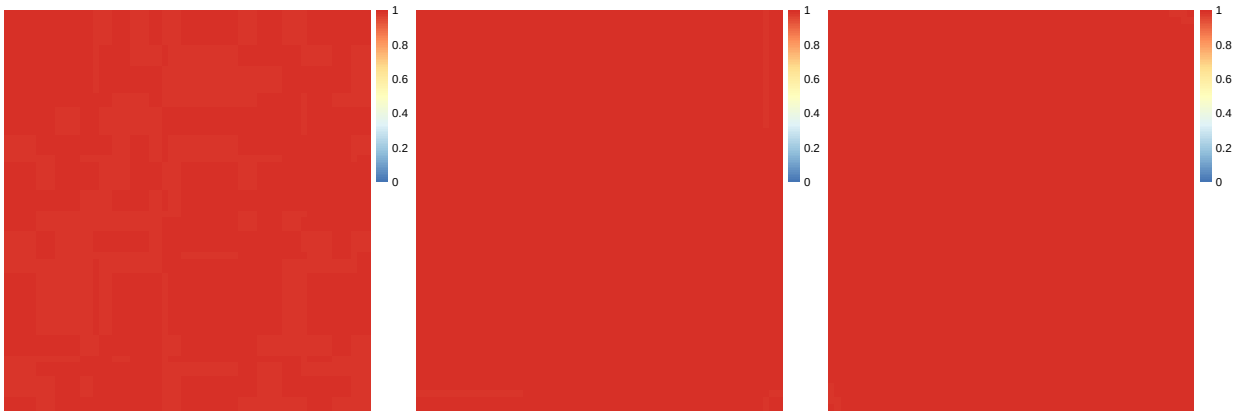
Figure 5: ARI of each partition with respect to the previous partition throughout the 300 MCMC iterations sequentially.



strates unsatisfactory performance of the full Bayesian clustering approach , showing only one cluster. This is somewhat expected based on the performance of DEEM in the simulation studies. DEEM is a frequentist analogue to the Bayesian mixture model and it is found to underestimate the true number of clusters under all cases in the simulation study. Importantly, even with a full Bayesian implementation, the complexity of the real data combined with a moderate sample size, makes the clustering results from the full Bayesian mixture model of matrix normal distributions practically useless in our real data. Furthermore, the

BTC approximation is computationally less expensive than the full Bayesian mixture model, as presented in Table 3.

Figure 6: Cluster structure for EEG data on 58 ASD children using a full Bayesian mixture of matrix normals.



(a) $K = 5$ (b) $K = 4$ (c) $K = 3$

Table 3: Runtime (in seconds) for 400 iterations for BTC and Full Bayesian implementation for $K = 5, 4, 3$ for ASD data.

Method	*Runtime (secs)*
BTC	148.10
Full Bayesian $k = 5$	341.20
Full Bayesian $k = 4$	271.03
Full Bayesian $k = 3$	211.72

References

Anderlucci, L., C. Viroli, et al. (2015). Covariance pattern mixture models for the analysis of multivariate heterogeneous longitudinal data. *The Annals of Applied Statistics 9*(2), 777–800.

Antoniak, C. E. (1974). Mixtures of dirichlet processes with applications to bayesian non-parametric problems. *The annals of statistics*, 1152–1174.

Banerjee, A., S. Merugu, I. Dhillon, and J. Ghosh (2004, April). Clustering with Bregman Divergences. In *Proceedings of the 2004 SIAM International Conference on Data Mining*, pp. 234–245. Society for Industrial and Applied Mathematics.

Cai, T., W. Liu, and X. Luo (2011, June). A constrained l1 minimization approach to sparse precision matrix estimation. *Journal of the American Statistical Association 106*(494), 594–607.

Celeux, G., K. Kamary, G. Malsiner-Walli, J.-M. Marin, and C. P. Robert (2019). Computational solutions for bayesian inference in mixture models. In *Handbook of Mixture Analysis*, pp. 73–96. Chapman and Hall/CRC.

Chi, E. C., B. J. Gaines, W. W. Sun, H. Zhou, and J. Yang (2020). Provable convex co-clustering of tensors. *J. Mach. Learn. Res. 21*, 214–1.

Chi, E. C. and T. G. Kolda (2012). On tensors, sparsity, and nonnegative factorizations. *SIAM Journal on Matrix Analysis and Applications 33*(4), 1272–1299.

Dunson, D. B. and C. Xing (2009). Nonparametric bayes modeling of multivariate categorical data. *Journal of the American Statistical Association 104*(487), 1042–1051.

Ferguson, T. S. (1973). A bayesian analysis of some nonparametric problems. *The annals of statistics*, 209–230.

Fraley, C. and A. E. Raftery (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American statistical Association 97*(458), 611–631.

Fritsch, A., K. Ickstadt, et al. (2009). Improved criteria for clustering based on the posterior similarity matrix. *Bayesian analysis 4*(2), 367–391.

Fröhwirth-Schnatter, S. and S. Kaufmann (2008). Model-based clustering of multiple time series. *Journal of Business & Economic Statistics 26*(1), 78–89.

Frühwirth-Schnatter, S. (2006). *Finite mixture and Markov switching models*. Springer Science & Business Media.

Gao, X., W. Shen, L. Zhang, J. Hu, N. J. Fortin, R. D. Frostig, and H. Ombao (2020). Regularized matrix data clustering and its application to image analysis. *Biometrics*.

Gopalan, R. and D. A. Berry (1998). Bayesian multiple comparisons using dirichlet process priors. *Journal of the American Statistical Association 93*(443), 1130–1139.

Guha, S. and R. Guhaniyogi (2020). Bayesian generalized sparse symmetric tensor-on-vector regression. *Technometrics*, 1–11.

Guhaniyogi, R., S. Qamar, and D. B. Dunson (2017). Bayesian tensor regression. *The Journal of Machine Learning Research 18*(1), 2733–2763.

Guhaniyogi, R. and D. Spencer (2018). Bayesian tensor response regression with an application to brain activation studies. Technical report, Technical report, UCSC.

Hallac, D., S. Vare, S. Boyd, and J. Leskovec (2018). Toeplitz inverse covariance-based clustering of multivariate time series data. http://arxiv.org/abs/1706.03161.

Hartigan, J. A. and M. A. Wong (1979). Algorithm AS 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics) 28*(1), 100–108. Publisher: [Wiley, Royal Statistical Society].

Huang, H., C. Ding, D. Luo, and T. Li (2008). Simultaneous tensor subspace selection and clustering: the equivalence of high order svd and k-means clustering. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge Discovery and Data mining*, pp. 327–335.

Hubert, L. and P. Arabie (1985, December). Comparing partitions. *Journal of Classification 2*(1), 193–218.

Ieva, F., A. Paganoni, and N. Tarabelloni (2016). Covariance-based clustering in multivariate and functional data analysis. *Journal of Machine Learning Research 17*, 1–21.

Jegelka, S., S. Sra, and A. Banerjee (2009). Approximation algorithms for tensor clustering. In *International Conference on Algorithmic Learning Theory*, pp. 368–383. Springer.

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

Lau, J. W. and P. J. Green (2007, September). Bayesian Model-Based Clustering Procedures. *Journal of Computational and Graphical Statistics 16*(3), 526–558.

Lee, J., P. Müller, Y. Zhu, and Y. Ji (2013). A nonparametric bayesian model for local clustering with application to proteomics. *Journal of the American Statistical Association 108*(503), 775–788.

Lee, M., H. Shen, J. Z. Huang, and J. Marron (2010). Biclustering via sparse singular value decomposition. *Biometrics 66*(4), 1087–1095.

Liu, C. and R. Martin (2019, December). An empirical G-Wishart prior for sparse high-dimensional Gaussian graphical models. arXiv: 1912.03807.

Lock, E. F. (2018). Tensor-on-tensor regression. *Journal of Computational and Graphical Statistics 27*(3), 638–647.

Mai, Q., X. Zhang, Y. Pan, and K. Deng (2021a). A doubly enhanced em algorithm for model-based tensor clustering. *Journal of the American Statistical Association*, 1–15.

Mai, Q., X. Zhang, Y. Pan, and K. Deng (2021b, March). A Doubly Enhanced EM Algorithm for Model-Based Tensor Clustering. *Journal of the American Statistical Association 0*(0), 1–15. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/01621459.2021.1904959.

Medvedovic, M. and S. Sivaganesan (2002). Bayesian infinite mixture model based clustering of gene expression profiles. *Bioinformatics 18*(9), 1194–1206.

Medvedovic, M., K. Y. Yeung, and R. E. Bumgarner (2004). Bayesian mixture model based clustering of replicated microarray data. *Bioinformatics 20*(8), 1222–1232.

Müller, P., F. A. Quintana, A. Jara, and T. Hanson (2015). *Bayesian nonparametric data analysis*. Springer.

Oh, M.-S. and A. E. Raftery (2007). Model-based clustering with dissimilarities: A bayesian approach. *Journal of Computational and Graphical Statistics 16*(3), 559–585.

Pan, W. and X. Shen (2007). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research 8*(5).

Perrin, F., J. Pernier, O. Bertrand, and J. F. Echallier (1989, February). Spherical splines for scalp potential and current density mapping. *Electroencephalography and Clinical Neurophysiology 72*(2), 184–187.

Raftery, A. E. and N. Dean (2006). Variable selection for model-based clustering. *Journal of the American Statistical Association 101*(473), 168–178.

Rothman, A. J., P. J. Bickel, E. Levina, and J. Zhu (2008). Sparse permutation invariant covariance estimation. *Electronic Journal of Statistics 2*(0), 494–515.

Scheffler, A. W., D. Telesca, C. A. Sugar, S. Jeste, A. Dickinson, C. DiStefano, and D. Şentürk (2019, December). Covariate-adjusted region-referenced generalized functional linear model for EEG data. *Statistics in medicine 38*(30), 5587–5602.

Spencer, D., R. Guhaniyogi, and R. Prado (2020). Joint bayesian estimation of voxel activation and inter-regional connectivity in fmri experiments. *Psychometrika*, 1–25.

Stephens, M. (2000). Dealing with label switching in mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 62*(4), 795–809.

Sun, W., J. Wang, and Y. Fang (2012). Regularized k-means clustering of high-dimensional data and its asymptotic consistency. *Electronic Journal of Statistics 6*, 148–167. Publisher: The Institute of Mathematical Statistics and the Bernoulli Society.

Sun, W. W. and L. Li (2019a). Dynamic tensor clustering. *Journal of the American Statistical Association 114*(528), 1894–1907.

Sun, W. W. and L. Li (2019b, October). Dynamic Tensor Clustering. *Journal of the American Statistical Association 114*(528), 1894–1907.

Tan, K. M. and D. M. Witten (2014). Sparse biclustering of transposable data. *Journal of Computational and Graphical Statistics 23*(4), 985–1008.

Viroli, C. (2011). Finite mixtures of matrix normal distributions for classifying three-way data. *Statistics and Computing 21*(4), 511–522.

Wang, M. and Y. Zeng (2019). Multiway clustering via tensor block models. *arXiv preprint arXiv:1906.03807*.

Wang, S. and J. Zhu (2008). Variable selection for model-based high-dimensional clustering and its application to microarray data. *Biometrics 64*(2), 440–448.

Welch, P. (1967, June). The use of fast Fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms. *IEEE Transactions on Audio and Electroacoustics 15*(2), 70–73. Conference Name: IEEE Transactions on Audio and Electroacoustics.

Zhong, S. and J. Ghosh (2003). A unified framework for model-based clustering. *The Journal of Machine Learning Research 4*, 1001–1037.

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.