UNIVERSITY OF CALIFORNIA
SANTA CRUZ

HIGH-DIMENSIONAL INFERENCE AND UNCERTAINTY QUANTIFICATION FOR VARIABLE SELECTION, CLUSTERING AND OBJECT-ORIENTED ANALYSIS WITH BAYESIAN AND APPROXIMATE BAYESIAN METHODS

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

STATISTICAL SCIENCE

by

Rene Gutierrez

December 2021

The Dissertation of Rene Gutierrez
is approved:

Rajarshi Guhaniyogi, Chair

Raquel Prado

Herbert Lee

Peter Biehl
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

High-Dimensional Inference and Uncertainty Quantification for Variable Selection, Clustering and Object-oriented Analysis with Bayesian and Approximate Bayesian Methods

by

Rene Gutierrez

Bayesian computation of High-Dimensional problems using Markov Chain Monte Carlo (MCMC) or its variants can be extremely slow or completely prohibitive since these methods perform costly computations at each iteration of the sampling chain. While some non-Bayesian alternatives have been somewhat successful in estimation, they struggle to provide uncertainty quantification. These problems are aggravated if the data size is large. To address these problems, the first chapter proposes a novel dynamic feature partitioned regression (DFP) for efficient online inference for high dimensional linear regressions with large or streaming data. DFP constructs a *pseudo posterior density* of the parameters at every time point and quickly updates the pseudo posterior when a new block of data (data shard) arrives. DFP updates the pseudo posterior at every time point suitably and partitions the set of parameters to exploit parallelization for efficient posterior computation. The proposed approach is applied to high dimensional linear regression models with Gaussian scale mixture priors and spike and slab priors on large parameter spaces, along with large data, and yields state-of-the-art inferential performance. Over time, the algorithm enjoys theoretical support, as pseudo posterior densities get arbitrarily close to the full posterior as the data size grows, as shown in the appendix.

While the first chapter advances methodology for ordinary high dimensional

regression, the second chapter focuses on regressions with multiple objects as predictors. Clinical researchers often collect multiple images from separate modalities (sources) to investigate fundamental questions of human health that are inadequately explained by considering one image source at a time. Viewing the collection of images as multiple objects, the successful integration of multi-object data produces a sum of information greater than the individual parts. This chapter is motivated by a multi-modal imaging application where structural/anatomical information from grey matter (GM) and brain connectivity information in the form of a brain connectome network from functional magnetic resonance imaging (fMRI) are available for multiple subjects. The primary goal in this chapter is to develop a regression model to predict a scalar response from multiple objects, and to identify regions significantly related to the response. Existing Bayesian regression literature with multi-object predictors either ignores the topology of some/all of these objects or does not adequately make use of the information shared by multiple object predictors. In contrast, this chapter develops a flexible Bayesian regression framework exploiting network information of the brain connectome while leveraging linkages among connectome network and anatomical information from GM to draw inference on significant ROIs and offer predictive inference on the response. The principled Bayesian framework allows precise characterization of the uncertainty in ascertaining a region as influential for predicting the response and the quantification of predictive uncertainty for the response. The framework is implemented using an efficient Markov Chain Monte Carlo algorithm. Empirical results in simulation studies illustrate substantial inferential and predictive gains of the proposed framework over its popular competitors.

While the first two chapters focus on high-dimensional and object-oriented regressions, the third chapter offers a novel clustering technique for high-dimensional

tensors with limited sample size. 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, the sample size is small, and clusters of tensors mainly reveal differences in their variability. This chapter develops a novel clustering technique for high dimensional tensors with limited sample sizes when the clusters show differences in their covariances rather than their means. The proposed approach constructs several matrices from a tensor to adequately estimate its variability along with different modes and implements a model-based approximate Bayesian clustering algorithm with the matrices, thus constructed with the original tensor data. Although some information in the data is discarded, we gain substantial computational efficiency and accuracy in clustering. The simulation study assesses the proposed approach and its competitors in terms of estimating the number of clusters, identifying the modal cluster membership, and the probability of misclassification in clustering (a measure of uncertainty in clustering). Clustering of tensors obtained from EEG data demonstrates an advantage of the proposed approach vis-a-vis its competitors.

To my parents.

Acknowledgments

I am incredibly grateful to my advisor, Raj Guhaniyogi, an essential part of this work. His guidance in the research process was the perfect combination of support and freedom that allowed me to develop as a researcher. It has left an invaluable mark in my academic style that I will carry proudly with me.

I am also thankful to the Statistics faculty that provided me with the tools and knowledge to do my research and offered their support throughout this journey. The Department of Applied Mathematics, Computer Science and Language and Applied Linguistics faculty allowed me to expand my knowledge and satisfy my curiosity.

I am thankful to have shared this experience with the students in the Departments of Statistics and Applied Mathematics. I have learned a lot from them through their friendship and advice.

I have always found support in my parents, brothers, and sister. It has been extremely helpful to know that I can always count on them.

I appreciate the support of CONACYT and UC MEXUS. Their financial support is of great importance for Mexican students.

Finally, I would like to thank Aaron Scheffler, a great research collaborator that not only provided data for two chapters of this work but helped me understand its neurological nature.

Chapter 1

Introduction

With recent technological progress, complex data structures are ubiquitous in scientific applications. For example, scientific applications often present scenarios where there are many variables and a large sample size with complex interdependence between variables. Another such data structure is that of the tensor, also known as a multidimensional array. This data structure expands on the notion of a vector or matrix into higher dimensions, typically codifying information about a datum and its position within the tensor.

Scenarios with high-dimensional variables and massive sample size are often encountered in financial data. For example, real-time data on a large number of stocks are easily available, and it may be of interest to predict the progression of a stock based on the progression of many other stocks. On the other hand, tensor-valued data are mainly encountered in medical imaging. Imaging scanners use various methods to obtain information about different body parts, such as a lung or a brain. These images are sometimes two-dimensional, showing a slice of a three-dimensional structure. In other cases, the images are three-dimensional, capturing information throughout an entire organ or body part. Accurate analytical techniques for these types of data are useful for quantifying medical diagnosis

and treatment. Additionally, there are scenarios where multiple objects are presented in the analysis. These scenarios are mainly motivated by the multi-modal imaging data.

In the following chapters, three different methods for analyzing high-dimensional regressions and tensor objects are presented. The next section provides a brief overview of the three chapters.

1.1 Thesis Outline

Chapter 2 will introduce a novel online approximate Bayesian algorithm, referred to as the dynamic feature partition (DFP), for drawing inference with high dimensional regressions with large sample size and a large number of predictors. The approach divides big data into shards and sequentially feeds the shards to the regression model. At each time point with the arrival of a data shard, the posterior distributions of the parameters are updated through a hybrid procedure that includes point estimates and Markov chain Monte Carlo samples. Further, while updating the posterior for parameters, conditional independence is assumed between blocks of parameters that have relatively smaller correlations between themselves. Unlike variational Bayes approaches, our framework refrains from assuming marginal independence between blocks of parameters. Rather, we assume conditional independence between blocks of parameters that are determined by the model fitting process. The entire exercise gives rise to a highly efficient approximate Bayesian approach. We evaluate the performance of this approach with continuous and discrete Gaussian scale mixture priors. The method has also been illustrated through a financial dataset. Appendix A provides a theoretical result proving the algorithm asymptotically draws samples from the full posterior distribution. To the best of our knowledge, the DFP approach is the first approach

to provide theoretically guaranteed and computationally efficient Bayesian inference in high dimensional regression with large sample size and a large number of predictors.

While high dimensional regression has gained enormous attention from the statistical community in the last decade, lately, there has been an increasing focus on regressions where either response or predictor is an object. We broadly refer to such regression scenarios as Object data regression. Object data regression approaches are primarily motivated by imaging data obtained from different imaging modalities and develop inferential tools to arrive at scientifically meaningful conclusions from such datasets by harnessing their inherent topological structure. Some of the important advances in this area include regression of a scalar variable on a tensor or a network predictor and regressions of tensor/network variables on vector predictors. However, there is still a dearth of methodology in object data regressions involving multi-object predictors obtained from multiple imaging modalities. Chapter 3 introduces a novel multi-object regression approach where a scalar response (a specific phenotype) is regressed on grey matter (GM) and brain connectivity network. More specifically, we develop prior distributions which simultaneously achieve several objectives. First, it allows inference on influential brain ROIs significantly related to the phenotype of interest from multi-object predictors jointly. Second, the prior artifact respects structure of the brain network and imposes the condition that if an ROI is unimportant in predicting the response, all network edges related to the ROI are also unimportant in predicting the response. Third, if an ROI is unimportant in predicting the response, all voxels within an ROI are also un-influential in predicting the response. Fourth, it satisfies the transitivity property of the brain network predictors introduced in Chapter 3. Finally, the framework allows predictive inference for a phenotype with

multi-modal predictors. We demonstrate the importance of exploiting the topological structures of different predictors and the linkage of information between them to draw superior inference to models that partially use such information. To the best of our knowledge, this is the first approach that develops a multimodal regression framework with a scalar response that simultaneously addresses all these inferential questions.

Finally, Chapter 4 addresses an important problem of clustering high dimensional tensors where clusters only differ in the variabilities rather than their means. Clustering has been an extremely well-researched topic in the Bayesian paradigm. However, most unsupervised clustering efforts are limited to scalar- or vector-valued objects, and there is a relatively sparse Bayesian literature for clustering tensor objects. While it is possible to vectorize a tensor and apply algorithmic clustering approaches, they mainly cluster subjects based on the difference in their means and are not applicable in our settings. Bayesian mixture model-based clustering circumvents this problem, though they are computationally tedious when the tensor dimensions are large. Moreover, they yield inaccurate inference when both tensor dimensions are large, and the sample size is small. As a solution to this problem, we develop an approximate Bayesian clustering approach based on the estimated covariance structure of the tensor data. Our approach offers rapid cluster identification along with clustering uncertainties. Theoretical results and details of posterior computation corresponding to each chapter are included in separate chapters in the Appendix.

Chapter 2

Bayesian Dynamic Feature Partitioning in High-Dimensional Regression with Big Data

2.1 Introduction

With recent technological progress, data containing many predictors (a couple of thousand or more) are ubiquitous. In such settings, it is commonly of interest to consider the linear regression model

$$y = \boldsymbol{x}'\boldsymbol{\beta} + \epsilon, \;\; \epsilon \sim N(0, \sigma^2), \tag{2.1}$$

where $\boldsymbol{x}$ is a $p \times 1$ predictor, $\boldsymbol{\beta}$ is the corresponding $p \times 1$ coefficient, y is the continuous response and σ^2 is the error variance. Bayesian methods for estimating $\boldsymbol{\beta}$ provide a natural probabilistic characterization of uncertainty in the parameters and predictions. Fitting Bayesian linear regression models in the presence of very high dimensional predictors present onerous computational burdens either due to

decomposition of large matrices or due to poor convergence and inferential issues caused by the high correlations among the parameters. This chapter develops a dynamic approach, called Dynamic Feature Partitioning (DFP), for boosting the scalability of high dimensional Bayesian linear models for large/streaming data.

Broadly, two classes of prior distributions on $\boldsymbol{\beta}$ are typically employed in high dimensional regression literature. The traditional approach is to develop a discrete mixture of prior distributions (George and McCulloch, 1997; Scott and Berger, 2010). These methods enjoy the advantage of inducing exact sparsity for a subset of parameters and minimax rate of posterior contraction (Castillo et al., 2015) in high dimensional regression, but face computational challenges when the number of predictors is even moderately large. As an alternative to this approach, continuous shrinkage priors (Armagan et al., 2013; Carvalho et al., 2010) have emerged which induce approximate sparsity in high-dimensional parameters. Such prior distributions can mostly be expressed as global-local scale mixtures of Gaussians (Polson and Scott, 2010) and offer an approximation to the operating characteristics of discrete mixture priors. Global-local priors allow parameters to be updated in blocks via a fairly automatic Gibbs sampler, leading to rapid mixing and convergence of the resulting Gibbs sampler. However, unless care is exercised, sampling can be expensive for large values of p. In fact, existing algorithms (Rue, 2001) to sample from the full conditional posterior of $\boldsymbol{\beta}$ require storing and computing the Cholesky decomposition of a $p \times p$ matrix, which necessitates p^3 floating-point operations, which can be severely prohibitive for large p. There are available linear algebra artifacts such as the Sherman-Woodbury-Morrison matrix identity (Hager, 1989) to enable efficient computations in high dimensional regressions involving small n and large p. However, it is unclear how these approaches can be adapted when the number of samples is massive, or data is observed in a

stream. Besides, having a small sample size may limit the inferential accuracy for large p.

In fact, when the number of observations is massive, data processing and computational bottlenecks render all the above-mentioned methods for high dimensional regression infeasible as they demand likelihood evaluations for updating model parameters at every sampling iteration, which can be costly. Matters are more complicated in the case of streaming data, where the posterior distribution changes once a new data shard arrives since the MCMC samples from the posterior distribution up to the last time point become useless.

We propose a novel online Bayesian sampling algorithm, referred to as Dynamic Feature Partitioning (DFP) that enables efficient computation of high dimensional regression in the presence of a large number of parameters and a large sample size. DFP works with data shards that are sequentially fed to the model. The DFP framework *dynamically* partitions the set of parameters into disjoint subsets with the onset of a new data shard and obtains posterior samples for each subset of the parameters by sampling from a distribution that conditions on functions of the point estimates of the remaining parameters and sufficient statistics from the data observed so far, instead of sampling from the full conditional distribution. While the ordinary un-approximated full conditional posterior distributions of these parameter subsets would have been updated sequentially at each iteration of the Markov Chain, DFP constructs approximations of the conditional posterior distributions of each parameter subset, allowing posterior updates of these parameter subsets at different processors in parallel. This leads to a significant gain in computational efficiency over the sequential updating of parameter subsets in the ordinary MCMC. Additionally, the algorithm needs storing and propagating only a few lower-dimensional sufficient statistics of the data over

time, implying storage efficiency in the model fitting procedure. Moreover, we show that the DFP algorithm approximates the conditional distributions producing samples from the correct target posterior asymptotically. The DFP algorithm is demonstrated to be highly versatile and efficient across a variety of high dimensional linear regression settings, enabling online sampling of parameters with dramatic reductions in the per-iteration computational requirement.

We now offer a brief description of some of the important approaches in online Bayesian learning and highlight the contribution of the DFP algorithm to the literature. To this end, online variational Bayes algorithms perform an approximation of the full data posterior with a product of block independent marginal posteriors (Hoffman et al., 2010; Campbell et al., 2015) and are popular for efficient online Bayesian learning for streaming data. Although the DFP framework proposes approximating the full posterior distribution, the approximation technique is fundamentally different from variational approximations. While variational Bayes approximates the full posterior distribution by a distribution with block independent marginals, the DFP framework invokes approximation of posterior conditional distributions for subsets of parameters. More importantly, variational approximations often pre-decide parameter blocks considered independent in the posterior inference, while DFP dynamically adapts to ensure the efficient construction of mutually exhaustive and exclusive subsets of parameters. As a result, variational approximation may underestimate uncertainty from the variationally approximated posterior distribution of $\boldsymbol{\beta}$, while DFP is demonstrated to have close to nominal coverage in almost all high-dimensional simulation examples.

In the general Bayesian literature of streaming data, Sequential Monte Carlo (SMC) (Lopes and Tsay, 2011; Doucet et al., 2001; Moral et al., 2017) is one of the most popular online methods. SMC relies on resampling particles sequentially

as data shards arrive over time. A naive implementation of SMC might be less efficient and less accurate involving large n and p due to the need to employ vast numbers of particles to obtain adequate approximations and prevent particle degeneracy. The latter is addressed through rejuvenation steps using all the data (or sufficient statistics), which may become expensive in an online setting (Snyder et al., 2008). There are approaches in recent years to overcome the dimensionality issues in the SMC algorithm, mainly in the context of fitting state-space models. To this end, carefully constructed SMC algorithms (Chopin et al., 2004; Beskos et al., 2014; Carvalho et al., 2010) show promise in terms of scaling in a polynomial complexity with the number of parameters. However, the complexity as a function of the size of the dataset is growing with time (e.g., to Chopin et al. (2004)) or is not apparent from the context. Rebeschini and Handel (2015) develop a blocking strategy for high dimensional particle learning (PL) where the error of approximation is free of the dimension of the parameter space). Unfortunately, the numerical examples for high dimensions provided by Rebeschini and Handel (2015) do not demonstrate satisfactory performance with large state-space models. Furthermore, the results rely on the decay of correlations for state-space varying parameters in the fitted model, which is suitable in the context of state-space models but less satisfactory for our problem of interest. Wigren et al. (2018) propose another approach for high-dimensional particle learning in state-space models, though the numerical illustration of the approach may struggle to scale beyond a few dozen dimensional state-space models comfortably. Lindsten et al. (2017) propose a new SMC algorithm based on parameter partitioning in the high-dimensional space, though difficulties may arise when joining the partitions, which requires careful resampling. In the same vein, Gunawan et al. (2018) propose an approach that employs a sub-sampling technique to combat the problem of large

data in the realm of high-dimensional problems. Arguably, there is a general lack of extensive empirical investigations of SMC or PL algorithms proposed for high-dimensional problems, and most of them do not come with any open source code for implementation.

On a separate note, Hamiltonian Monte Carlo (HMC) methods with stochastic gradient descent can also leverage the online nature of the data (Betancourt, 2018) while exploring the distribution efficiently. However, HMC may not be suitable for computing high dimensional regressions with a discrete mixture of prior distributions involving a large number of binary variables, which can be easily accommodated by the DFP algorithm (see Section 2.2.3).

In the context of distributed model fitting in high dimensional regression, Christidis et al. (2020) have recently developed a compelling method to build an ensemble of models by splitting the set of covariates into different but possible overlapping groups. A penalty term is introduced to encourage diversity between groups, and model stacking is used to generate accurate predictions. Our approach is fundamentally different from their approach in a number of ways. While "splitting" in the context of DFP algorithm refers to partitioning of the parameters to update their conditional posterior distributions separately for computational advantages, splitting generates different models that try to achieve more accuracy when stacked in Christidis et al. (2020). Importantly, Christidis et al. (2020) is not designed to draw online inference in streaming data which is the goal of our approach. Thus, our approach allows the number and constitution of parameter partitions to evolve over time, while their approach fixes the number of partitions. Nevertheless, incorporating some overlapping in our partitioning of parameters similar to Christidis et al. (2020) might help improving inference further over the current implementation of DFP, which we plan to explore elsewhere.

The rest of the chapter is organized as follows. Section 2.2 introduces a number of shrinkage priors and variable selection priors in high dimensional regression and describes the computational challenges with big n and p. Section 2.3 introduces the assumptions, notations and then the description of the DFP algorithm. Section 2.4 demonstrates the performance of DFP for high dimensional linear regression with (1) the Bayesian Lasso and (2) the Horseshoe shrinkage prior distributions and (3) the Spike and Lasso discrete mixture prior distribution for variable selection (described in Section 2.2.3). Further evidence on the empirical performance of DFP is provided in the analysis of a financial dataset consisting of the minute-by-minute average log-prices of the NASDAQ stock exchange from September 10 2018 to November 13 2018 during trading hours in Section 2.5. Finally, Section 2.6 concludes the chapter with discussions. Theoretical insights into the convergence behavior of the DFP algorithm are provided in appendix A.

2.2 Computational Challenges in High-Dimensional Regression Models

This section motivates the need for the dynamic feature partitioning algorithm by highlighting the issues of performing online inference in Bayesian high-dimensional linear models with big or streaming data. Let $\boldsymbol{D}_t = \{\boldsymbol{X}_t, \boldsymbol{y}_t\}$ be the data (responses and predictors) shard observed at time t and $\boldsymbol{D}^{(t)} = \{\boldsymbol{D}_s, s = 1, \ldots, t\}$ denote the data observed through time t, $t = 1, ..., T$. We assume that shards are of equal size, with each shard containing n samples, i.e., $\boldsymbol{X}_t$ is of dimension $n \times p$ and $\boldsymbol{y}_t$ is of dimension $n \times 1$. We emphasize that such an assumption is not required for the algorithmic development in the next section and is kept merely to simplify notations.

In the context of the linear regression model in (2.1), without the focus being on regularization or variable selection, a Bayesian hierarchical model is set up by assigning a prior $\boldsymbol{\beta}|\sigma^2 \sim N(\boldsymbol{\mu}_{\boldsymbol{\beta}}, \sigma^2\boldsymbol{\Sigma}_{\boldsymbol{\beta}})$ and $\sigma^2 \sim IG(a, b)$. With data $\boldsymbol{D}^{(t)}$ observed through time t, the marginal posterior density of parameters σ^2 and $\boldsymbol{\beta}$ at time t appear in closed form and are given by $IG(a_t^*, b_t^*)$ and $Multivariate - t_{2a_t^*}(\boldsymbol{\mu}_t^*, (b_t^*/a_t^*)\boldsymbol{V}_t^*)$ respectively, where $\boldsymbol{\mu}_t^* = (\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1} + \sum_{s=1}^t \boldsymbol{X}'_s\boldsymbol{X}_s)^{-1}(\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} + \sum_{s=1}^t \boldsymbol{X}'_s\boldsymbol{y}_s)$, $\boldsymbol{V}_t^* = (\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1} + \sum_{s=1}^t \boldsymbol{X}'_s\boldsymbol{X}_s)^{-1}$, $a_t^* = a + nt/2$, $b_t^* = b + (\boldsymbol{\mu}'_{\boldsymbol{\beta}}\boldsymbol{\Sigma}_{\boldsymbol{\beta}}^{-1}\boldsymbol{\mu}_{\boldsymbol{\beta}} + \sum_{s=1}^t \boldsymbol{y}'_s\boldsymbol{y}_s - \boldsymbol{\mu}_t^{*'}\boldsymbol{V}_t^{*-1}\boldsymbol{\mu}_t^*)/2$. Notably, posterior distributions depend on the data only through the three sufficient statistics $\sum_{s=1}^t \boldsymbol{X}'_s\boldsymbol{X}_s$, $\sum_{s=1}^t \boldsymbol{X}'_s\boldsymbol{y}_s$ and $\sum_{s=1}^t \boldsymbol{y}'_s\boldsymbol{y}_s$. Hence, the posterior distribution at time t with the onset of data $\boldsymbol{D}_t$ can readily be constructed by storing and updating the sufficient statistics without having the need to store the entire data $\boldsymbol{D}^{(t)}$ through time t. When p is large, the major challenge in computing posterior distributions at time t comes from evaluating $\boldsymbol{V}_t^*$ which involves taking the inverse of a $p \times p$ matrix. However, the marginal posterior distribution of $\boldsymbol{\beta}$ being in closed form, operating characteristics of the posteriors are available analytically, bypassing the need to follow an iterative sampling scheme to estimate these operating characteristics.

Such closed form expressions for the marginal posterior distributions of parameters are hard to come by when the focus is on Bayesian high dimensional regularization (shrinkage) or variable selection priors. This chapter considers the Bayesian Lasso and Horseshoe priors as two representative priors from the class of shrinkage priors and the Spike and Lasso prior from the class of variable selection priors. Below we briefly introduce online posterior computation with these priors with large or streaming data and describe computational challenges with large p. The computational challenges are similar in other Bayesian shrinkage or variable selection priors.

2.2.1 Bayesian Lasso Shrinkage Prior

The Bayesian Lasso shrinkage prior stands as an important example of the *global-local (GL) scale mixtures* (Polson and Scott, 2010) of normal prior distributions. The prior takes the specific form $p(\beta_j|\sigma^2,\lambda) = \frac{\lambda}{2\sigma}\exp(-\lambda|\beta_j|/\sigma)$, $j = 1,..,p$, $\lambda^2 \sim G(r,d)$, with the conditional posterior distribution of $\boldsymbol{\beta}$ given other parameters not available in closed form. However, conditional distributions can be obtained in closed form using a data augmentation approach. In fact, the hierarchical data augmented model with the Bayesian Lasso prior on $\boldsymbol{\beta}$ with data $\boldsymbol{D}^{(t)} = \{(\boldsymbol{y}_s, \boldsymbol{X}_s) : s = 1,...,t\}$ up to time t is given by

$$\begin{aligned}
\boldsymbol{y}_s|\boldsymbol{X}_s,\boldsymbol{\beta},\sigma^2 &\sim N_n\left(\boldsymbol{X}_s\boldsymbol{\beta},\sigma^2\boldsymbol{I}_n\right),\ \ s=1,...,t\\
\boldsymbol{\beta}|\boldsymbol{\tau}^2,\sigma^2 &\sim N_p\left(\boldsymbol{0},\sigma^2\boldsymbol{M}_\tau\right),\ \ \pi\left(\sigma^2\right)\propto\frac{1}{\sigma^2},\\
\lambda^2 \sim G(r,d),\ \ \tau_j^2 &\sim Exp\left(\frac{\lambda^2}{2}\right),\ \ j=1,...,p
\end{aligned}$$

where $\tau_1^2,...,\tau_p^2$ are predictor specific latent variables employed for data augmentation, $\boldsymbol{\tau}^2 = (\tau_1^2,...,\tau_p^2)'$ and $\boldsymbol{M}_\tau = diag(\boldsymbol{\tau}^2)$. The batch MCMC implemented using the customary Gibbs sampler alternates between the full conditional distributions of (i) $\boldsymbol{\beta}|\sigma^2,\lambda^2,\boldsymbol{\tau}^2,\boldsymbol{D}^{(t)}$; (ii) $\sigma^2|\boldsymbol{\beta},\lambda^2,\boldsymbol{\tau}^2,\boldsymbol{D}^{(t)}$; (iii) $\lambda^2|\boldsymbol{\beta},\sigma^2,\boldsymbol{\tau}^2,\boldsymbol{D}^{(t)}$ and (iv) $\tau_j^2|\sigma^2,\lambda^2,\boldsymbol{\beta},\boldsymbol{D}^{(t)}$, $j = 1,...,p$, given by

$$\begin{aligned}
\boldsymbol{\beta}|\sigma^2,\boldsymbol{\tau}^2,\lambda^2,\boldsymbol{D}^{(t)} &\sim N_p\left(\left(\boldsymbol{S}_1^{(t)}+\boldsymbol{M}_\tau^{-1}\right)^{-1}\boldsymbol{S}_2^{(t)},\sigma^2\left(\boldsymbol{S}_1^{(t)}+\boldsymbol{M}_\tau^{-1}\right)^{-1}\right)\\
\sigma^2|\boldsymbol{\beta},\boldsymbol{\tau}^2,\lambda^2,\boldsymbol{D}^{(t)} &\sim IG\left(\frac{nt+p}{2},\frac{\left(\boldsymbol{S}_3^{(t)}+\boldsymbol{\beta}'\boldsymbol{S}_1^{(t)}\boldsymbol{\beta}-2\boldsymbol{\beta}'\boldsymbol{S}_2^{(t)}\right)+\boldsymbol{\beta}'\boldsymbol{M}_\tau^{-1}\boldsymbol{\beta}}{2}\right)\\
\frac{1}{\tau_j^2}|\boldsymbol{\beta},\sigma^2,\lambda^2,\boldsymbol{D}^{(t)} &\sim Inv-Gaussian\left(\sqrt{\frac{\lambda^2\sigma^2}{\beta_j^2}},\lambda^2\right)\\
\lambda^2|\boldsymbol{\beta},\sigma^2,\boldsymbol{\tau}^2,\boldsymbol{D}^{(t)} &\sim IG\left(p+r,\frac{\sum_{j=1}^p\tau_j^2}{2}+d\right).
\end{aligned} \tag{2.2}$$

The full conditional posterior distributions at time t depend on the data $\boldsymbol{D}^{(t)}$ only through a few sufficient statistics $\boldsymbol{S}_1^{(t)} = \boldsymbol{S}_1^{(t-1)} + \boldsymbol{X}_t'\boldsymbol{X}_t$, $\boldsymbol{S}_2^{(t)} = \boldsymbol{S}_2^{(t-1)} + \boldsymbol{X}_t'\boldsymbol{y}_t$ and $\boldsymbol{S}_3^{(t)} = \boldsymbol{S}_3^{(t-1)} + \boldsymbol{y}_t'\boldsymbol{y}_t$, which are updated at the onset of a new data shard. At each time $t = 1, ..., T$, the main computational issue lies in the Gibbs sampling step of $\boldsymbol{\beta}$ that requires decomposing a $p \times p$ covariance matrix costing $\sim p^3$ floating point operations (flops) and $\sim p^2$ storage units, and is rendered infeasible.

2.2.2 Horseshoe Shrinkage Prior

We also consider the popularly used Horseshoe (Carvalho et al., 2010) shrinkage prior on high dimensional predictor coefficients, which is well recognized in the Bayesian shrinkage literature for its ability to artfully shrink unimportant predictor coefficients while applying minimum shrinkage on important coefficients. Several recent articles theoretically prove its ability to estimate true predictor coefficients a-posteriori in presence of both high and low sparsity (Armagan et al., 2013).

Similar to the Bayesian Lasso, the Horseshoe shrinkage prior also does not admit a closed form full posterior of $\boldsymbol{\beta}$. Thus, Gibbs sampling is implemented by invoking a data augmentation approach similar to the Bayesian Lasso. The hierarchical data augmented model with the Horseshoe shrinkage prior is given by

$$\begin{gathered}
\boldsymbol{y}_s|\boldsymbol{X}_s, \boldsymbol{\beta}, \sigma^2 \sim N_n\left(\boldsymbol{X}_s\boldsymbol{\beta}, \sigma^2\boldsymbol{I}_n\right), \;\; s = 1, .., t, \\
\boldsymbol{\beta}|\sigma^2, \tau^2, \boldsymbol{\lambda} \sim N_p\left(\boldsymbol{0}, \tau^2\sigma^2\boldsymbol{M}_{\boldsymbol{\lambda}}\right), \;\; \pi(\sigma^2) \propto \frac{1}{\sigma^2}, \;\; \tau^2 \mid \xi \sim \mathcal{IG}\left(\frac{1}{2}, \frac{1}{\xi}\right), \\
\xi \sim \mathcal{IG}\left(\frac{1}{2}, 1\right), \;\; \lambda_j^2 \mid \nu_j \sim \mathcal{IG}\left(\frac{1}{2}, \frac{1}{\nu_j}\right), \;\; \nu_j \sim \mathcal{IG}\left(\frac{1}{2}, 1\right), \;\; j = 1, .., p,
\end{gathered}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)'$, $\boldsymbol{M}_\lambda = diag\left(\lambda_1^2, \ldots, \lambda_p^2\right)$, $\boldsymbol{\lambda} = \left(\lambda_1^2, \ldots, \lambda_p^2\right)'$ and $\boldsymbol{\nu} =$

$(\nu_1, \ldots, \nu_p)'$. The data augmentation allows the batch MCMC procedure to draw MCMC samples at time t from the following full conditional distributions,

$$\begin{aligned}
&\boldsymbol{\beta}|\sigma^2, \boldsymbol{\tau}^2, \lambda^2, \boldsymbol{D}^{(t)} \sim N_p\left(\left(\boldsymbol{S}_1^{(t)} + \frac{\boldsymbol{M}_\lambda^{-1}}{\tau^2}\right)^{-1} \boldsymbol{S}_2^{(t)}, \sigma^2\left(\boldsymbol{S}_1^{(t)} + \frac{\boldsymbol{M}_\lambda^{-1}}{\tau^2}\right)^{-1}\right) \\
&\sigma^2|\boldsymbol{\beta}, \boldsymbol{\tau}^2, \lambda^2, \boldsymbol{D}^{(t)} \sim IG\left(\frac{nt+p}{2}, \frac{\boldsymbol{S}_3^{(t)} + \boldsymbol{\beta}'\boldsymbol{S}_1^{(t)}\boldsymbol{\beta} - 2\boldsymbol{\beta}'\boldsymbol{S}_2^{(t)}}{2} + \frac{\boldsymbol{\beta}'\boldsymbol{M}_\lambda^{-1}\boldsymbol{\beta}}{2\tau^2}\right) \\
&\lambda_j^2|\beta_j, \nu_j, \tau^2, \sigma^2, \boldsymbol{D}^{(t)} \sim IG\left(1, \left[\frac{1}{\nu_j} + \frac{\beta_j^2}{2\tau^2\sigma^2}\right]\right), \ \nu_j|\lambda_j^2, \boldsymbol{D}^{(t)} \sim IG\left(1, \left(1 + \frac{1}{\lambda_j^2}\right)\right) \\
&\xi|\boldsymbol{\beta}, \sigma^2, \boldsymbol{\tau}^2, \boldsymbol{D}^{(t)} \sim IG\left(1, 1 + \frac{1}{\tau^2}\right), \tau^2|\boldsymbol{\beta}, \boldsymbol{\lambda}, \sigma^2, \boldsymbol{D}^{(t)} \sim IG\left(\frac{p+1}{2}, \frac{1}{\xi} + \frac{\boldsymbol{\beta}'\boldsymbol{M}_\lambda^{-1}\boldsymbol{\beta}}{2\sigma^2}\right).
\end{aligned} \tag{2.3}$$

The conditional distributions are dependent on the data $\boldsymbol{D}^{(t)}$ only through sufficient statistics $\boldsymbol{S}^{(t)} = \{\boldsymbol{S}_1^{(t)}, \boldsymbol{S}_2^{(t)}, \boldsymbol{S}_3^{(t)}\}$ which are updated using $\boldsymbol{S}_1^{(t)} = \boldsymbol{S}_1^{(t-1)} + \boldsymbol{X}_t'\boldsymbol{X}_t$, $\boldsymbol{S}_2^{(t)} = \boldsymbol{S}_2^{(t-1)} + \boldsymbol{X}_t'\boldsymbol{y}_t$ and $\boldsymbol{S}_3^{(t)} = \boldsymbol{S}_3^{(t-1)} + \boldsymbol{y}_t'\boldsymbol{y}_t$. Similar to the Bayesian Lasso, the Gibbs sampling step of $\boldsymbol{\beta}$ involves decomposing and storing a $p \times p$ matrix per iteration that becomes costly with big p.

2.2.3 Spike and Lasso Variable Selection Prior

Although shrinkage priors are designed to shrink the posterior distributions of unimportant predictor coefficients close to zero, the shrinkage frameworks do not allow detection of unimportant predictors. In contrast, the spike and slab discrete mixture of distributions are specifically designed for variable selection in high dimensional regressions (George and McCulloch, 1997). In this section, a

variant of the spike and slab mixture prior is introduced as,

$$\beta_j|\sigma^2, \tau_j^2, \gamma_j \sim \gamma_j N\left(0, \sigma^2\tau_j^2\right) + (1-\gamma_j)N\left(0, \sigma^2 c^2\right)$$
$$\tau_j^2 \sim Exp(\lambda^2/2),\ \gamma_j \sim Ber(\theta),\ \lambda^2 \sim Ga(r,d),\ \theta \sim Beta(a,b).$$

Integrating over the latent variables τ_j^2, we obtain $\beta_j|\sigma^2, \lambda^2, \gamma_j \sim \gamma_j DE(\lambda/\sigma) + (1-\gamma_j)N(0, \sigma^2 c^2)$, for $j = 1, ..., p$, as a mixture of a double exponential and normal densities. We refer to this mixture distribution as the *Spike and Lasso* distribution. Choosing c^2 small, the prior performs simultaneous variable selection and parameter estimation, adaptively thresholding small effects with the concentrated normal spike while minimally shrinking the large effects with the heavy-tailed double exponential (DE) slab distribution. Allowing the prior inclusion probability θ to be random enables us to automatically adjust for multiple comparisons (Scott and Berger, 2010). Spike and slab discrete mixture priors enjoy attractive theoretical properties (Castillo et al., 2015) and a transformed spike and slab prior has recently been added as a penalty to the frequentist penalized optimization literature (Ročková and George, 2018).

With data up to time t, $\boldsymbol{D}^{(t)}$ and sufficient statistics $\boldsymbol{S}_1^{(t)}$, $\boldsymbol{S}_2^{(t)}$ and $\boldsymbol{S}_3^{(t)}$, the prior formulation and data model lead to the following closed form full conditional

posteriors facilitating implementation with a Gibbs sampler:

$$
\begin{aligned}
&\boldsymbol{\beta}|\sigma^2,\tau^2,\gamma,\boldsymbol{D}^{(t)} \sim N_p\left(\left(\boldsymbol{S}_1^{(t)}+\boldsymbol{M}^{-1}\right)^{-1}\boldsymbol{S}_2^{(t)},\sigma^2\left(\boldsymbol{S}_1^{(t)}+\boldsymbol{M}^{-1}\right)^{-1}\right)\\
&\sigma^2|\boldsymbol{\beta},\boldsymbol{\tau}^2,\lambda^2,\boldsymbol{D}^{(t)} \sim IG\left(\frac{nt+p}{2},\frac{\left(\boldsymbol{S}_3^{(t)}+\boldsymbol{\beta}'\boldsymbol{S}_1^{(t)}\boldsymbol{\beta}-2\boldsymbol{\beta}'\boldsymbol{S}_2^{(t)}\right)+\boldsymbol{\beta}'\boldsymbol{M}^{-1}\boldsymbol{\beta}}{2}\right)\\
&\lambda^2|\boldsymbol{\beta},\sigma^2,\tau^2,\boldsymbol{D}^{(t)} \sim IG\left(p+r,\frac{\sum_{j=1}^p\gamma_j\tau_j^2}{2}+d\right),\\
&\theta \sim Beta\left(a+\sum_{j=1}^p\gamma_j, b+p-\sum_{j=1}^p\gamma_j\right)\\
&\frac{1}{\tau_j^2}|\gamma_j=1,\boldsymbol{\beta},\sigma^2,\lambda^2,\boldsymbol{D}^{(t)} \sim Inv-Gaussian\left(\sqrt{\frac{\lambda^2\sigma^2}{\beta_j^2}},\lambda^2\right)\\
&\tau_j^2|\gamma_j=0,\boldsymbol{\beta},\sigma^2,\lambda^2,\boldsymbol{D}^{(t)} \sim Exp(\lambda^2/2),\ \ \gamma_j|\boldsymbol{\beta},\sigma^2,\tau^2,\theta,\boldsymbol{D}^{(t)} \sim Ber(\eta_j)\\
&\eta_j=\frac{\theta\left(\sigma^2\tau_j^2\right)^{-\frac{1}{2}}\exp\left(-\frac{\beta_j^2}{2\sigma^2\tau_j^2}\right)}{\theta\left(\sigma^2\tau_j^2\right)^{-\frac{1}{2}}\exp\left(-\frac{\beta_j^2}{2\sigma^2\tau_j^2}\right)+(1-\theta)\left(c^2\right)^{-\frac{1}{2}}\exp\left(-\frac{\beta_j^2}{2c^2}\right)}.
\end{aligned}
\tag{2.4}
$$

where $\boldsymbol{M}=diag(w_1,\ldots,w_p)$ with $w_j=\tau_j^2$ if $\gamma_j=1$; $w_j=c^2$ otherwise.

The computational issue arises from the Gibbs sampling step of $\boldsymbol{\beta}$ that incurs a complexity of $O(p^3)$, as well as due to updating γ_j's, $j=1,...,p$ resulting in high auto-correlation. Updating subsets of $\boldsymbol{\beta}$ parameters in smaller blocks may be an option. However, shrinkage or variable selection priors generally do not allow closed form marginal distributions for such blocks of regression parameters. Again, the sequential nature of Gibbs sampling prohibits updating blocks of parameter $\boldsymbol{\beta}$ in parallel. The dynamic feature partitioning strategy developed in the next section will provide a solution to this computational challenge by parallelizing the approximate Bayesian computation of blocks of parameters into different processors.

2.3 Dynamic Feature Partition in High-Dimensional Regression

The dynamic feature partitioning (DFP) is a general online algorithm for streaming data that partitions the large parameter set into mutually exclusive and exhaustive subsets and facilitates rapid Bayesian updating of different parameter subsets in parallel. While the algorithm is applied to mitigate the aforementioned computational issues in the Bayesian high dimensional linear regression, the algorithm per se is more general in nature and could be implemented beyond high dimensional linear regressions.

2.3.1 Relevant Notations and Details of DFP

Let $\boldsymbol{\Theta} = \{\theta_1, \ldots, \theta_q\}$ represent the parameter space with q parameters, which is bigger than p (the number of predictors), since the parameter space includes the error variance σ^2 as well as latent variables from the data augmentation procedures described in Section 2.2. We further assume

(1) q is fixed over time, i.e., the parameter space does not change with the arrival of new data shards.

(2) At each time point, the posterior distribution of the parameters $\boldsymbol{\Theta}$ depends on the data only through lower dimensional functions of $\boldsymbol{D}^{(t)}$ which are referred to as sufficient statistics. More formally, $\boldsymbol{S}^{(t)}$ is a vector of sufficient statistics for $\boldsymbol{\Theta}$ if $\boldsymbol{\Theta}|\boldsymbol{D}^{(t)}$ has the same distribution as $\boldsymbol{\Theta}|\boldsymbol{S}^{(t)}$. Denoting $f(\boldsymbol{\Theta}|\boldsymbol{D}^{(t)})$ as the full posterior distribution of $\boldsymbol{\Theta}$, this assumption implies that $f(\boldsymbol{\Theta}|\boldsymbol{D}^{(t)}) = f(\boldsymbol{\Theta}|\boldsymbol{S}^{(t)})$.

Referring to Section 2.2, both (1) and (2) are valid for linear regression models

with shrinkage prior distributions or discrete mixture variable selection priors on coefficients.

At time t, consider a partition of the parameter indices given by $\mathcal{G}^{(t)} = \{G_1^t, \ldots, G_{k_t}^t\}$, such that $G_l^t \cap G_{l'}^t = \emptyset, l \neq l'$ and $\bigcup_{l=1}^{k_t} G_l^t = \{1, \ldots, q\}$. Also let $\boldsymbol{\Theta}_{G_l^t} = \{\theta_i \mid i \in G_l^t\}$ and $\boldsymbol{\Theta}_{-G_l^t} = \boldsymbol{\Theta}_{\{1,\ldots,q\}\setminus G_l^t} = \{\theta_i \mid i \in \{1, \ldots, q\} \setminus G_l^t\} = \{\theta_i \mid i \notin G_l^t\}$ be parameters contained and not contained in the lth partition, respectively. We consider both the number of partitions k_t and the constitution of each partition to be adaptive and dynamically changing over time. The prior specifications and conditional independence assumptions often suggest natural parameter partitioning schemes. We provide an outline of the dynamic parameter partitioning schemes employed in this chapter in the context of high dimensional regressions with shrinkage and Spike and Lasso priors towards the end of this section.

Consider also a sequence of point estimates $\widehat{\boldsymbol{\Theta}}^{(t)}$ constructed dynamically over time for the parameter $\boldsymbol{\Theta}$. Given a partition of the parameter space at time t, the DFP approximation to the posterior full conditional distribution $f\left(\boldsymbol{\Theta}_{G_l^t} | \boldsymbol{\Theta}_{-G_l^t}, \boldsymbol{S}^{(t)}\right)$ of $\boldsymbol{\Theta}_{G_l^t}$ $(l = 1, ..., k_t)$, referred to as the *DFP pseudo conditional posterior*, is given by $f\left(\boldsymbol{\Theta}_{G_l^t} | \widehat{\boldsymbol{\Theta}}_{-G_l^t}^{(t-1)}, \boldsymbol{S}^{(t)}\right)$, with $\boldsymbol{\Theta}_{-G_l^t}$ replaced by its point estimate $\widehat{\boldsymbol{\Theta}}_{-G_l^t}^{(t-1)}$ at time $(t-1)$. Since the conditioning set remains fixed throughout time t, conditional distributions $\boldsymbol{\Theta}_{G_l^t}$'s for $l = 1, ..., k_t$ are not dependent on each other at time t. This eliminates the need to sequentially update parameter blocks $\boldsymbol{\Theta}_{G_l^t}$'s, and samples can rather be drawn rapidly from k_t DFP pseudo conditional posteriors in parallel. All these concepts and notations will be used to describe the DFP algorithm below.

2.3.2 DFP Algorithm for Online Approximate MCMC Inference

The DFP algorithm provides an online approximate MCMC sampling based on dynamically adaptive parameter partitions and their point estimates constructed sequentially over time. The algorithm begins by initializing the point estimate of $\boldsymbol{\Theta}$ (call it $\widehat{\boldsymbol{\Theta}}^{(0)}$) at some default value and initializing sufficient statistics $\boldsymbol{S}^{(0)}$ at $\mathbf{0}$. When new data shard $\boldsymbol{D}_t$ arrives at time t ($t = 1, ..., T$), sufficient statistics $\boldsymbol{S}^{(t)}$ are updated as a function of $\boldsymbol{S}^{(t-1)}$ and $\boldsymbol{D}_t$, denoted as $\boldsymbol{S}^{(t)} = g(\boldsymbol{S}^{(t-1)}, \boldsymbol{D}_t)$. In the examples of Section 2.2, $g(\cdot)$ is implicitly defined through the three equations, $\boldsymbol{S}_1^{(t)} = \boldsymbol{S}_1^{(t-1)} + \boldsymbol{X}_t'\boldsymbol{X}_t$, $\boldsymbol{S}_2^{(t)} = \boldsymbol{S}_2^{(t-1)} + \boldsymbol{X}_t'\boldsymbol{y}_t$ and $\boldsymbol{S}_3^{(t)} = \boldsymbol{S}_3^{(t-1)} + \boldsymbol{y}_t'\boldsymbol{y}_t$. The dynamic partitioning scheme (described later) then updates partitions of the set of parameters and creates new partitions $\mathcal{G}^{(t)}$ at time t. The DFP algorithm then proceeds by sampling from the DFP pseudo conditional posteriors at time t in parallel. If the DFP pseudo conditional posteriors are in closed form, one may consider block updating of $\boldsymbol{\Theta}_{G_l^t}$ from $f\left(\boldsymbol{\Theta}_{G_l^t}|\widehat{\boldsymbol{\Theta}}_{-G_l^t}^{(t-1)}, \boldsymbol{S}^{(t)}\right)$. Otherwise, the sampling in each partition proceeds by employing a Gibbs sampler with smaller blocks of parameters in the lth partition. More specifically, $\theta_j \in \boldsymbol{\Theta}_{G_l^t}$ is updated by drawing S (a moderately large number, taken to be 500 in Section 2.4) approximate MCMC samples $\tilde{\theta}_j^{(1,t)}, ..., \tilde{\theta}_j^{(S,t)}$ from $f\left(\theta_j|\boldsymbol{\Theta}_{G_l^t\setminus\{j\}}, \widehat{\boldsymbol{\Theta}}_{-G_l^t}^{(t-1)}, \boldsymbol{S}^{(t)}\right)$, where the tilde emphasises the fact that we are sampling from an approximation to the full conditional distribution, instead of the full conditional distribution. Often this distribution depends on a lower dimensional function of $\boldsymbol{\Theta}_{G_l^t\setminus\{j\}}$, $\widehat{\boldsymbol{\Theta}}_{-G_l^t}^{(t-1)}$ and $\boldsymbol{S}^{(t)}$, as we will see in Sections 2.4.1, 2.4.2 and 2.4.3. Once S approximate MCMC samples are drawn from DFP pseudo conditional posteriors fairly rapidly, we use these samples to construct the point estimates of parameters at time t, given by $\widehat{\boldsymbol{\Theta}}^{(t)}$. In our exposition, we use the mean of the S samples $\tilde{\theta}_j^{(1,t)}, ..., \tilde{\theta}_j^{(S,t)}$ to construct $\widehat{\theta}_j^{(t)}$.

The theoretical results in appendix A prove desirable performance of the proposed algorithm when the sequence of estimators $\widehat{\boldsymbol{\Theta}}^{(t)}$ is consistent in estimating the true parameters as $t \to \infty$. In practice, we found this assumption can be validated empirically for implementation of DFP in Sections 2.4.1, 2.4.2 and 2.4.3. In fact, the trace-plots of $\widehat{\boldsymbol{\Theta}}^{(t)}$ corresponding to representative regression parameters in Section 2.4 show convergence around the true data generating parameters. Efficient updating of DFP pseudo conditional posteriors using the sufficient statistics and point estimates of parameters from the previous time point lead to scalable inference.

Partitioning schemes:

As discussed before, an efficient partitioning of parameter indices $\mathcal{G}^{(t)}$ at the tth time is achieved by heavily exploiting the nature of the model and prior distributions. We believe that a general partitioning scheme that is applicable to any model and/or any prior distribution is unappealing since it will not be able to fully exploit the specific features of the model and prior distributions. Since the main focus of this chapter is on Bayesian shrinkage and variable selection priors in high dimensional linear regression models, broadly two different partitioning schemes are proposed, one for the model (2.1) with shrinkage priors and the other for spike and slab priors.

(A) Partitioning algorithm for shrinkage priors:

Referring to the discussion in Sections 2.2.1 and 2.2.2, the computational bottleneck mainly arises due to sampling from the posterior full conditional of $\boldsymbol{\beta}$. Therefore, in the course of developing a partitioning strategy for the set of parameters in (2.1) with shrinkage priors, the main focus rests on how to partition $\boldsymbol{\beta}$ into blocks of sub-vectors with a minimal loss of information due to separately updating these blocks residing in different subset partitions from their DFP full

Algorithm 1: Dynamic Feature Partition

Input: (1) Data shard $\boldsymbol{D}_t$ at time t; (2) Parameter partition $\mathcal{G}^{(t-1)}$; (3) Sufficient Statistics $\boldsymbol{S}^{(t-1)}$ (4) Approximate posterior draws $\tilde{\boldsymbol{\Theta}}^{(1,t-1)}, \ldots, \tilde{\boldsymbol{\Theta}}^{(S,t-1)}$ at time $(t-1)$; (5) Parameter Estimates $\widehat{\boldsymbol{\Theta}}^{(t-1)}$

Output: (1) Approximate posterior draws $\tilde{\boldsymbol{\Theta}}^{(1,t)}, \ldots, \tilde{\boldsymbol{\Theta}}^{(S,t)}$ at time t; (2) Sufficient Statistics $\boldsymbol{S}^{(t)}$; (3) Parameter Estimates $\widehat{\boldsymbol{\Theta}}^{(t)}$

DFP$(\boldsymbol{D}_t, \mathcal{G}^{(t)}, \boldsymbol{S}^{(t-1)}, \widehat{\boldsymbol{\Theta}}^{(t-1)})$
begin
  `/* Step 1: Update the partition of the set of parameters at time` t`: the partitioning schemes should ideally exploit the nature of the model and prior distributions. We propose partitioning schemes specific to the high dimensional linear regression with shrinkage priors and spike and slab priors in Section 2.3.2. */`
  $\mathcal{G}^{(t)} = \textbf{PartitionUpdate}\left(\tilde{\boldsymbol{\Theta}}^{(1,t-1)}, \ldots, \tilde{\boldsymbol{\Theta}}^{(S,t-1)}\right)$
  `/* step 2: Update Sufficient Statistics */`
  Update $\boldsymbol{S}^{(t)} = g(\boldsymbol{D}_t, \boldsymbol{S}^{(t-1)})$
  `/* step 3: Approximate Sampling for Parameter Blocks in Parallel */`
  for $G_l^t \in \mathcal{G}^{(t)}$ **do**
    for $\theta_j \in \boldsymbol{\Theta}_{G_l^t}$ **do**
      for *s=1:S* **do**
        sample $\tilde{\theta}_j^{(s,t)} \sim f\left(\theta_j | \boldsymbol{\Theta}_{G_l^t \setminus \{j\}}, \boldsymbol{S}^{(t-1)}, \widehat{\boldsymbol{\Theta}}_{-G_l^t}^{(t-1)}\right)$
      end
    end
  end
  `/* step 4: Update Estimates */`
  for $G_l^t \in \mathcal{G}^{(t)}$ **do**
    for $\theta_j \in \boldsymbol{\Theta}_{G_l^t}$ **do**
      `/* Compute relevant point estimates for the parameters from approximate MCMC samples. We consider the mean of the samples as the point estimate for each parameter */`
      set $\widehat{\theta}_j^{(t)} \leftarrow \text{stat}\left(\tilde{\theta}_j^{(1,t)}, \ldots, \tilde{\theta}_j^{(S,t)}\right)$
    end
  end
  return $\{\tilde{\boldsymbol{\Theta}}^{(1,t)}, \ldots, \tilde{\boldsymbol{\Theta}}^{(S,t)}\}, \boldsymbol{S}^{(t)}, \widehat{\boldsymbol{\Theta}}^{(t)}$
end

conditionals. To this end, we set the maximum size of each block of $\boldsymbol{\beta}$ residing in different partitions to be less than or equal to M at every time to keep a control on the computational complexity. M is user defined and its choice depends on the available computational resources. In our empirical investigations with high dimensional linear regression with Bayesian shrinkage priors, we find $M = 100$ to be sufficient and provide discussion on how the choice of small values of M affects inference. Thereafter we envision the problem of partitioning $\boldsymbol{\beta}$ at time t as a graph partitioning problem. To elaborate, at time t, for $j, j' \in \{1, ..., p\}$, let the sample correlation between S iterates of β_j and $\beta_{j'}$ from time $(t-1)$ following the DFP algorithm, given by $\{\tilde{\beta}_j^{(s,t-1)}\}_{s=1}^S$ and $\{\tilde{\beta}_{j'}^{(s,t-1)}\}_{s=1}^S$, be denoted by $r_{j,j'}$. A graph is constructed with nodes as the predictor indices $\{1, ..., p\}$ and an edge between two nodes j, j' if $r_{j,j'} > c$ where $c \in (0, 1)$. Our proposed scheme constructs different graphs in this manner corresponding to different choices of the cut-off $c \in$ `seq(0.01,0.99,by=0.01)`. Thereafter we find connected components of all these constructed graphs and look for the smallest value of c (say c^*) for which the size of all connected components are less than M. Such an implementation is readily achieved by the functionalities in the `igraph` package in `R`. Let there be b_t connected components corresponding to the cut-off value c^* at time t, which we denote by $\{\mathcal{P}_1^{(t)}, ..., \mathcal{P}_{b_t}^{(t)}\}$. These b_t connected components at time t are recognized as partitions of the indices $\{1, ..., p\}$ and β_j's corresponding to different connected components go to different partitions of the parameter sets at time t. Thus, $\boldsymbol{\beta}_{\mathcal{P}_1^{(t)}}, ..., \boldsymbol{\beta}_{\mathcal{P}_{b_t}^{(t)}}$ go to different subsets in the implementation of DFP at time t. Since the data augmentation approaches in Sections 2.2.1 and 2.2.2 introduce latent vectors ($\boldsymbol{\tau}^2$ in Section 2.2.1, $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ in Section 2.2.2) related to $\boldsymbol{\beta}$, we either keep all elements of a latent vector together in one partition or divide a latent vector into blocks with indices $\{\mathcal{P}_1^{(t)}, ..., \mathcal{P}_{b_t}^{(t)}\}$ and send the latent vector with

indices $\mathcal{P}_k^{(t)}$ to the same parameter subset where $\boldsymbol{\beta}_{\mathcal{P}_k^{(t)}}$ lies. Variance σ^2 and other hierarchical parameters are kept together in a separate partition. Since a partition involves blocks of $\boldsymbol{\beta}$ with size at most M, sampling them together from their DFP full conditionals incurs complexity at most of $O(M^3)$. We later empirically establish that the subsets of parameters constructed by the above partitioning scheme stabilize over time. In fact, our empirical analysis also demonstrates that the optimal value c^* also stabilizes as time progresses.

(B) Partitioning algorithm for Spike and Lasso priors: Since the Spike and Lasso example in Section 2.2.3 involves coefficients belonging to one of the two mixture components at every iteration of the posterior sampling, the parameter partitioning scheme adopted for shrinkage priors appears to be less efficient here. Instead, we propose a dynamic partitioning scheme of the parameter space by tacitly exploiting the natural partitioning of the $\boldsymbol{\beta}$ parameters and associated latent vector $\boldsymbol{\tau}$ into important and unimportant components. Define $\boldsymbol{\Theta}_{1t} = \{(\beta_j, \tau_j^2) : \widehat{\gamma}_j^{(t-1)} = 1\}$ and $\boldsymbol{\Theta}_{2t} = \{(\beta_j, \tau_j^2) : \widehat{\gamma}_j^{(t-1)} = 0\}$, where $\widehat{\gamma}_j^{(t-1)} \in \{0, 1\}$ corresponds to the point estimate of γ_j at time $(t-1)$. Thereafter our partitioning scheme suggests keeping the entire $\boldsymbol{\Theta}_{1t}$ in one partition and dividing $\boldsymbol{\Theta}_{2t}$ into subsets, with each subset of $\boldsymbol{\Theta}_{2t}$ containing (β_j, τ_j^2) for a single j. Additionally, all γ_j's are kept in the same partition and $\lambda^2, \sigma^2, \theta$ in another partition. Since spike and slab priors are typically employed to recover $\boldsymbol{\beta}$ parameters which are sparse in nature in the truth, $\boldsymbol{\Theta}_{1t}$ is expected to be of small to moderate size with cardinality much smaller than p as time progresses. Thus, updating $(\beta_j : \beta_j \in \boldsymbol{\Theta}_{1t})'$ together requires computational complexity of order $|\boldsymbol{\Theta}_{1t}|^3 << p^3$. On the other hand, β_j's for $j \in \boldsymbol{\Theta}_{2t}$ are updated individually without incurring any notable computational burden. A similar strategy is followed when the double exponential slab distribution in the Spike and Lasso prior is replaced by any other distribution.

2.4 Illustrations of DFP with Shrinkage and Discrete Mixture Priors in High-Dimensional Regressions

This section illustrates parametric and predictive performances of the online DFP algorithm for (i) Bayesian Lasso, (ii) Horseshoe and (iii) Spike and Lasso discrete mixture priors. For the simulation examples in (i)-(iii), shards of size $n = 1000$ observations arrive sequentially over $T = 500$ time horizons. Data shard $\boldsymbol{D}_t$ at time t consists of an $n \times 1$ response vector $\boldsymbol{y}_t$ and an $n \times p$ predictor matrix $\boldsymbol{X}_t = (\boldsymbol{x}_{1t}, ..., \boldsymbol{x}_{nt})'$, $t = 1, ..., T$. At each time, $S = 500$ approximate MCMC samples of $\boldsymbol{\Theta}_{G_1^t}, ..., \boldsymbol{\Theta}_{G_{k_t}^t}$ are drawn from their respective DFP pseudo conditional posteriors to approximate the full posterior distribution $f(\boldsymbol{\Theta}|\boldsymbol{D}^{(t)})$.

The $p \times 1$ predictor vector $\boldsymbol{x}_{jt}$ $(j = 1, ..., n)$ at time t is generated as $\boldsymbol{x}_{jt} \sim N(\boldsymbol{0}, \boldsymbol{H})$, where $\boldsymbol{H} = \text{Block-diag}(\boldsymbol{H}_1, ..., \boldsymbol{H}_{100})$, with each $\boldsymbol{H}_l$ being a 50×50 Toeplitz structured matrix having the (m, m')th element as $\rho^{|m-m'|}$, $\rho \in (0, 1)$. This is to mimic the scenario where there are blocks of predictors such that predictors within a block are correlated and predictors across blocks are uncorrelated. All simulation examples consider high correlations among predictors in a block with $\rho = 0.9$. This presumably induces strong associations among parameters, which is often challenging for any high dimensional regression framework to estimate. The inferential challenge appears to be more critical for the DFP framework as it relies on parameter partitioning, which might naturally weaken correlations a-posteriori among parameters. To simulate the true predictor coefficients $\boldsymbol{\beta} = (\beta_1, ..., \beta_p)'$, the following scenarios are considered:

Simulation 1: 50 randomly selected β_j's are drawn i.i.d. from N(3,1), 50 randomly selected β_j's are drawn i.i.d. from N(1,1), rest are all set to 0.

Simulation 2: 50 randomly selected β_j's are drawn i.i.d. from N(3,1), rest are all set to 0.

Simulation 3: All β_j's are drawn i.i.d. from $U(-1, 1)$.

Simulation 1 focuses on a sparse case with varying magnitudes of nonzero coefficients. We will refer to it as the *low and high sparse case*. Simulation 2 corresponds to a *sparse case* with similar magnitudes of nonzero coefficients, while Simulation 3 corresponds to a *dense case* which is motivated by practical applications where each of the covariates has a small effect on the outcome. The responses $\boldsymbol{y}_t$ for $t = 1, ..., T$ are generated from $\boldsymbol{X}_t$ and the true predictor coefficients using (2.1), with σ^2 chosen so as to keep a signal to noise ratio of 1 for the generated data.

Competitors. The performance of DFP is compared with a set of competitors suitable for high dimensional linear regression models. We specifically compare with (a) batch MCMC that draws S MCMC samples from the full conditional distributions at every time point with the full data $\boldsymbol{D}^{(t)}$ through time t at disposal; and (b) Conditional Density Filtering (CDF) (Guhaniyogi et al., 2013). Batch MCMC offers the "gold standard" for ordinary Gibbs sampling that uses the full data $\boldsymbol{D}^{(t)}$ at time t. For batch MCMC, we randomly fix the partition for the coefficients at the beginning to $\boldsymbol{\Theta}_{G_1}, ..., \boldsymbol{\Theta}_{G_k}$ and sequentially draw S MCMC samples from the conditional distribution of each block of parameters given the rest, $\boldsymbol{\Theta}_{G_l}|\boldsymbol{\Theta}_{-G_l}$, at each time. In our implementation, each partition of the coefficients is kept at an equal size of 50. At time t, batch MCMC initializes the MCMC chain at the last iterate in time $(t - 1)$. In examples (i)-(iii), the conditional posterior distributions depend on the data through lower dimensional sufficient statistics, and hence batch MCMC only stores and propagates the sufficient statistics to update the conditional distributions in successive time points. Conditional density filtering is proposed in the same vein as DFP and follows the same algorithm

outlined in Algorithm 1 with an important difference. While DFP partitioning of the set of parameters is dynamic in DFP, CDF works with parameter partitions fixed over time. To keep things consistent, we use the same partition for CDF that we use for batch MCMC. While DFP and CDF both allow drawing samples from approximate conditional posterior distributions for each partition in parallel, we find that implementation of CDF in this way demonstrates considerably inferior performance than DFP. To make CDF more competitive, we employ a version of CDF that draws samples from parameter partitions sequentially rather than in parallel, to be able to use samples from one partition to construct more accurate point estimates for the other partitions at every time. Such an implementation of CDF considerably improves its performance, though at the expense of added computational burden. Overall, comparison with this improved version of CDF will demonstrate the advantages of dynamic partitioning over fixed partitioning as a tool to provide a better approximation to the full posterior distribution of parameters. Online variational inference provides an alternate strategy to draw approximate inference in presence of big data and a large number of parameters. However, in absence of any open source code for online variational inference in high dimensional linear regression, we refrain from employing it as a competitor. Finally, we compare our approach with a variant of the Sequential Monte Carlo (SMC) approach. As discussed in the introduction, most of the developments in SMC and PL algorithms have taken place in the high-dimensional state-space models and they do not assume seamless extensions to high dimensional static parametric models with p as high as 5000.

Therefore, we adapt the recent sub-sampled SMC approach outlined in Gunawan et al. (2018) to our setting. Note that the approach in Gunawan et al. (2018) is designed for the scenario when the entire dataset is available to the user.

To adapt it to the streaming data context, we employ a data annealing approach instead of the temperature annealing approach used by the authors. Our data annealing approach performs data sub-sampling from the entire data $\boldsymbol{D}^{(t)}$ when a new batch arrives at time t and uses the sub-sampling density approximation as well as the Hamiltonian Monte-Carlo technique for efficient drawing of high dimensional Monte Carlo samples. This approach uses the entire data set (up to time t) $\boldsymbol{D}^{(t)}$ in drawing SMC samples at time t, and strictly speaking is not an online Bayesian competitor. Nevertheless, it can demonstrate the state-of-the-art performance from SMC which will be helpful in assessing the performance of DFP. We refer to this approach as sub-sampled SMC (SSMC).

Assessing parametric inference with DFP. Parametric inference with DFP is demonstrated using plots of kernel density estimates for marginal approximate DFP posterior densities of representative model parameters shown at various time points. Kernel density estimates for the batch MCMC at the same time points are also overlaid to assess quality of parametric inference with the DFP approximation in comparison with the "gold standard." The true value of the respective parameters are overlaid to assess the point estimation of parameters from DFP. Additionally trace-plots of $\widehat{\theta}_j^{(t)}$ over time t for representative parameters are also presented to provide evidence of convergence of $\widehat{\boldsymbol{\Theta}}^{(t)}$ to the true parameter as time progresses.

Assessing predictive inference with DFP and competitors. To measure the predictive performance of competitors, we report: (a1) mean squared prediction error (MSPE); (a2) Interval score (Gneiting and Raftery, 2007) of the 95% predictive interval; (a3) coverage of the 95% predictive interval and (a4) average run time for each batch or shard. Note that (a1) demonstrates the performance in terms of point prediction, while (a2) and (a3) show how well calibrated the pre-

dictions turn out to be. Finally, (a4) helps readers gauge the computation time vis-a-vis accuracy of the competitors. At time $(t-1)$, evaluations of predictive performance metrics (a1)-(a3) are based on the data shard observed at time t. All results are based on averages over 10 independent replications. All computation times are based on an R implementation in a cluster computing environment with three interactive analysis servers, 32 cores each with the Dell PE R820: 4x Intel Xeon Sandy Bridge E5-4640 processor, 16GB RAM and 1TB SATA hard drive.

Assessing dynamic partitions of the set of parameters over time. For the strategies implemented to dynamically construct subsets in high dimensional regression with either shrinkage priors or variable selection priors, we monitor the stability of subsets as time progresses. To this end, we evaluate the Adjusted Rand Index (ARI) (Hubert and Arabie, 1985) between partitions of parameters corresponding to two successive time points and plot the ARI over time. The ARI evaluates the agreement in subset assignment between two subsetting/partitioning configurations and is corrected for chance. It ranges between -1 and 1, with larger values indicating agreement between partitioning configurations. Thus, the ARI should converge around 1 as time progresses if the partitions stabilize over time. For the partitioning algorithm implemented for shrinkage priors, we additionally check trace-plot for the optimal value c^* over time and offer an understanding of the sensitivity of inference to the choice of M. In order to being not repetitive, we present trace-plot of c^* or sensitivity to the choice of M only for the Bayesian Lasso prior. The conclusions are similar for the Horseshoe prior.

2.4.1 DFP with Bayesian Lasso

We consider the first application of DFP with the popular Bayesian Lasso (Park and Casella, 2008a) shrinkage prior on high dimensional predictor coef-

ficients. Details of the Bayesian Lasso prior and challenges regarding posterior computation with the Bayesian Lasso prior has already been presented in Section 2.2.1.

The DFP algorithm applied to this setting proposes dynamic partitioning of the parameter space over $k_t = b_t + 1$ subsets at time t. Let the partition of the parameter space at time t be defined by

$$\boldsymbol{\Theta}_{G_l^{(t)}} = \left\{\beta_{i^{(t)}_{m_1+\cdots+m_{l-1}+1}}, \tau^2_{i^{(t)}_{m_1+\cdots+m_{l-1}+1}}, ..., beta_{i^{(t)}_{m_1+\cdots+m_l}}, \tau^2_{i^{(t)}_{m_1+\cdots+m_l}}\right\},\ l = 1, .., b_t,$$
$$\boldsymbol{\Theta}_{G^{(t)}_{b_t+1}} = \left\{\sigma^2, \lambda^2\right\},$$

where the lth partition, $l = 1, .., b_t$ consists of $2m_l$ parameters (m_l is also a function of t) and $i^{(t)}_{m_1+\cdots+m_{l-1}+1}, ..., i^{(t)}_{m_1+\cdots+m_l} \in \{1, ..., p\}$ correspond to the indices of predictor coefficients and latent variables belonging to the lth partition at time t. Let at time t, $\boldsymbol{\beta}_l = \left(\beta_{i^{(t)}_{m_1+\cdots+m_{l-1}+1}}, ..., \beta_{i^{(t)}_{m_1+\cdots+m_l}}\right)'$, $\boldsymbol{\tau}^2_l = \left(\tau^2_{i^{(t)}_{m_1+\cdots+m_{l-1}+1}}, ..., \tau^2_{i^{(t)}_{m_1+\cdots+m_l}}\right)'$, $\boldsymbol{M}_{\tau,l} = diag(\boldsymbol{\tau}^2_l)$ and $\boldsymbol{\beta}_{-l}$ be the vector of all β_j's except those included in $\boldsymbol{\beta}_l$. $\widehat{\boldsymbol{\beta}}^{(t-1)}_l$, $\widehat{\boldsymbol{\beta}}^{(t-1)}_{-l}$, $\widehat{\boldsymbol{\tau}}^{2(t-1)}_l$ are the point estimates of $\boldsymbol{\beta}_l, \boldsymbol{\beta}_{-l}, \boldsymbol{\tau}^2_l$ respectively at time $(t-1)$. $\boldsymbol{S}^{(t)}_{1,l}$ and $\boldsymbol{S}^{(t)}_{2,l}$ are analogously defined. Also assume $\boldsymbol{S}^{(t)}_{1,l,-l} = \boldsymbol{S}^{(t-1)}_{1,l,-l} + \boldsymbol{X}'_{t,l}\boldsymbol{X}_{t,-l}$, where $\boldsymbol{X}_{t,l}$ and $\boldsymbol{X}_{t,-l}$ are the sub-matrices of $\boldsymbol{X}_t$ corresponding to $\boldsymbol{\beta}_l$ and $\boldsymbol{\beta}_{-l}$, respectively. Section A.2.1 of appendix A describes details of implementing of Algorithm 1 for the Bayesian Lasso.

Due to space constraint, density estimates for a few selected predictor coefficients are displayed at $t = 250, 500$. Since Simulation 1 is the most interesting scenario, posterior densities of a randomly chosen zero coefficient, a nonzero coefficient with a lower magnitude and a nonzero coefficient with a higher magnitude are presented in Figure 2.1. Posterior densities of the selected β_j's in the batch MCMC and DFP tend to show discrepancies in the earlier time points. These dis-

crepancies diminish at $t = 500$, empirically validating the fact that approximate DFP draws converge to the full posterior distribution in time. This conclusion remains valid for Simulations 2 and 3.

While drawing inference from DFP, we also investigate convergence of model parameters and convergence of dynamic partitions of the set of parameters over time. The trace-plot of the ARI between parameter partitions at successive time points shown in Figure 2.1 under Simulation 1 indicates convergence around 1 within the first 100 time points. We also monitor the optimal value c^* chosen over time by the DFP algorithm and found it to stabilize rapidly (see Figure 2.1). Similar investigation in Simulations 2 and 3 lead to equivalent conclusions and hence they have not been included in the figures. Further, we monitor the convergence of $\widehat{\beta}_j^{(t)}$ over time for β_j corresponding to a high signal, low signal and zero signal in the truth. Figure 2.2 shows $\widehat{\beta}_j^{(t)}$ values concentrating around the true data generating parameter as time progresses. It also serves as an empirical assurance that the convergence of $\widehat{\boldsymbol{\Theta}}^{(t)}$ to the true parameter is a reasonable assumption in the theoretical study of DFP.

We also present MSPE, coverage, interval score for the 95% predictive intervals and computation time in seconds per batch of the competing methods for Simulation 1 in Figure 2.1. Figures 2.3 and 2.4 highlight the same quantities for Simulations 2 and 3 respectively, except the computation time which is similar for competitors across the three simulations. Batch MCMC, being a batch method, is expected to converge faster. The predictive inference of DFP improves rapidly and becomes indistinguishable from batch MCMC within $t \approx 100 - 150$ for all three simulations. In contrast, the predictive performance of CDF appears to be inferior to batch MCMC even at $t = 150$. To ensure that the faster decay in MSPE of DFP compared to CDF can actually be attributed to dynamic construction of

parameter subsets at each time, we explore three other versions of DFP for which we update partitions of the parameter set in every 10, 50 and 100 batches. We refer to them as lagged DFP with lag $= 10, 50, 100$, respectively. The regular DFP corresponds to lag $= 1$. The trace-plots of MSPE for the regular DFP (i.e., with lag$= 1$) along with lagged DFP for the Bayesian Lasso Model in the three simulation settings are shown in Figure 2.5. As the value of lag increases, it takes more time for MSPE in the lagged DFP to stabilize. In fact, the figure shows that the MSPE for a lagged DFP with lag$= 100$ takes about 50 more data shards to stabilize compared to the MSPE of the regular DFP. Thus, dynamic partitioning learns posterior correlations among parameters accurately which yields a better approximation of the full posterior than CDF or any other lagged version of DFP in the earlier time points.

The average MSPE, run time, coverage and interval scores of 95% predictive intervals over the last 100 time points for all the competitors are presented in Table 2.1. The results show that in all three simulations, DFP emerges as a computationally efficient replacement for batch MCMC, both in terms of point prediction as well as characterizing predictive uncertainties. Batch MCMC being the gold standard, it shows little higher coverage with little smaller interval length than DFP and CDF. This is due to a little improved point estimation by batch MCMC than both DFP and CDF. CDF demonstrates about the same length of confidence interval with practically the same coverage. As mentioned earlier, naive implementation of CDF demonstrates inferior predictive inference. An improved implementation of CDF presented here, in contrast, loses appeal with minimal gain in computation time over batch MCMC. The SSMC approach also demonstrates similar inferential performance with DFP with a higher computation time.

Sensitivity to the choice of M. Our investigation reveals that for any choice of M,

Table 2.1: Bayesian Lasso performance statistics for MCMC, CDF, DFP and SSMC. Coverage and length are based on the average of the 95% credible predictive intervals in the last 100 batches. The subscript provides standard errors calculated over 10 replications.

Low & High Sparse				
Method	*Predictive Coverage*	*MSPE*	*Int. Score*	*Runtime (sec)*
MCMC	$0.914_{0.019}$	$0.002_{0.000}$	$3.827_{0.345}$	$339.578_{66.343}$
DFP	$0.897_{0.021}$	$0.002_{0.000}$	$3.925_{0.370}$	$148.292_{43.878}$
CDF	$0.902_{0.021}$	$0.002_{0.000}$	$3.897_{0.370}$	$303.215_{73.600}$
SSMC	$0.903_{0.018}$	$0.002_{0.000}$	$3.811_{0.355}$	$234.198_{57.627}$
Sparse				
Method	*Predictive Coverage*	*MSPE*	*Int. Score*	*Runtime (sec)*
MCMC	$0.915_{0.021}$	$0.002_{0.000}$	$3.502_{0.345}$	$400.203_{88.666}$
DFP	$0.898_{0.023}$	$0.002_{0.000}$	$3.592_{0.393}$	$162.788_{58.104}$
CDF	$0.903_{0.023}$	$0.002_{0.000}$	$3.556_{0.380}$	$365.983_{71.200}$
SSMC	$0.912_{0.021}$	$0.002_{0.000}$	$3.512_{0.346}$	$289.179_{66.265}$
Dense				
Method	*Predictive Coverage*	*MSPE*	*Int. Score*	*Runtime (sec)*
MCMC	$0.940_{0.017}$	$4e-05_{1e-05}$	$1.629_{0.121}$	$377.822_{128.891}$
DFP	$0.917_{0.019}$	$4e-05_{1e-05}$	$1.662_{0.148}$	$145.340_{48.056}$
CDF	$0.919_{0.018}$	$4e-05_{1e-05}$	$1.654_{0.143}$	$352.099_{105.388}$
SSMC	$0.943_{0.016}$	$4e-05_{1e-05}$	$1.628_{0.121}$	$278.354_{65.505}$

the mean squared prediction error (MSPE) starts decreasing as time progresses and finally stabilizes. It is also interesting to note that they stabilize at similar values for various choices of M. This is not surprising, since the posterior correlations between parameters become less important factors in prediction when sample size is much larger than the number of parameters. However, for a larger value of M, MSPE stabilizes much more rapidly over time. This is demonstrated for the Bayesian Lasso shrinkage prior with $M = 10$ and $M = 60$ under the three simulation settings, see Figure 2.6. We conclude that when inference is necessary at the earlier time points, one should perhaps adopt a larger choice of M. In contrast, when inference is only required at very large time points, one may construct a more efficient DFP algorithm with a smaller value of M.

Figure 2.1: Performance measures for MCMC, DFP and CDF in the case of Bayesian Lasso under the high and low sparse case are presented in the first row. Coverage and Interval scores are based on the average of the 95% predictive intervals. Confidence bands are based on repeating the analysis over 10 replications. The second row shows estimated densities of selected parameters at $t = 250$ and $t = 500$ for DFP and batch MCMC. Finally, third row presents the trace-plot of the ARI between partitions in two successive time points for DFP and the trace-plot for the optimal value c^* of DFP.

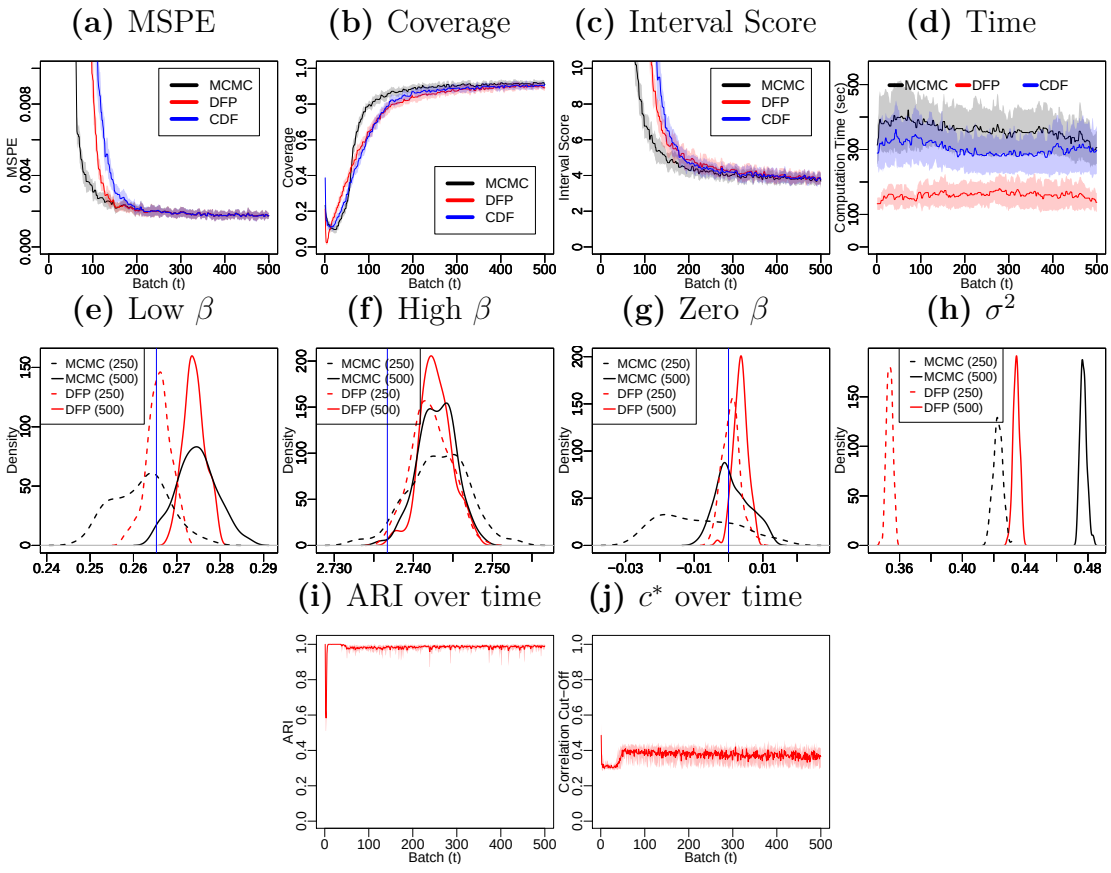


Figure 2.2: Trace-plots of $\widehat{\beta}_j^{(t)}$ for representative β_j parameters under DFP Bayesian Lasso implementation in Simulation 1. We present $\widehat{\beta}_j^{(t)}$ for a low signal, a high signal and a zero signal in the truth. The horizontal line specifies the true value of a parameter.

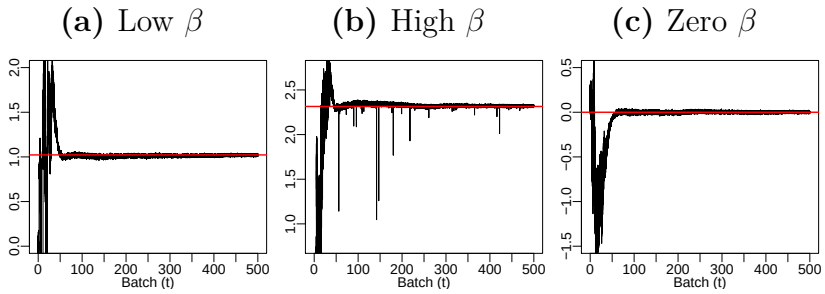


Figure 2.3: Performance measures for MCMC, DFP and CDF for Bayesian Lasso under the sparse case (Simulation 2) are presented. Coverage and Interval scores are based on the average of the 95% predictive intervals. We also show estimated densities for a selected β_j at $t = 250$ and $t = 500$ for both batch MCMC and DFP.

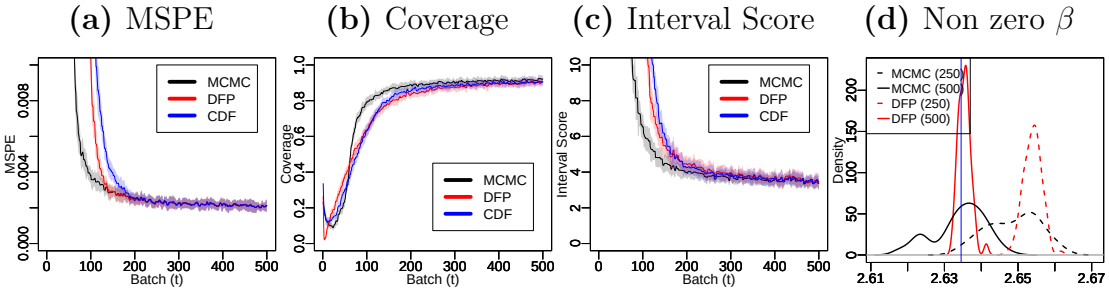


Figure 2.4: Performance measures for MCMC, DFP and CDF for Bayesian Lasso under the dense case (Simulation 3). Coverage and Interval scores are based on the average of the 95% predictive intervals. Estimated densities of selected parameters at $t = 250$ and $t = 500$ for both batch MCMC and DFP are also added.

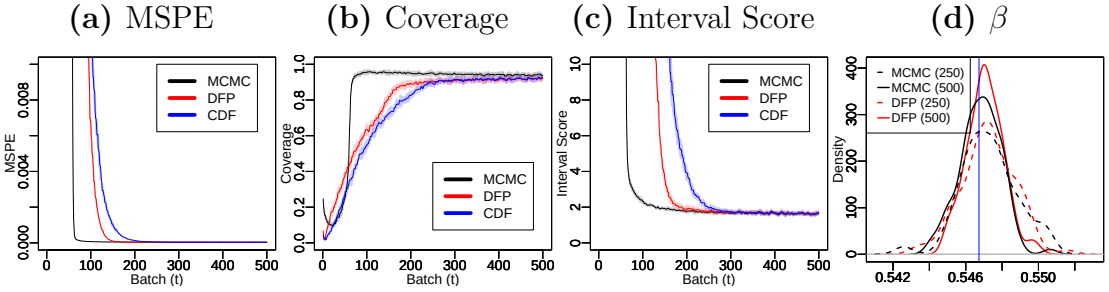


Figure 2.5: The trace-plots of MSPE for regular DFP (lag = 1) and lagged DFP with lag $= 10, 50, 100$ implemented using Bayesian Lasso in Simulations 1-3.

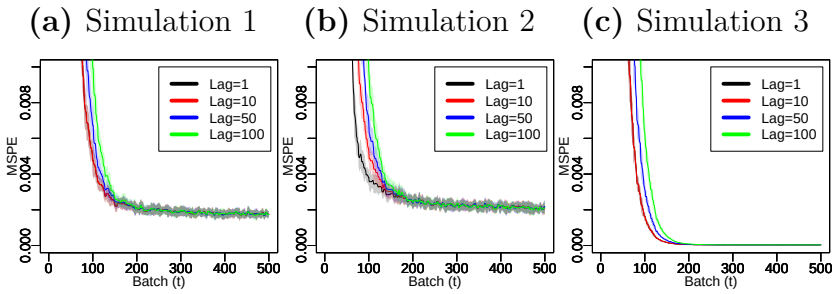
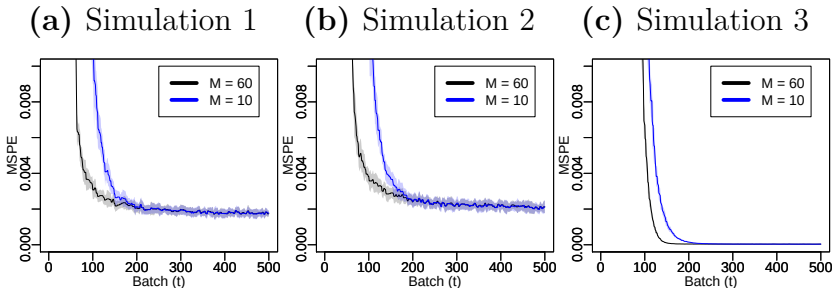


Figure 2.6: Trace-plots of MSPE for $M = 10, 60$ implemented using Bayesian Lasso prior in Simulations 1-3.



2.4.2 DFP with Horseshoe

Our second application considers implementing DFP on the Horseshoe shrinkage prior (Carvalho et al., 2010). The full conditional distributions of parameters along with computational issues in implementing Gibbs sampling with the Horseshoe shrinkage prior are given in Section 2.2.2. The DFP algorithm is employed to incur computational benefits in situations with large p.

The DFP algorithm applied to this problem considers partitioning the parameters $\boldsymbol{\Theta} = \{\boldsymbol{\beta}, \boldsymbol{\lambda}, \boldsymbol{\nu}, \sigma^2, \tau^2, \xi\}$ into $k_t = b_t + 2$ subsets at time t given by

$$\boldsymbol{\Theta}_{G_l^{(t)}} = \left\{ \beta_{i^{(t)}_{m_1+\cdots+m_{l-1}+1}}, \lambda^2_{i^{(t)}_{m_1+\cdots+m_{l-1}+1}}, ..., \beta_{i^{(t)}_{m_1+\cdots+m_l}}, \lambda^2_{i^{(t)}_{m_1+\cdots+m_l}} \right\}, \;\; l = 1, .., b_t,$$

$$\boldsymbol{\Theta}_{G^{(t)}_{b_t+1}} = \left\{ \boldsymbol{\nu} \right\}, \;\; \boldsymbol{\Theta}_{G^{(t)}_{b_t+2}} = \left\{ \sigma^2, \tau^2, \xi \right\}.$$

Table 2.2: Horseshoe performance statistics for MCMC, CDF, SSMC and DFP. Coverage and interval scores are based on the average of the 95% credible predictive intervals of the last 100 batches. Subscripts provide standard errors over 10 simulations.

Low & High Sparse				
Method	*Predictive Coverage*	*MSPE*	*Int. Score*	*Runtime (sec)*
MCMC	$0.924_{0.019}$	$0.002_{0.001}$	$3.725_{1.006}$	$298.126_{52.808}$
DFP	$0.905_{0.020}$	$0.002_{0.000}$	$3.715_{0.341}$	$143.587_{30.989}$
CDF	$0.909_{0.020}$	$0.002_{0.000}$	$3.704_{0.338}$	$289.120_{58.688}$
SSMC	$0.922_{0.021}$	$0.002_{0.001}$	$3.722_{1.006}$	$288.783_{83.226}$
Sparse				
Method	*Predictive Coverage*	*MSPE*	*Int. Score*	*Runtime (sec)*
MCMC	$0.925_{0.021}$	$0.002_{0.001}$	$3.375_{1.004}$	$357.010_{64.220}$
DFP	$0.906_{0.021}$	$0.002_{0.000}$	$3.386_{0.343}$	$164.555_{42.560}$
CDF	$0.910_{0.022}$	$0.002_{0.000}$	$3.372_{0.349}$	$329.129_{83.201}$
SSMC	$0.923_{0.022}$	$0.002_{0.001}$	$3.377_{1.026}$	$338.996_{66.246}$
Dense				
Method	*Predictive Coverage*	*MSPE*	*Int. Score*	*Runtime (sec)*
MCMC	$0.931_{0.018}$	$0.001_{0.000}$	$2.383_{21.448}$	$262.594_{34.915}$
DFP	$0.891_{0.022}$	$4e-05_{1e-05}$	$1.749_{0.180}$	$117.416_{14.589}$
CDF	$0.903_{0.021}$	$3e-05_{1e-05}$	$1.696_{0.162}$	$261.798_{68.321}$
SSMC	$0.932_{0.017}$	$0.001_{0.001}$	$2.221_{3.996}$	$311.438_{70.867}$

Let $\boldsymbol{\beta}_l$ and $\boldsymbol{\lambda}_l$ be the vector of β_js and λ_j^2s, respectively, corresponding to the lth partition. Define $\boldsymbol{S}_{1,l}^{(t)}$, $\boldsymbol{S}_{2,l}^{(t)}$ and $\boldsymbol{S}_{1,l,-l}^{(t)}$ as in Section 2.4.1. Let $\boldsymbol{M}_{\lambda,l} = diag(\boldsymbol{\lambda}_l)$ and $\boldsymbol{\beta}_{-l}$ be the β_js not contained in $\boldsymbol{\beta}_l$. A detailed implementation of DFP for the Horseshoe prior is described in Section A.2.2 of the appendix A.

Figure 2.7 presents dynamically evolving MSPE, coverage, interval score for the 95% predictive interval and computation time in seconds per batch of the competing methods for Simulation 1. As observed in Section 2.4.1, MSPE for DFP falls sharply as time progresses and becomes indistinguishable with the MSPE of batch MCMC after $t \approx 200 - 250$. While accurate point prediction is one of our primary objectives, characterizing uncertainty is of paramount importance given the recent development in the frequentist literature on characterizing uncertainties in high dimensional regressions (Van de Geer et al., 2014; Zhang and Zhang, 2014). Although Bayesian procedures provide an automatic characterization of uncertainty, the resulting credible intervals may not possess the correct frequen-

Figure 2.7: Performance measures for MCMC, DFP and CDF in the case of Horseshoe under the high and low sparse case (Simulation 1) are presented in the first row. Coverage and Interval scores are based on the average of the 95% predictive intervals. The second row shows estimated densities of selected parameters at $t = 250$ and $t = 500$ for both batch MCMC and DFP. Confidence bands are based on the analysis over 10 replications.

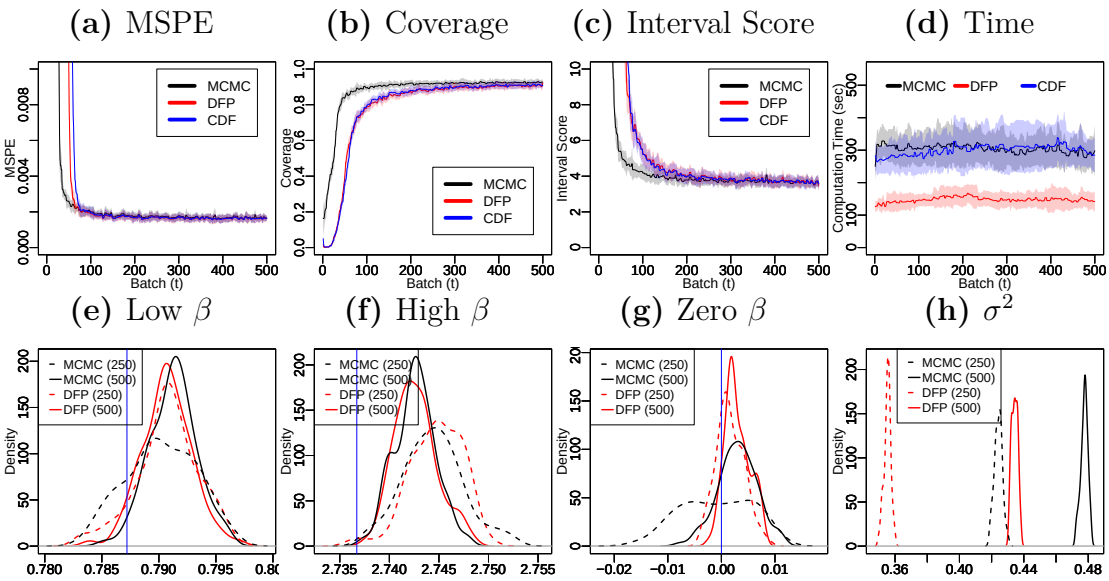
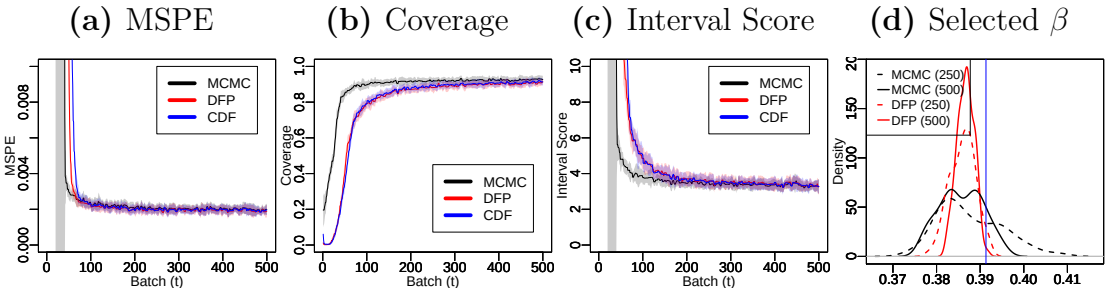


Figure 2.8: Performance measures for MCMC, DFP and CDF for Horseshoe under the sparse case (Simulation 2) are presented. Coverage and Interval scores are based on the average of the 95% predictive intervals. We also show estimated densities of a selected β_j at $t = 250$ and $t = 500$ for both batch MCMC and DFP.



tist coverage in nonparametric/high-dimensional problems (Szabó et al., 2015). An attractive adaptive property of the shrinkage priors, including Horseshoe, is that the lengths of the intervals automatically adapt between the signal and noise variables, maintaining close to nominal coverage. Approximate Bayesian infer-

ence with the DFP algorithm is found to preserve this desirable property of the Horseshoe prior. In fact, Figures 2.7, 2.8 and 2.9 show similar coverage and interval scores for DFP and batch MCMC as time progresses. The inference on the last 100 batches is provided in Table 2.2. Similar to Section 2.4.1, batch MCMC demonstrates marginally better coverage with little narrow predictive intervals, which is due to the little more precise point prediction offered by batch MCMC. In contrast, CDF and DFP offer practically indistinguishable predictive coverage and length. However, we have employed an improved version of CDF for comparison with computation time twice as compared to DFP. SSMC also shows state-of-the-art predictive inference with much higher computation time.

Density estimates for a few selected predictor coefficients are displayed at $t = 250, 500$. Since Simulation 1 is the most interesting scenario, posterior densities of a randomly chosen zero coefficient, a nonzero coefficient with a lower magnitude and a nonzero coefficient with a higher magnitude are presented in Figure 2.7. For nonzero coefficients, the density estimates seem to be similar in DFP and in batch MCMC, though DFP yields marginally narrower credible intervals than batch MCMC corresponding to zero coefficients. We refrain from adding any further discussion on the convergence of partitions or convergence of c^*, since the conclusion is very similar to Bayesian Lasso.

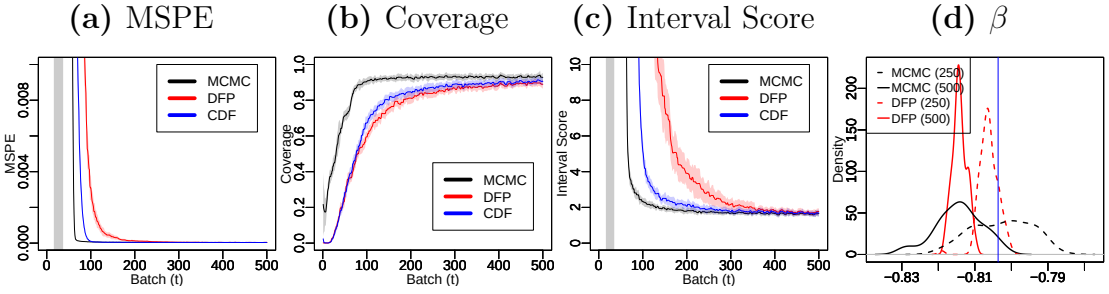
One fundamental advantage of the Horseshoe shrinkage prior over frequentist penalized optimization is its ability to accurately characterize parametric and predictive uncertainties without any user dependent choice of tuning parameters. However, it might lose this appeal due to its high computation time and inability to provide rapid inference with big n and p. DFP applied to the Horseshoe prior solves the computational bottleneck for big n and p, perhaps offering wider applicability to the Horseshoe prior in regression problems at a much larger scale.

Table 2.3: Spike and Lasso performance statistics for MCMC, CDF, SSMC and DFP. MSPE, Coverage and interval scores are based on the average of the 95% credible predictive intervals for the last 100 batches.

Sparse				
Method	*Predictive Coverage*	*MSPE*	*Int. Score*	*Runtime (sec)*
MCMC	$0.921_{0.021}$	$0.002_{0.000}$	$3.479_{0.335}$	$396.730_{97.681}$
DFP	$0.898_{0.023}$	$0.002_{0.000}$	$3.587_{0.388}$	$9.262_{3.476}$
CDF	$0.894_{0.023}$	$0.002_{0.000}$	$3.595_{0.385}$	$395.402_{136.833}$
SSMC	$0.922_{0.02}$	$0.002_{0.001}$	$3.483_{0.379}$	$311.897_{52.019}$
Low & High Sparse				
Method	*Predictive Coverage*	*MSPE*	*Int. Score*	*Runtime (sec)*
MCMC	$0.922_{0.019}$	$0.002_{0.000}$	$3.795_{0.324}$	$393.422_{55.556}$
DFP	$0.897_{0.021}$	$0.002_{0.000}$	$3.929_{0.385}$	$9.406_{2.886}$
CDF	$0.892_{0.021}$	$0.002_{0.000}$	$3.982_{0.380}$	$407.424_{50.365}$
SSMC	$0.925_{0.017}$	$0.002_{0.001}$	$3.802_{0.333}$	$314.783_{45.451}$

We expect similar conclusions to hold for other state-of-the-art shrinkage priors such as, the Generalized Double Pareto (Armagan et al., 2013) and the normal gamma (Griffin et al., 2010) prior distributions.

Figure 2.9: Performance measures for MCMC, DFP and CDF for Horseshoe under the dense case (Simulation 3) are presented. Coverage and Interval scores are based on the average of the 95% predictive intervals. We also show estimated densities of a selected β_j at $t = 250$ and $t = 500$ for both batch MCMC and DFP.



2.4.3 Spike and Lasso

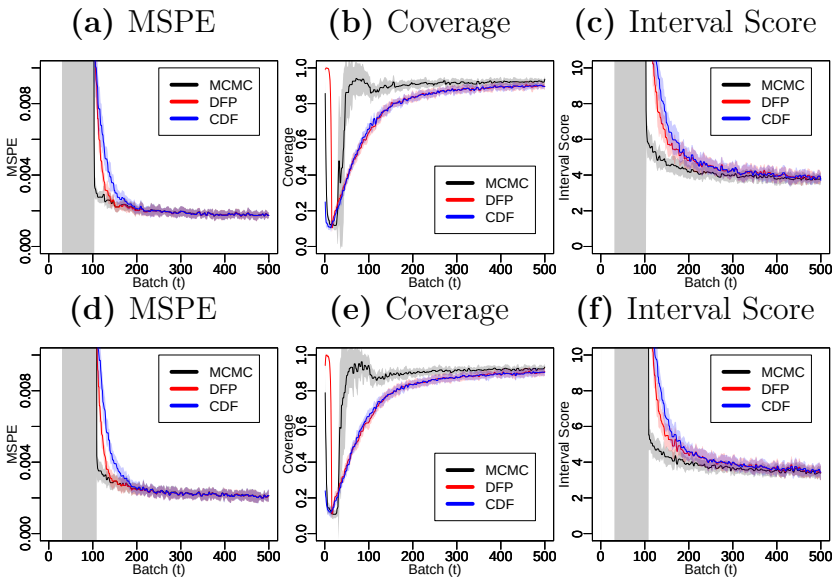
Since spike and slab prior distributions are primarily designed to identify important variables in sparse high dimensional regressions, we investigate DFP with the Spike and Lasso prior for Simulations 1 and 2. Again, Section A.2.3 of appendix A details out the implementation of Algorithm 1 of Spike & Lasso prior.

Figure 2.10 presents the dynamic progression of various performance metrics for DFP, batch MCMC and CDF over $T = 500$ time points. Unlike Sections 2.4.1 and 2.4.2, the operating characteristics of the Spike and Lasso applied to all three competitors take longer time to stabilize. This is not surprising, given that batch MCMC with spike and slab mixture priors is known to offer less accurate performance with a smaller sample size due to the high correlation between various γ_j's. As before, DFP approximates batch MCMC accurately in terms of the operating characteristics. Similar to the earlier sections, Table 2.3 presents predictive inference averaged over last 100 batches for all the competitors under all simulation cases. As observed in the earlier sections, DFP and CDF show practically indistinguishable performance, while batch MCMC yields marginally lower interval scores and little higher coverage, perhaps due to a more precise point estimation. SSMC continues to show competitive performance with a much higher computation time compared to DFP. DFP dynamically learns the partition based on $\mathbf{\Theta}_{1t}$ and $\mathbf{\Theta}_{2t}$. Since we consider sparse examples, the cardinality of the set $\mathbf{\Theta}_{1t}$ is never large, and hence the parameters therein can be updated quickly. Our detailed investigation also reveals that even a large number of partitions of $\mathbf{\Theta}_{2t}$ does not compromise the accuracy of the inference and prediction. This helps to accrue substantial gains in computation time for DFP compared to its competitors, as demonstrated in Table 2.3. In contrast, CDF fixes the partitions in the beginning and is unable to leverage the information of the zero and nonzero β_j's as the approximate posterior sampling progresses.

Representative posterior densities of β_j's from DFP and batch MCMC (presented in Figure 2.11) are centered around the truth and have similar tails. Both Simulations 1 and 2 involve high sparsity, resulting in the posterior density of θ centered at a small value. Again there is a considerable agreement in the posterior

densities of θ from DFP and batch MCMC. Finally, posterior densities of σ^2 for DFP and batch MCMC are found to differ by a small margin from the truth. The trace-plots of $\widehat{\beta}_j^{(t)}$ for representative coefficients with zero, low and high signals in the truth are also shown in Figure 2.12 and they are found to converge to the true parameter values. Finally, we explore how the partitions evolve dynamically and observe that the ARI between partitions at two successive time points quickly converges to 1 with time (see Figure 2.12).

Figure 2.10: Performance measures for MCMC, DFP and CDF with the Spike and Lasso prior under Simulations 1 (1st row) and 2 (second row). Coverage and interval scores are based on the average of the 95% predictive intervals.



2.4.4 Sensitivity to the choice of S

One of the important ingredients in the development of DFP is the choice of the number of Monte Carlo samples S at every time and it is instructive to see the effect on inference with different choices of S. The simulation section presents results of DFP with $S = 500$. To assess the sensitivity to the choice of S in our simulations, we compute DFP after moderately perturbing S. Table 2.4 presents

Figure 2.11: Estimated densities for a few selected β_js, σ^2 and θ at $t = 250$ and $t = 500$. The first row presents results for Simulation 1 while the second row demonstrates performance of DFP in Simulation 2.

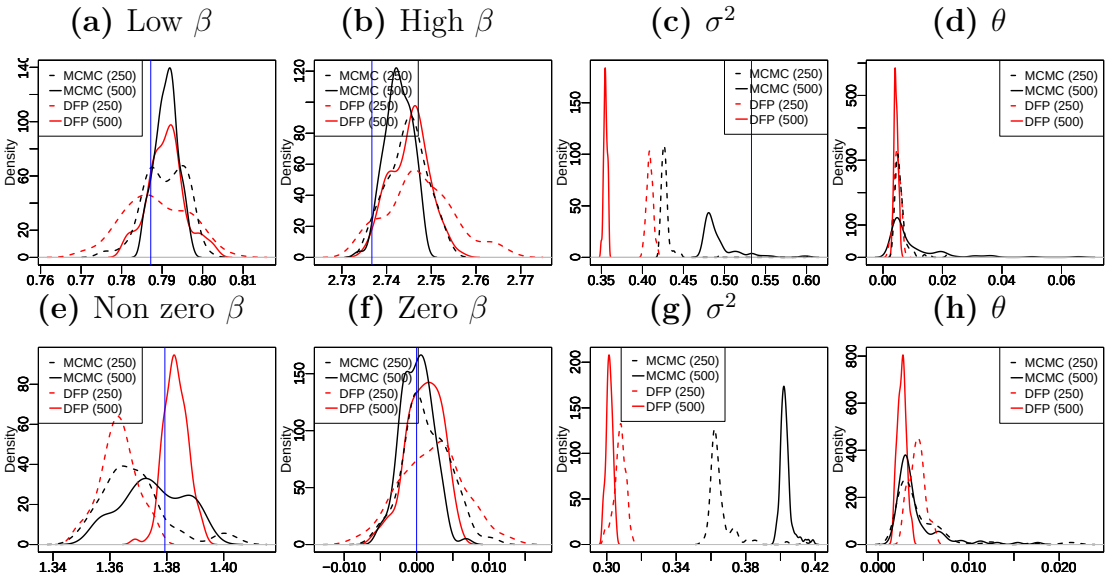
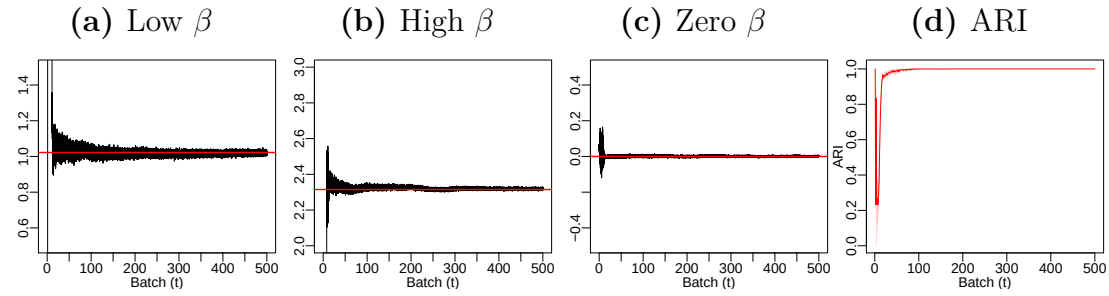


Figure 2.12: Trace-plots for $\widehat{\beta}_j^{(t)}$ for representative parameters in DFP Spike & Lasso implementation under Simulation 1. We include plots for representative predictor coefficients with low signal, high signal and zero signal in the truth. The horizontal line specifies the true value of the parameters. The left most column shows the trace-plot of the ARI for the parameter set partitions at two successive time points.



the predictive inference with DFP for $S = 500, 750, 1000$ in the different simulation cases with the Bayesian Lasso prior. The results show practically indistinguishable inference with different choices of S, with $S = 750$ and $S = 1000$ naturally incurring much more computational cost. In our experience, the inference can be marginally improved with much larger choices of S, though such choices practically

Table 2.4: Bayesian Lasso performance statistics for DFP with $S = 500, 750, 1000$. Coverage and length are based on the average of the 95% predictive intervals on the last 100 batches. The subscript provides standard errors calculated over 10 replications.

Low & High Sparse				
Method	*Predictive Coverage*	*MSPE*	*Int. Score*	*Runtime (sec)*
DFP($S = 500$)	$0.897_{0.021}$	$0.002_{0.000}$	$3.925_{0.370}$	$148.292_{43.878}$
DFP($S = 750$)	$0.906_{0.024}$	$0.002_{0.000}$	$3.957_{0.344}$	$243.176_{48.245}$
DFP($S = 1000$)	$0.912_{0.015}$	$0.002_{0.000}$	$3.954_{0.358}$	$309.542_{44.268}$
Sparse				
Method	*Predictive Coverage*	*MSPE*	*Int. Score*	*Runtime (sec)*
DFP($S = 500$)	$0.898_{0.023}$	$0.002_{0.000}$	$3.592_{0.393}$	$162.788_{58.104}$
DFP($S = 750$)	$0.903_{0.028}$	$0.002_{0.000}$	$3.578_{0.369}$	$248.927_{54.200}$
DFP($S = 1000$)	$0.911_{0.022}$	$0.002_{0.000}$	$3.589_{0.327}$	$316.178_{59.264}$
Dense				
Method	*Predictive Coverage*	*MSPE*	*Int. Score*	*Runtime (sec)*
DFP($S = 500$)	$0.917_{0.019}$	$4e-05_{1e-05}$	$1.662_{0.148}$	$145.340_{48.056}$
DFP($S = 750$)	$0.919_{0.017}$	$4e-05_{1e-05}$	$1.684_{0.143}$	$234.099_{46.498}$
DFP($S = 1000$)	$0.919_{0.016}$	$4e-05_{1e-05}$	$1.678_{0.141}$	$305.354_{46.491}$

diminish any computational advantage of DFP.

2.5 Application to Financial Stock Database

To illustrate the performance of DFP, we implement DFP for a financial data set consisting of minute by minute average log-prices of the NASDAQ stock exchange from September 10, 2018 to November 13, 2018 during trading hours. The data consists of log-prices of Apple stocks along with 3430 assets, and the aim of the data analysis is to evaluate the elasticity of the price of Apple stocks with respect to the prices of the remaining assets. This is of particular interest, since Apple, one of the biggest publicly traded companies in the world, is ubiquitous in portfolios ranging from retirement funds to small portfolios managed by individuals in the financial market. Thus accurate inference on the relationship between Apple and other financial stocks allows better portfolio diversification. We envision it as a high dimensional linear regression problem with the log-price

of the Apple stock as the response and log-prices of other assets as predictors. Along with prediction, the inferential interest lies mainly in identifying important predictors significantly associated with the response. Hence the *Spike & Lasso* prior on regression coefficients are employed.

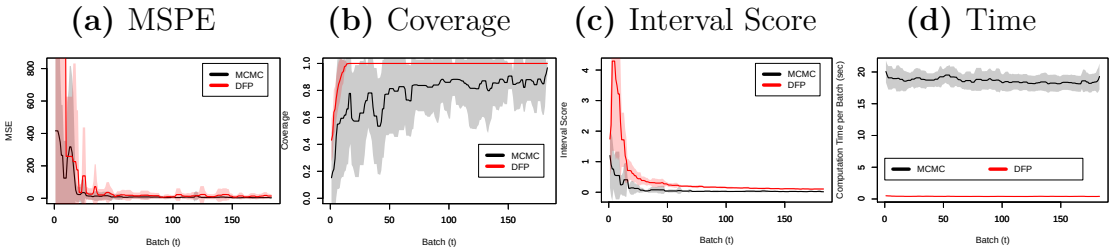
The data includes several assets, such as ETFs, Trust Funds, stock tracker indexes, and banks, which as expected, present a very high degree of collinearity. To avoid less desirable inference due to high collinearity, a few financial assets are removed along with assets which have very few transactions (less than 40), yielding 2015 predictors for the analysis. The data set consists of 18330 observations collected over two months.

To compare the predictive inference of DFP with respect to the gold standard "batch MCMC," the dataset is divided into 183 approximately equal shards to implement DFP and the batch MCMC. Both are implemented 10 times with 10 different permutations of the dataset to minimize the effect of sample ordering on the identification of influential variables. Furthermore, this allows us to examine if the predictive inferential mechanism in DFP is sufficiently robust to the inaccurate posterior approximations at earlier time points.

Figure 2.13 tracks the progression of MSPE, interval score and coverage of 95% predictive intervals for both DFP and batch MCMC as more batches are processed. At time t, the predictive inference is assessed with the data shard obtained at time $t+1$. Similar to simulation studies, the behavior of DFP in the early batches is somewhat erratic due to the inaccurate posterior approximation in the initial phase of the algorithm, though it stabilizes as more data shards arrive. Furthermore, the performances of the competitors become closer as time progresses, with batch MCMC demonstrating marginally superior performance at higher time points. While batch MCMC runs 500 iterations per batch in 18.35

seconds, DFP finishes 500 iterations per batch in 0.40 seconds. Such a dramatic improvement in computation time can be attributed to efficient partitioning of the parameter space as well as parallel inference on parameter partitions at each time.

Figure 2.13: Performance measures for MCMC and DFP. MSPE, coverage and interval scores for 95% predictive intervals are presented. Confidence bands (in a lighter color) are calculated by observing the variations of these metrics over 10 permutations.



Model fitting observes a high degree of multi-modality in the posterior distribution is known to have minimal effects on the predictive inference, but may provide somewhat unreliable inference in terms of variable selection. This is observed and noted in the earlier literature on high dimensional regression (see e.g., Guhaniyogi et al. (2013)). In such cases, it is customary to run the posterior computation multiple times, record the set of variables being identified in each of these runs, and finally declare those variables as influential which have appeared as influential in more than half of the runs. Due to the multi-modality in the posterior distribution, we observe that 10 runs of both DFP and batch MCMC do not lead to the same set of variables identified. In fact, we find a difference in the conclusion between DFP and MCMC in terms of identified variables.

To ensure more reliable inference from DFP and the “gold standard" batch MCMC for variable selection, we run both these competitors 10 more times on the dataset of interest. In these 10 runs, the data is divided into 163 shards with the first shard having 20% observations, and the rest 162 shards all approximately

Table 2.5: Number of times a stock is selected under DFP and MCMC out of 10 runs of both methods.

Company	*DFP*	*MCMC*	*Company*	*DFP*	*MCMC*
Allscripts Healthcare Solutions, Inc.	10	10	SeaSpine Holdings Corporation	6	10
Alphabet Inc.	10	10	Qorvo, Inc.	7	10
Century Aluminum Company	10	10	Costco Wholesale Corporation	7	0
Ferroglobe PLC	10	10	iQIYI, Inc.	8	0
Skyworks Solutions, Inc.	10	10	The Ultimate Software Group, Inc.	7	0
Red Robin Gourmet Burgers, Inc.	9	10	Global Water Resources, Inc.	0	10
Viavi Solutions Inc.	9	10	Kala Pharmaceuticals, Inc.	0	10
The Kraft Heinz Company	8	10	National General Holdings Corp	0	10
Amazon.com, Inc.	7	10	Applied Optoelectronics, Inc.	0	9
Popular, Inc.	7	9	Atlas Air Worldwide Holdings	0	9
Caesarstone Ltd.	7	9	Baozun Inc.	0	9
Microsoft Corporation	8	9	Genprex, Inc.	0	9

equal. We observe that feeding more data early on leads to reliable variable selection with minimal variation between different runs. To provide concrete evidence on this observation, we refer to Table 2.5 which presents all predictors identified by either DFP or batch MCMC in any of the 10 runs. The table also records the number of times among the 10 runs they are identified as influential. It shows that the number of times a predictor is selected by either batch MCMC or DFP is very close to 0 or 10, indicating quite reliable variable selection. Importantly, much less discrepancy is observed between DFP and batch MCMC, with them identifying 17 and 21 variables as influential respectively, with 14 identified by both.

2.6 Conclusion

The emergence of large volumes of high dimensional data mandates that model fitting tools evolve quickly to keep pace with the rapidly growing dimension and size of data. The DFP algorithm proposed in this chapter dynamically partitions the parameter space after observing every data shard and employs fast and approximate Bayesian inference at each partition in parallel. The detailed simulation studies of DFP with popular Bayesian shrinkage priors (Bayesian Lasso, Horse-

shoe and Spike and Lasso) show indistinguishable inference from batch MCMC with a considerable reduction of per batch computation time. Appendix A contains the proof of convergence of the DFP algorithm for high dimensional linear regression as time $t \to \infty$.

Chapter 3

Bayesian Multi-Object Regression

3.1 Introduction

Similar to Chapter 2, Chapter 3 focuses on regressions. However, the predictor variables in the regression framework in Chapter 3 are objects with topological structures, rather than ordinary scalar-valued variables. The motivation of such regressions mainly comes from biomedical applications. Aided by technological advances in both biomedical hardware and software, neuroscientists routinely collect high-dimensional imaging data from multiple sources (modalities) to interrogate the human brain (Sui et al., 2012). Inspection of multiple brain images produces complementary cross information that can be leveraged to combat Alzheimer's disease (AD) and other neurodegenerative disorders (NDs) by advancing foundational cognitive theory, models of disease progression, and biomarker development (Ossenkoppele et al., 2016). For example, ND progression is best tracked via disruptions of brain structure and networks, and are detected by stitching together information across these images (Mandelli et al., 2016; Gorno-Tempini et al., 2008; Brown et al., 2019). This chapter focuses on multi-modal imaging data which include: (a) *network information* in the human brain estimated using functional

magnetic resonance imaging(fMRI) and (b) *structural information* in the human brain obtained using structural magnetic resonance imaging (sMRI), e.g., grey matter (GM) images. Both images are collected on a common brain atlas which segments a human brain into different regions of interest (ROI).

There are several compelling possibilities in terms of relating data on these brain-modalities to phenotypic traits of individuals. This chapter is motivated by a clinical application where we consider predicting a scalar cognitive score used to measure primary progressive aphasia (PPA), an ND with similar pathology to Alzheimer's disease and fronto-temporal dementia, from multi-modal imaging predictors (a) and (b). There are three major inferential objectives of our clinical study. First, neuroscientists are often interested in identifying regions of the brain which are influential in predicting a cognitive score measuring the degree of PPA. Second, it is important to draw predictive inference on a cognitive score based on the multi-modal imaging predictors. Finally, it is of practical interest to examine the inferential and predictive advantage of exploiting cross-information from multi-modal predictors over a single modality.

From a statistical point of view, our inferential problem can be formulated under a regression framework with a scalar response and multi-object predictors that includes a network object. Traditional regression approaches involving a scalar response and object predictors mostly ignore predictor topology leading to sub-optimal inference. As an example, in an object-oriented regression with a network valued predictor as an object, the most popular approaches either choose a few summary measures from the network as predictors (Bullmore and Sporns, 2009), or vectorize the network object into a high dimensional collection of edge weights (Craddock et al., 2009; Richiardi et al., 2011). While this approach can make use of the latest developments in high-dimensional frequentist and Bayesian

regression literature (Tibshirani, 1996; Park and Casella, 2008b; Carvalho et al., 2010), they ignore the fact that coefficients corresponding to edges connected to a common ROI in the network predictor are expected to be correlated a priori.

Of late, there are developments of scalar on image regression approaches exploiting topological information of the image predictors. To this end, a class of methods envisions an image as a collection of spatially correlated predictors (Goldsmith et al., 2014; Li et al., 2015; Feng et al., 2019; Kang et al., 2018; Huang et al., 2013), and accounts for the spatial association between regression coefficients corresponding to the image pixels while estimating them. Another class of approaches visualizes a brain imaging predictor having a structure of a multi-dimensional array or tensor, and proposes regression with a scalar response and a tensor predictor (Zhou et al., 2013; Zhou and Li, 2014; Guhaniyogi et al., 2017; Fan et al., 2019). The latter approaches enjoy the advantage of being more computationally scalable than the former by implicitly using the spatial information. While these methods can be directly employed to identify important edges related to the scalar response or to assess the impact of each edge in predicting the response, they have not been applied in the literature to identify important ROIs influencing the response. One can perhaps add a post-processing step to these methods and declare an ROI to be influential if at least one of edges related to the ROI is deemed influential, though this strategy does not lead to uncertainty quantification in identifying influential ROI. As we highlight later, our proposed approach provides model-based uncertainty quantification for inference on influential ROIs. Yet another class of methods (Guha and Guhaniyogi, 2021; Guha and Rodriguez, 2020) proposes regression approaches with scalar response and network valued predictors and enables drawing inference on influential ROIs. While these regression approaches establish the importance of preserving the structure and

network topology in image objects for better inference and prediction, the referenced works mainly regard object topology for a single image object but principled linkages among image objects are not made and thus inference on scalar outcomes is made without regard to valuable information that are shared across these objects. For instance, in our clinical case study of language dysfunction in PPA patients, existing methods do not directly combine structural information relating to neuronal atrophy with network information on brain connectivity to jointly model deficits in language comprehension scores. Failure to consider the structure and cross information from multiple images have generally a negative impact on ND research in terms of lower detection power (Li et al., 2018), bias in estimated effects (Dai and Li, 2021), statistical inefficiency (Dai and Li, 2021), and sensitivity of results to noise (Calhoun and Sui, 2016a). Additionally, all these approaches consider low-dimensional structure of the network coefficient, while the approach we propose does not rely on such assumptions.

We employ a Bayesian model for multi-object regression with the brain network and GM images as predictors with a scalar response. In particular, we construct a prior distribution framework on multi-modal predictor coefficients which exploits ideas from variable selection and Bayesian shrinkage framework in high dimensions to account for both the structural and network topology in multi-object data and allows the information in separate image objects to complement and re-enforce each other in their relation to the scalar outcome. To elaborate on it, we begin with a common brain atlas for both image predictors to ensure that it provides an organizing principle that links together structural and network information via a shared set of ROIs. ROI-specific binary latent indicators taking values in $\{0, 1\}$ are then introduced. To jointly borrow information from the GM image and the network predictor, our prior construction on their respective coefficients ensures

that an ROI-specific binary indicator estimated to be zero automatically enforces that the structural information corresponding to all voxels from that particular ROI and all network edges connected to that particular ROI have no effect on the response. Further, the prior construction on network predictor coefficient preserves *transitivity* property and *hierarchical* constraint of the network predictor, as described in Section 3.2. While this chapter does not explicitly make use of the structural information in the GM images by careful spatial modeling of GM image coefficient, it partially exploits the structural information by respecting the *hierarchical* arrangement of voxels and ROIs in the prior construction step. A more explicit spatially varying coefficient modeling of GM image coefficients in the multi-modal regression is computationally challenging and is left for future exploration. The prior construction achieves efficient computation and accurate predictive inference of the language score, and offers identification of ROIs which are key to study neuronal atrophy. Moreover, our framework attaches uncertainty in identifying these ROIs and produces well-calibrated interval estimates for the multi-modal regression coefficients.

Our proposed approach is considerably different from the existing statistical literature on multi-modal data integration. In particular, there have been a class of unsupervised multi-modal analysis built on matrix or tensor factorization (Lock et al., 2013), or methods exploiting structural connectivity information from diffusion tensor imaging (DTI) in the prior construction for the functional connectivity analysis from functional MRI (fMRI) data (Xue et al., 2015). In contrast, we focus on the supervised analysis with a scalar response and multi-modal predictors. To this end, Xue et al. (2018) proposes regression on disease status on low-frequency fluctuation (fALFF) from resting-state fMRI scans, voxel based morphometry (VBM) from T1-weighted MRI scans, and fractional anisotropy (FA) from DTI

scans. In the same vein Li and Li (2021) develops a factor analysis-based linear regression model, and Dai and Li (2021) extends this framework to account for non-linear association between a scalar response and multi-modal predictors. While these supervised approaches do form linkages among image modalities, they do not properly model within image correlations and thus are not able to address our inferential goals of jointly modeling information across images while maintaining within image topology. Moreover, all these approaches are frequentist in nature and do not naturally offer uncertainty in predicting the response.

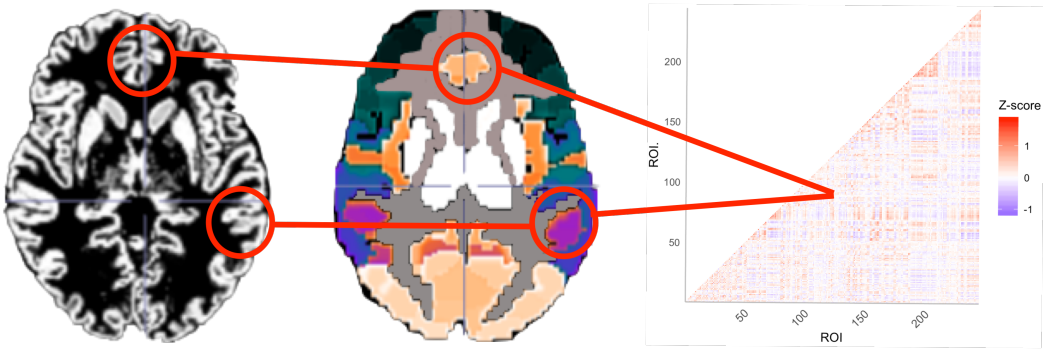
The rest of the chapter proceeds as follows. Section 3.2 provides a description of the multi-modal data that motivates our development in this chapter. Section 3.3 describes the novel prior framework to draw inference and prediction with multi-modal predictors and Section 3.4 discusses posterior computation of the proposed model. Empirical investigations with data generated under various simulation settings are reported in Section 3.5. Section 3.6 shows an analysis of the multi-modal dataset with simulated response. Finally, Section 3.7 summarizes the idea laid out in this chapter and highlights some of the extensions of our model to be explored in the near future.

3.2 Motivating clinical application

This chapter is motivated by a clinical application derived from multimodal imaging studies conducted on patients with a diagnostic variant of Primary Progressive Aphasia (PPA), known as the nonfluent/aggramatic variant (nfvPPA) characterized by motor speech and grammar loss and left inferior frontal atrophy (Gorno-Tempini et al., 2008).

Clinical images: Imaging data is acquired on 26 nfvPPA patients during the course of clinical research activity. Data is collected from the following imag-

Figure 3.1: Schematic of the multi-object brain imaging data structure for a PPA patient. (a) Structural image encoding voxel-level gray matter (GM) probability, (b) Brainnetome atlas parcellation of the brain into anatomical ROIs, (c) Network image obtained by calculating the pairwise Pearson correlation Z-score for the average fMRI signal in each ROI. Red circles and lines connect (a) structural and (c) network information from images via the (b) parcellated atlas. Thus, the atlas provides an organizing hierarchy that links together structural information (GM) at the voxel level with network information indexed by pairs of ROIs (fMRI).



(a) Structural image (b) Atlas (c) Network image

ing modalities: sMRI derived gray matter (GM) (Figure 3.1a) which measures the likelihood a voxel containing neuronal cell bodies; and task-free resting state functional magnetic resonance imaging (fMRI) to measure brain activation via neuronal oxygen consumption in subjects at rest. All images are registered to the Montreal Neurological Institute (MNI) template space with voxels parcellated into 246 ROIs using the Brainnetome atlas such that images across modalities and subjects can be directly compared and each voxel is nested in an anatomically defined ROI (Figure 3.1b) (Fan et al., 2016). Findings from the prior clinical studies allows us to focus only on 19 of these ROIs which are more likely to be related to nfvPPA.

For each subject, a 'brain network' represented by a symmetric adjacency matrix is obtained from the fMRI image by considering rows and columns of this matrix corresponding to different ROIs and entries corresponding to the Z-scores obtained by transforming the Pearson correlation between average fMRI data of two ROIs (Figure 3.1c).

Language loss in nfvPPA patients is driven by neurodegeneration in the *left inferior frontal* region but the dual role of structural damage and brain connectivity in language loss is not well characterized (Mandelli et al., 2016). To better understand the neural underpinnings of language dysfunction in 26 nfvPPA patients, measures of language deficiency must be regressed on sophisticated multimodal images, specifically the GM map which capture focal neurodegeneration, and, fMRI brain connectivity networks which capture disruptions of brain connectivity. The scientific objective also includes identifying ROIs influential in related to the language loss. The next section describes a novel regression framework needed to answer these inferential questions.

3.3 Bayesian Multi-object Regression

This section details out the model development and prior formulation, including the hyper-parameter specification.

3.3.1 Model Framework

For the ith subject, let $y_i \in \mathbb{R}$ denote the observed continuous scalar response (e.g., a language score) and $\boldsymbol{A}_i$ denote the weighted network predictor. We assume that network predictors of all subjects are defined on a common set of nodes, with elements of $\boldsymbol{A}_i$ encoding the strength of network connections between different nodes for the i-th subject. In particular, the network predictor $\boldsymbol{A}_i$ is expressed in the form of a $P \times P$ matrix with the (p, p')-th entry of the matrix $a_{i,(p,p')}$ signifying the strength of association between the pth and p'th node, where $p, p' = 1, ..., P$ and P is the number of network nodes. This chapter specifically focuses on networks that contain no self relationship, i.e., $a_{i,(p,p)} \equiv 0$, and are undirected ($a_{i,(p,p')} = a_{i,(p',p)}$).

Such assumptions hold for the data pertaining to Section 3.2, where $\boldsymbol{A}_i$ represents the brain connectome network matrix obtained from the fMRI scan, with each node representing a specific brain region of interest (ROI). Let $\boldsymbol{g}_{i,1}$,...,$\boldsymbol{g}_{i,P}$ denote the $V_1, ..., V_P$ dimensional structural objects in regions $\mathcal{R}_1$,...,$\mathcal{R}_P$, respectively. In the context of Section 3.2, they represent volumetric elements (voxels) of the GM image from the P ROIs. This multi-object characterization of fMRI and GM data allows structural information across these images to share a common set of ROIs, where each voxel is nested within an ROI. The nested structure of voxels within ROIs provides biologically plausible organization and is instrumental for variable selection and computation as detailed in the upcoming methodological development.

With an additional information on covariates $\boldsymbol{x}_i$ of dimension $m \times 1$ (which may be behavioral or biological, e.g., age, gender, race/ ethnicity), we propose the linear model with multi-object predictors as,

$$y_i = \beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}_{\boldsymbol{x}} + \sum_{p=1}^{P} \boldsymbol{g}_{i,p}^T \boldsymbol{\beta}_p + \langle \boldsymbol{A}_i, \boldsymbol{\Theta} \rangle / 2 + \epsilon_i, \;\; \epsilon_i \sim N(0, \tau^2). \tag{3.1}$$

Here $\boldsymbol{\beta}_p$ is the coefficient of dimension $V_p \times 1$ corresponding to the structural image in $\mathcal{R}_p$, $\boldsymbol{\beta}_{\boldsymbol{x}}$ represents coefficients of $\boldsymbol{x}_i$ in $\mathbb{R}^m$, τ^2 is the variance of the observational error and $\langle \cdot, \cdot \rangle$ is the Frobenius inner product between two matrices. Similar to $\boldsymbol{A}_i$, the $P \times P$ coefficient matrix $\boldsymbol{\Theta} = ((\theta_{p,p'}))$ is assumed to be symmetric with zero diagonal entries, so that $\langle \boldsymbol{A}_i, \boldsymbol{\Theta} \rangle = 2 \sum_{1 \le p < p' \le P} a_{i,(p,p')} \theta_{p,p'}$. Such a simplification is useful in drawing connection between the multi-object regression model (3.1) to a linear regression framework given by,

$$y_i = \beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}_{\boldsymbol{x}} + \sum_{p=1}^{P} \boldsymbol{g}_{i,p}^T \boldsymbol{\beta}_p + \sum_{1 \le p < p' \le P} a_{i,(p,p')} \theta_{p,p'} + \epsilon_i, \;\; \epsilon_i \sim N(0, \tau^2). \tag{3.2}$$

(3.2) keeps a tab on the network node index of the structural object $\boldsymbol{g}_{i,p}$. Thus such a formulation allows linking the information on ROI level data from the two types of predictors to draw inference on influential ROIs through biologically inspired prior construction on model coefficients as described in the next section.

3.3.2 Prior Distribution on Multi-Object Coefficients

Our joint prior construction on coefficients $\boldsymbol{\beta}_p$'s and $\{\theta_{p,p'} : p < p'\}$ for multi-object predictors is fundamental to exploiting topology of the objects and cross-information among them by forming principled linkages among objects at the node/ROI level. The prior construction is aimed at: (a) identification of influential nodes/ROIs; (b) accurate estimation of voxel-level coefficients for structural objects; and (c) guaranteeing efficient computation of the posterior for the proposed prior. We envision the problem identifying influential ROIs/nodes from the multi-object predictors as a high-dimensional variable selection problem and formulate prior distributions on multi-object coefficients building upon the existing literature on prior constructions for high-dimensional regression coefficients.

To this end, two classes of prior distributions on coefficients are typically employed in an ordinary high dimensional regression literature. The traditional approach is to develop a discrete mixture of prior distributions (George and McCulloch, 1993, 1997; Scott and Berger, 2010). These methods enjoy the advantage of inducing exact sparsity for a subset of parameters, but may face computational challenges when the number of predictors is large. As an alternative to this approach, continuous approximation to the discrete mixture priors (Carvalho et al., 2010; Armagan et al., 2013) have emerged which induce approximate sparsity in high-dimensional parameters. Such prior distributions can mostly be expressed as global-local scale mixtures of Gaussians (Polson and Scott, 2010), are compu-

tationally efficient and offer an approximation to the operating characteristics of discrete mixture priors.

The direct application of a variable selection prior on multi-object coefficients is unappealing for multiple reasons. First, an ordinary variable selection prior on coefficients aims at identifying important predictors (which in our application are network edges and voxel-level structural images), rather than influential ROIs/nodes. Second, we seek to impose an additional restriction on the prior construction of $\boldsymbol{\Theta}$ motivated by the neuro-scientific application, that is, if any of the pth and p'th nodes are un-influential in predicting the response, the edge coefficient $\theta_{p,p'}$ corresponding to the edge between pth and p'th nodes is unimportant. Third, our prior specification should ensure that if a node/ROI is not important in predicting the response, then any voxel in the ROI is also uninfluential in predicting the response. This restriction is relevant due to the hierarchical arrangement of voxels and ROIs in our motivating application and is referred to as the *hierarchical constraint*. Finally, we expect the matrix of coefficients $\boldsymbol{\Theta}$ (which itself can be regarded as describing a weighted network) to exhibit *transitivity effects*, that is, we expect that if the interactions between nodes p and p' and between nodes p' and p'' both influence the response, the interaction between nodes p and p'' will likely be influential (see, e.g., Li et al. (2013)). An ordinary variable selection prior on multi-object coefficients does not necessarily conform to all these requirements.

We offer a prior construction exploiting the literature on both discrete and continuous mixture variable selection priors to fulfill our inferential goals. To elaborate on it, let $\xi_1, ..., \xi_P$ denote the binary inclusion indicators corresponding to the P ROIs/nodes taking values in $\{0, 1\}$, with $\xi_p = 0$ determining no effect of the pth ROI/nodes on the response from all covariates. The network edge

coefficient $\theta_{p,p'}$ is then endowed with a variable selection prior given by

$$\theta_{p,p'}|\lambda_{p,p'},\tau,\sigma_\theta,\xi_p,\xi_{p'} \overset{ind.}{\sim} \xi_p\xi_{p'}N(0,\tau^2\sigma_\theta^2\lambda_{p,p'}^2)+(1-\xi_p\xi_{p'})\delta_0,\ \ p<p', \tag{3.3}$$

where δ_0 corresponds to the Dirac-delta function, $\lambda_{p,p'}$ are local parameter corresponding to the (p,p')th edge and σ_θ is the global parameter for the network coefficient. The prior closely mimics the spike-and-slab variable selection structure with an important difference. While an ordinary spike-and-slab prior introduces a binary inclusion indicator corresponding to each variable, (3.3) enforces $\theta_{p,p'}=0$ when either $\xi_p=0$ or $\xi_{p'}=0$. Such a formulation is sensible from a network perspective as it implies that the edge connecting two network nodes is insignificant in predicting the response when at least one of the network nodes is not influential. Additionally, the formulation naturally incorporates transitivity effects in the network coefficient $\boldsymbol{\Theta}$. We further assign half-Cauchy distributions on $\sigma_\theta \sim C^+(0,1)$ and $\lambda_{p,p'} \overset{ind.}{\sim} C^+(0,1)$ to complete the prior specification on the network coefficient. Integrating out σ_θ and $\lambda_{p,p'}$ in (3.3), $\theta_{p,p'}|\tau,\xi_p=1,\xi_{p'}=1$ follows the popular horseshoe prior (Carvalho et al., 2010) which offers a flexible prior structure for precise estimation of nonzero network edge coefficients a posteriori.

The structural object coefficient $\boldsymbol{\beta}_p \in \mathcal{R}^{V_p}$ for the pth ROI/nodes is modeled using $\boldsymbol{\beta}_p=\xi_p\boldsymbol{\gamma}_p$, where $\boldsymbol{\gamma}_p=(\gamma_{p,1},...,\gamma_{p,V_p})^T$ is a vector of the same dimension as $\boldsymbol{\beta}_p$. To estimate voxel level effects in the pth ROI on the response, each element of $\boldsymbol{\gamma}_p$ is assigned a horseshoe shrinkage prior which takes the following scale-mixture representation,

$$\gamma_{p,j}|\eta_{p,j},\Delta_p,\tau \sim N(0,\tau^2\Delta_p^2\eta_{p,j}^2),\ \ \eta_{p,j} \overset{i.i.d.}{\sim} C^+(0,1),\ \ \Delta_p \overset{i.i.d.}{\sim} C^+(0,1), \tag{3.4}$$

for $j=1,...,V_p$; $p=1,...,P$. The prior structure (3.4) induces approximate spar-

sity in voxel-level GM coefficients $\boldsymbol{\gamma}_p$ by shrinking the components which are less influential toward zero while retaining the true signals (Polson and Scott, 2010). Finally, the binary inclusion indicators are assigned Bernoulli prior distribution $\xi_p \overset{i.i.d.}{\sim} Ber(\nu)$ with $\nu \sim Beta(a_\nu, b_\nu)$ to account for multiplicity correction (Scott and Berger, 2010). Notably, an estimate of the posterior probability of the event $\{\xi_p = 1\}$ shows the uncertainty in identifying the pth ROI to be influential. Thus, $P(\xi_p = 1|\text{Data})$ close to 1 or 0 signifies strong evidence in favor of identifying the pth ROI to be active. The prior specification is completed by assigning a normal prior on coefficients of $\boldsymbol{\beta}_x$, α and $IG(a_\tau, b_\tau)$ on the error variance τ^2.

3.4 Posterior Computation

Although summaries of the posterior distribution cannot be computed in closed form, full conditional distributions for all the parameters are available and mostly correspond to standard families (described in appendix B). Thus, posterior computation can proceed through a Markov chain Monte Carlo algorithm. While a naive implementation of such an algorithm to jointly update $(\boldsymbol{\beta}_p^T : p = 1, .., P)^T$ and $(\theta_{p,p'} : 1 \leq p < p' \leq P)^T$ is viable for small values of P and $V_1, ..., V_P$, it entails complexity of $\sim Q^3$, where $Q = \sum_{p=1}^{P} V_p + P(P-1)/2$, which may lead to intractable computation for moderately large values of P and $V_1, ..., V_P$. To address this issue, we follow the procedure outlined in Guha and Guhaniyogi (2021) which allows computation at $\sim n^3$ complexity. Since in multi-modal neuroimaging applications n is typically much smaller than Q, this approach leads to substantial computational savings. Details of the Markov chain Monte Carlo algorithm and the efficient sampling procedure for multi-modal coefficients are presented in Appendix B.

The MCMC sampler is run for $10,000$ iterations, with the first 5000 discarded

as burn-in. All posterior inference is based on post burn-in samples. The average effective sample size for post burn-in iterations averaged over all $\boldsymbol{\Theta}$ and $\boldsymbol{\beta}_p$'s for all cases are over 1500, indicating fairly uncorrelated post burn-in iterates to draw inference.

We have implemented our code in R (without using any C++, Fortran, or Python interface) on a cluster computing environment with three interactive analysis servers, 56 cores each with the Dell PE R820: 4x Intel Xeon Sandy Bridge E5-4640 processor, 16GB RAM and 1TB SATA hard drive. Different replications of the model are implemented under a parallel architecture by making use of the packages `doparallel` and `foreach` within R. The computation times of running 5000 MCMC iterations with $P = 20$ and $V_1 = ... = V_P = 20$ is given by 2.82 min on average across all simulations.

L (suitably thinned) post burn-in MCMC samples $\xi_p^{(1)}, ..., \xi_P^{(L)}$ of the binary indicator ξ_p are used to empirically assess if the pth ROI is significantly associated with the response. In particular, the pth ROI $\mathcal{R}_p$ is related to the response if $\sum_{l=1}^{L} \xi_p^{(l)} > t$, for $0 < t < 1$. The ensuing simulation section computes the true positive rates (TPR) and false positive rates (FPR) for various choices of t. For the real data section, we use $t = 0.5$ to decide which ROIs are influential in predicting the response.

3.5 Simulation Studies

In this section we compare inferential and out-of-sample predictive performance of our proposed Bayesian Object Oriented Modeling (BOOM) approach to that of a few representative ordinary Bayesian and frequentist high dimensional regression methods. Both ordinary frequentist and Bayesian high-dimensional regression competitors treat edges between nodes in the undirected network pre-

dictor $\boldsymbol{A}_i$ as a "long vector of predictors" and regress response y_i on vectors $\boldsymbol{a}_i = (a_{i,(p,p')} : 1 \leq p < p' \leq P)^T$ and $\boldsymbol{g}_i = (\boldsymbol{g}_{i,1}^T, ..., \boldsymbol{g}_{i,P}^T)^T$, thereby ignoring the relational nature of $\boldsymbol{A}_i$. Horseshoe prior (Carvalho et al., 2010) is used on the regression coefficients in the Bayesian high dimensional regression competitor due to its state-of-the-art empirical performance in regressions with both sparse and not-so-sparse settings. We implement the Horseshoe prior regression using the R package `horseshoe` (van der Pas et al., 2019) by setting the `method.tau` as "haflCauchy" and `tau` as "Jeffreys" (which stands for assigning Jeffrey's prior on the variance component of horseshoe) to obtain a traditional full Bayesian implementation of Horseshoe. Furthermore, we obtain $10,000$ samples and burn-in 5000 to obtain 5000 samples with no thining. On the other hand, the frequentist high dimensional regression competitor adopts a penalized optimization framework with the minimax convex penalty (MCP) on the predictor coefficients (Zhang, 2010). MCP is implemented using the `ncvreg` (Breheny and Huang, 2011) package in R, with the penalty parameter of MCP chosen through ten-fold cross validation technique. An in depth comparison with these methods will indicate the relative advantage of exploiting the structure of the network predictor and utilizing the linkage of information in the multi-modal predictors by our BOOM approach under various degrees of sparsity and the number of estimated parameters in the model. Horseshoe is allowed to draw 10,000 MCMC samples, with the first 5000 used as burn-in and inference is drawn on the remaining 5000 samples.

3.5.1 Data Generation

In all our simulations, we generate response from the following model,

$$y_i = \beta_{0,t} + \sum_{p=1}^{P} \boldsymbol{g}_{i,p}^T \boldsymbol{\beta}_{p,t} + \langle \boldsymbol{A}_i, \boldsymbol{\Theta}_t \rangle / 2 + \epsilon_i, \;\; \epsilon_i \sim N(0, \tau_t^2), \tag{3.5}$$

where the subscript t indicates the true data generating parameters. We set the number of regions to be equal to $P = 20$ and the sample size $n = 150$ in all simulations. The number of cells in every region in the structural object is considered equal, i.e., $V_1 = \cdots = V_P = V$ in all simulations. We present simulation cases by varying V, as discussed later.

Simulating true coefficients $\boldsymbol{\Theta}_t$ and $\boldsymbol{\beta}_{p,t}$. To simulate the true coefficients $\boldsymbol{\Theta}_t$ and $\boldsymbol{\beta}_{p,t}$, we first simulate binary variables $\xi_{1,t}, ..., \xi_{P,t} \overset{i.i.d.}{\sim} Ber(\nu_t)$ with $\xi_{p,t} = 1$ sets the p-th region to be influential in predicting the response. Since $(1 - \nu_t)$ is the probability of a region not being "influential," it is referred to as the node sparsity parameter. The coefficient corresponding to the edge connecting the p-th and p'-th region is drawn from the following mixture distribution,

$$\theta_{p,p',t}|\xi_{p,t}, \xi_{p',t} \sim \xi_{p,t}\xi_{p',t}N(\mu_\theta, \sigma_\theta^2) + (1 - \xi_{p,t}\xi_{p',t})\delta_0, \quad \theta_{p,p',t} = \theta_{p',p,t}; \ p < p'. \quad (3.6)$$

(3.6) ensures that any edge connecting to the p-th region in the network predictor is un-influential if the p-th region is un-influential, i.e., $\xi_{p,t} = 0 \Rightarrow \theta_{p,p',t} = 0$ for all $p' \in \{1, .., P\}$. Similarly, corresponding to each un-influential region $\mathcal{R}_p$, the $V \times 1$ dimensional structural predictor coefficient $\boldsymbol{\beta}_{p,t}$ is set at $\mathbf{0}$. When $\xi_{p,t} = 1$, i.e., the p-th region is influential, we randomly choose $\upsilon_t = 0.4$ proportion of cell coefficients in the p-th region to be nonzero and rest are set to be zero. These nonzero coefficients within $\boldsymbol{\beta}_{p,t}$ are simulated from $N(\mu_{\beta_p}, \sigma^2_{\beta_p})$. All simulations fix $\mu_\theta = 1$, $\sigma^2_\theta = 1$, $\mu_{\beta_p} = 1$ and $\sigma^2_{\beta_p} = 1$.

Simulation cases. For a comprehensive simulation study, we consider 14 cases after varying V and the node sparsity parameter $(1 - \nu_t)$, as summarized in Table 3.1. These 14 cases include two different scenarios to simulate the network predictor, as we describe below.

Scenario 1: In Cases 1–12, the upper triangular entries of the undirected net-

Table 3.1: It presents the different simulation cases. Here ν_t is the probability of a region being active and V is the number of cells per region. Cases 1-12 represent dense network predictors with all edges present and referred to as *Scenario 1*, where as Cases 13 and 14 use network predictors with generated from different stochastic block models. Thus these two cases are referred to as *Scenario 2*.

Case	Node sparsity $(1-\nu_t)$	Cells per region (V)	Case	Node sparsity $(1-\nu_t)$	Cells per region (V)
Case 1	0.9	10	Case 8	0.8	25
Case 2	0.9	15	Case 9	0.7	10
Case 3	0.9	20	Case 10	0.7	15
Case 4	0.9	25	Case 11	0.7	20
Case 5	0.8	10	Case 12	0.7	25
Case 6	0.8	15	Case 13	0.7	20
Case 7	0.8	20	Case 14	0.7	20

work predictor matrix $\boldsymbol{A}_i$ is simulated from a standard normal distribution, resulting in a dense network predictor (i.e., there is an edge between any pair of nodes). These 12 cases are together referred to as *Scenario 1*.

Scenario 2: To further assess the performance of competitors under network predictors with different structures, the network predictor is generated following a stochastic block-model in Cases 13 and 14. In Case 13, we assume that each brain network has three local clusters with high within-cluster and low between-cluster connectivity. More specifically, the matrices $\boldsymbol{A}_i$ consist of three symmetric block diagonal matrices of dimensions 6×6, 7×7, and 7×7, respectively. Elements in these matrices are drawn from $\mathrm{N}(j, j^2)$ where $j \in \{1, 2, 3\}$, for the j-th block diagonal. The off-diagonal blocks are highly sparse, with very few non-sparse elements denoting connections between nodes in different clusters, randomly chosen from N(0, 1). In Case 14, each network predictor consists of 3 block diagonal matrices of dimensions 6×6, 7×7, and 7×7. As before, the elements in these matrices have been drawn from $\mathrm{N}(j, j^2)$ where $j \in \{1, 2, 3\}$, for the j-th block

diagonal. However, in this case the elements in the off-diagonal matrices have been drawn from $\mathrm{N}(2,1)$, $\mathrm{N}(3,1)$, and $\mathrm{N}(4,1)$. These two cases are referred to as *Scenario 2* to differentiate them from the cases in *Scenario 1*. The error variance τ_t^2 is fixed at 1 under all simulation settings.

Identification of Influential Regions

Table 3.2 shows the true positive rate (TPR) and false positive rate (FPR) of identifying the truly influential ROIs by the three competing models, averaged over 100 simulations. While BOOM approach allows identification of influential ROIs from the ROI specific latent binary indicators in a principled Bayesian manner as described in Section 3.4, ordinary MCP and Horseshoe are not designed for ROI identification. However, to compare them with BOOM in terms of inference on ROIs, we devise a strategy to select influential ROIs from MCP and Horseshoe via post-processing methods. For MCP, the p-th ROI $\mathcal{R}_p$ is identified as influential if at least one of the voxels in $\mathcal{R}_p$ in the structural predictor or one of the edges connected to the p-th ROI in the network predictor turns out to be significant in the regression. In other words, $\mathcal{R}_p$ is influential if one of elements of $\boldsymbol{\beta}_p$, $\{\theta_{p,p'} : p' > p\}$ and $\{\theta_{p',p} : p' < p\}$ (referring to (3.2)) is estimated to be nonzero. Since the ordinary Horseshoe prior estimates all coefficient to be nonzero using their posterior median, we first apply a post-processing step following Li and Pati (2017) to identify which of these coefficients are nonzero. We then consider $\mathcal{R}_p$ to be influential if at least one of the coefficients in $\boldsymbol{\beta}_p$, $\{\theta_{p,p'} : p' > p\}$ and $\{\theta_{p',p} : p' < p\}$ is estimated to be nonzero in the post-processing step of Horseshoe.

The TPR values close to 1 and FPR values close to 0 under all cases (see Table 3.2), except Case 12, suggests overwhelmingly accurate detection of influential ROIs by BOOM. MCP is the second best performer in this regard, also yielding

Table 3.2: True Positive Rates (TPR) and False Positive Rates (FPR) for identifying the truly influential regions for the three competitors are presented under all simulation cases. Highest TPR and lowest FPR are boldfaced in each case. Results are averaged over 100 replications.

Node Sparsity $(1 - \nu_t) = 0.9$, Scenario 1

		True Positive Rate			*False Positive Rate*		
Cases	V	BOOM	HS	MCP	BOOM	HS	MCP
1	10	**1.00**	**1.00**	0.87	**0.00**	0.82	0.02
2	15	0.99	**1.00**	0.84	**0.00**	0.86	0.03
3	20	**1.00**	**1.00**	0.87	**0.00**	0.84	0.04
4	25	0.99	**1.00**	0.79	**0.00**	0.95	0.04

Node Sparsity $(1 - \nu_t) = 0.8$, Scenario 1

		True Positive Rate			*False Positive Rate*		
Cases	V	BOOM	HS	MCP	BOOM	HS	MCP
5	10	**1.00**	**1.00**	0.99	**0.00**	0.80	0.04
6	15	0.99	**1.00**	0.99	**0.00**	0.87	0.04
7	20	0.99	**1.00**	0.99	**0.01**	0.84	0.03
8	25	**1.00**	**1.00**	0.91	**0.10**	0.86	0.08

Node Sparsity $(1 - \nu_t) = 0.7$, Scenario 1

		True Positive Rate			*False Positive Rate*		
Cases	V	BOOM	HS	MCP	BOOM	HS	MCP
9	10	**1.00**	**1.00**	0.91	**0.00**	0.82	0.03
10	15	**1.00**	**1.00**	0.88	**0.00**	0.77	0.02
11	20	**1.00**	**1.00**	0.79	**0.14**	0.90	0.08
12	25	**1.00**	**1.00**	0.54	**0.74**	0.98	0.03

Node Sparsity $(1 - \nu_t) = 0.7$, Scenario 2

		True Positive Rate			*False Positive Rate*		
Cases	V	BOOM	HS	MCP	BOOM	HS	MCP
13	20	**1.00**	**1.00**	0.79	**0.14**	0.87	0.08
14	20	0.99	**1.00**	0.83	**0.11**	0.90	0.19

high TPR and low FPR for high sparsity cases. On the other hand, Horseshoe shows both high TPR and FPR, identifying almost all of the ROIs as influential. As sparsity decreases and V increases, the performance of all models tend to dete-

riorate, with case 12 demonstrating high FPR for BOOM and low TPR for MCP. Importantly, BOOM is able to offer uncertainty in identifying $\mathcal{R}_p$ as influential using $P(\xi_p = 1|\text{Data})$, unlike its competitors. The performance does not appear to be affected by different structures of the network predictor in *Scenario 1* and *Scenario 2*.

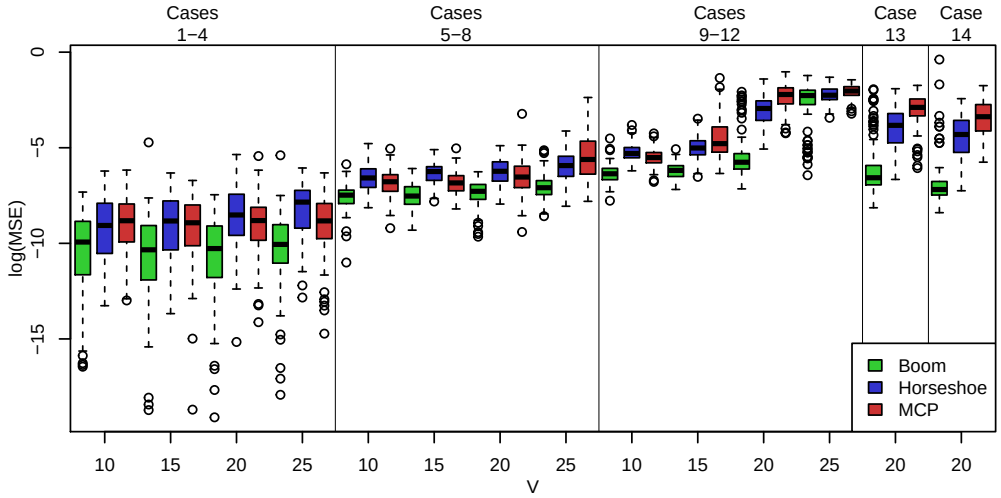
Estimation of Parameters $\boldsymbol{\Theta}_t$ and $\boldsymbol{\beta}_{p,t}$

Figures 3.2 and 3.3 present point estimation along with uncertainty quantification in estimating $\boldsymbol{\Theta}_t$ and $\boldsymbol{\beta}_t = (\boldsymbol{\beta}_{1,t}^T, ..., \boldsymbol{\beta}_{P,t}^T)^T$, respectively. The point estimation of every competitor is assessed using mean squared errors (MSE) of estimating the coefficients corresponding to the network predictor and the structural predictor. Since both $\boldsymbol{\Theta}$ and $\boldsymbol{\Theta}_t$ are symmetric with zero diagonals, the MSE for the network predictor coefficient is given by $2\sum_{p<p'}(\theta_{p,p',t} - \widehat{\theta}_{p,p'})^2/P(P-1)$, where $\widehat{\theta}_{p,p'}$ is the point estimate of $\theta_{p,p'}$. Similarly, we compute and present MSE for the structural predictor coefficients given by $\sum_{p=1}^{P} ||\boldsymbol{\beta}_{p,t} - \widehat{\boldsymbol{\beta}}_p||^2/VP$, with $\widehat{\boldsymbol{\beta}}_p$ representing the point estimate of $\boldsymbol{\beta}_p$. The point estimates are taken to be the posterior median for the Bayesian competitors.

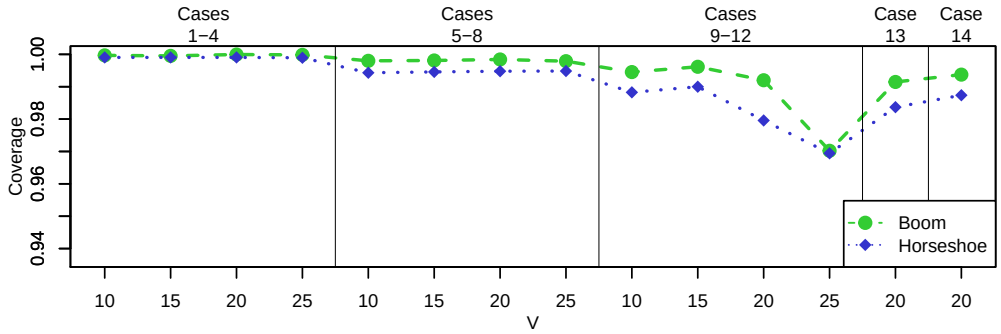
Both Figures 3.2 and 3.3 show that the proposed Bayesian Object Oriented Method (BOOM) outperforms Horseshoe and MCP in all 14 cases. When both V is moderately large and true sparsity level is moderate, i.e. in cases 11, 12, 13, 14, we perform overwhelmingly better than both competitors due to exploiting the network information and linkage between the network and the structural predictor. Since an overwhelming number of coefficients are set to zero in cases 1-8, we modestly outperform our competitors. This might be attributed to the fact that very high degree of sparsity in the truly influential ROIs leads to high degree to sparsity in the regression coefficients in the truth, which is conducive

Figure 3.2: Estimation of $\mathbf{\Theta}$: Figures present mean squared error, coverage and length of 95% credible interval for $\mathbf{\Theta}$.

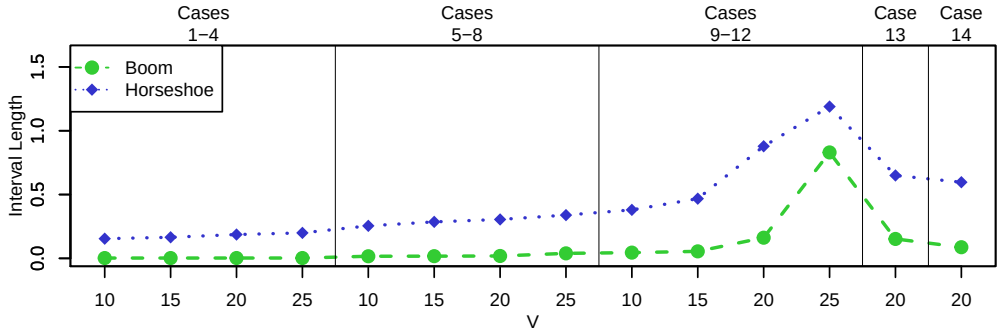
(a) Mean Squared Error



(b) Coverage of 95% CI



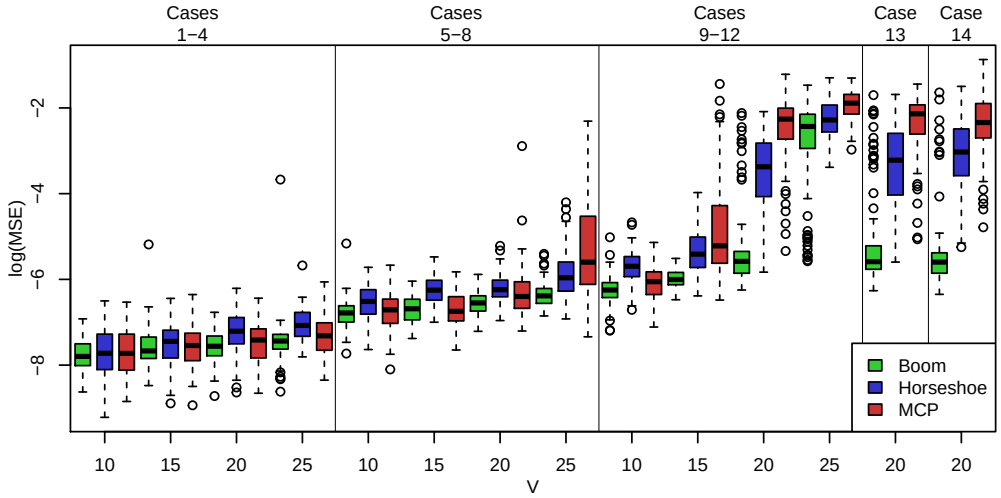
(c) Length of 95% CI



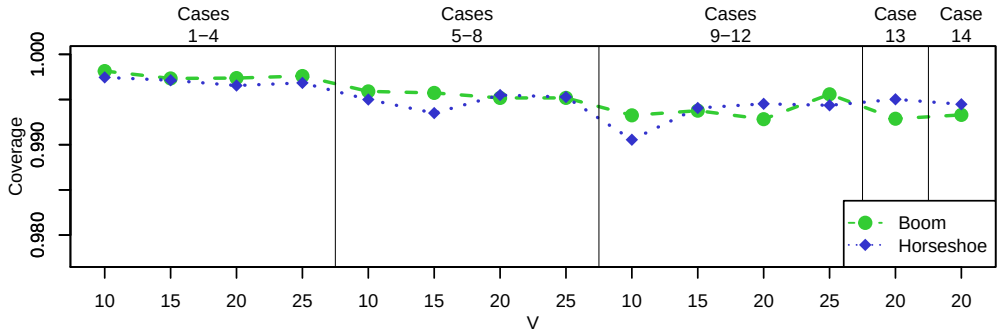
for ordinary high dimensional regression which treats network edges as one set of predictors. As we decrease sparsity or increase V keeping sample size fixed, the

Figure 3.3: Estimation of $\boldsymbol{B}$: Figures present mean squared error, coverage and length of 95% credible interval for $\boldsymbol{B}$.

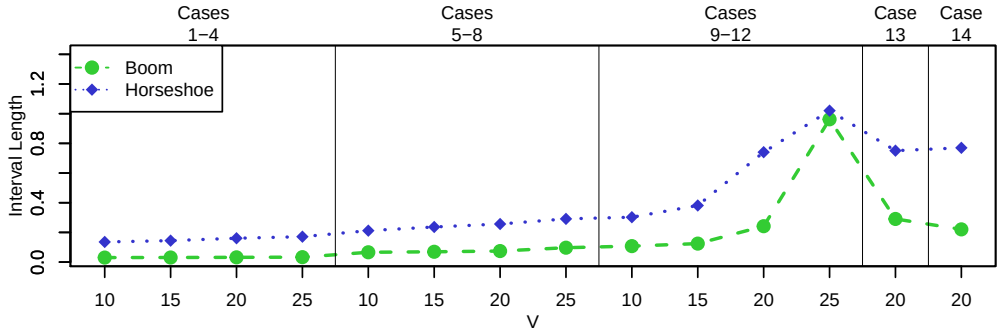
(a) Mean Squared Error



(b) Coverage of 95% CI



(c) Length of 95% CI



performance of all competitors deteriorate significantly, with BOOM continuing to show an edge over the other two.

While both Bayesian competitors BOOM and Horseshoe provide automatic characterization of uncertainty, the resulting confidence intervals may not have the correct frequentist coverage in high dimensional regressions (Szabó et al., 2015). Thus, in order to assess uncertainty in estimating $\boldsymbol{\Theta}_t$ from Bayesian competitors, we evaluate the length and coverage of 95% credible intervals averaged across coefficients in $\boldsymbol{\Theta}$ and present them in Figure 3.2 for all cases. Similar quantities are presented for $\boldsymbol{B}$ in Figure 3.3. Both figures show close to nominal coverage of BOOM under all cases. As sparsity decreases and V increases, the uncertainty associated with BOOM seems to be more which results in an increase in the length of the credible intervals. Importantly, under all cases, BOOM enjoys similar coverage as Horseshoe with much narrower credible intervals. The more precise uncertainty quantification of BOOM over Horseshoe is presumably due to incorporating the structural information of predictors. One important observation from the simulation studies is that we consistently obtain average coverage of the regression coefficients above the nominal coverage of 95%. While this is somewhat less desirable, it is a general phenomenon with high dimensional Bayesian shrinkage priors in presence of large number of unimportant predictors. A brief discussion on this issue is offered by Bhattacharya et al. (2016a), where they also observe average coverage of all coefficients being 1, while the average coverage of truly nonzero coefficients is close to nominal for horseshoe shrinkage priors. To confirm with their earlier observation on horseshoe shrinkage prior, Table 3.3 separately shows average coverage of truly active and inactive coefficients for both BOOM and Horseshoe in two randomly chosen cases (cases 5 and 7). Note that an element $\theta_{p,p'}$ in the network coefficient matrix is truly nonzero, referred to as "active" coefficients, if $\xi_{p,t} = 1$ and $\xi_{p',t} = 1$ in the simulation design; $\theta_{p,p'}$ is otherwise referred to as an "inactive" otherwise. The results corroborate the findings of

Table 3.3: Coverage of the 95% Probability Intervals for of network coefficients for Cases 5 and 7.

	$\boldsymbol{\Theta}$		
Method	*Active*	*Inactive*	*All*
	Case 5		
Boom	**0.928** (0.141)	**1** (0)	**0.997** (0.00399)
Horseshoe	0.82 (0.216)	**1** (0.00157)	0.993 (0.00811)
	Case 7		
Boom	**0.882** (0.253)	**1** (0.000617)	**0.997** (0.00589)
Horseshoe	0.746	**1**	0.996

Bhattacharya et al. (2016a), i.e., the average coverage of truly active coefficients is little below the nominal coverage, while the noisy/inactive predictive coefficients show coverage of 1, which is responsible for inflating the average coverage of regression coefficients. We found similar behavior for the voxel coefficients $\boldsymbol{B}$.

Predictive Inference

The predictive inference from different models are compared based on the point prediction and characterization of predictive uncertainties using $n^* = 100$ out-of-sample observations. We employ the mean squared prediction error (MSPE) to assess point prediction, which is the average squared error distance between the true and predictive response. Additionally, coverage and length of 95% predictive intervals are used to evaluate uncertainty quantification from the Bayesian competitors. Table 3.4 present results on predictive inference for all competitors.

With high sparsity in cases 1-4, all competitors perform similarly in terms of MSPE. BOOM starts outperforming its competitors as sparsity decreases, i.e., in cases 5-14, and the performance gap widens as we increase V and decrease sparsity. Digging a bit further, we observe that with moderate degree of sparsity

Table 3.4: Mean Squared Prediction Error (MSPE), average coverage and average length of 95% predictive intervals for BOOM, Horseshoe (HS) and MCP are presented for various simulation scenarios. Lowest MSPE in each case in boldfaced. Results are averaged over 100 replications.

Node Sparsity $(1-\nu_t) = 0.9$, Scenario 1								
		MSPE			*Avg. Coverage of 95% PI*		*Avg. length of 95% PI*	
Cases	V	BOOM	HS	MCP	BOOM	HS	BOOM	HS
1	10	**1.12**	1.14	1.14	0.72	0.88	1.38	2.62
2	15	**1.20**	1.24	1.22	0.77	0.92	1.71	3.17
3	20	**1.21**	1.38	1.29	0.81	0.95	2.03	3.89
4	25	1.45	1.52	**1.44**	0.99	0.99	0.02	0.17
Node Sparsity $(1-\nu_t) = 0.8$, Scenario 1								
		MSPE			*Avg. Coverage of 95% PI*		*Avg. length of 95% PI*	
Cases	V	BOOM	HS	MCP	BOOM	HS	BOOM	HS
5	10	**1.24**	1.49	1.43	0.74	0.87	2.25	4.00
6	15	**1.42**	1.89	1.68	0.79	0.92	2.83	5.07
7	20	**1.61**	2.16	2.32	0.84	0.96	3.47	6.08
8	25	**1.99**	3.12	7.03	0.78	0.91	4.57	7.80
Node Sparsity $(1-\nu_t) = 0.7$, Scenario 1								
		MSPE			*Avg. Coverage of 95% PI*		*Avg. length of 95% PI*	
Cases	V	BOOM	HS	MCP	BOOM	HS	BOOM	HS
9	10	**1.60**	2.26	2.01	0.81	0.92	3.18	5.74
10	15	**1.97**	3.37	8.55	0.88	0.96	4.27	8.22
11	20	**6.01**	22.74	52.69	0.92	0.96	7.64	18.20
12	25	**39.29**	68.21	89.46	0.95	0.92	21.79	28.03
Node Sparsity $(1-\nu_t) = 0.7$, Scenario 2								
		MSPE			*Avg. Coverage of 95% PI*		*Avg. length of 95% PI*	
Cases	V	BOOM	HS	MCP	BOOM	HS	BOOM	HS
13	20	**6.11**	24.09	49.28	0.92	0.94	7.61	17.92
14	20	**6.72**	27.28	53.35	0.92	0.94	7.17	18.44

and larger values of V, both Horseshoe and MCP over-shrink nonzero coefficients in order to estimate zero coefficients, which leads to poor performance of these methods. In terms of predictive uncertainty, BOOM offers slight under-coverage with high sparsity and smaller values of V, and the coverage becomes very close to nominal as sparsity decreases and V increases. While Horseshoe offers marginally better coverage than BOOM, it yields predictive intervals almost twice of the size of BOOM. Overall, BOOM appears to be a much better performer than its competitors in terms of both inference and prediction under a variety of simulation settings.

Sensitivity Analysis

Our model fitting requires user-dependent choice of the hyper-parameters $a_\tau, b_\tau, a_\nu, b_\nu$. Further, the MCMC algorithm needs to set the initial value for the region indicators. This section investigates sensitivity of inference to such choices. Recall that the analysis presented before sets $a_\tau = b_\tau = 1$ and $a_\nu = b_\nu = 1$. This section explores model performance by setting $a_\tau = b_\tau$, $a_\nu = b_\nu$, and varies them in a wide range. More specifically, we monitor performance by varying $a_\tau = b_\tau \in \{0.1, 10\}$ and $a_\nu = b_\nu \in \{0.1, 10\}$. We also explore two starting points for the MCMC iterations of region selection indicators. The first one starts by setting all region-specific indicators equal to 0, $\xi_p^{(0)} = 0 \quad \forall p \in \{1, \ldots, P\}$, i.e., no region is influential. The second one starts by setting all region-specific indicators equals 1, i.e., $\xi_p^{(0)} = 1 \quad \forall p \in \{1, \ldots, P\}$. Table 3.5 shows the MSE for the different combinations after simulating 100 datasets according to a representative case (case 7) of our simulation study. We find that the impact of this different specifications makes no significant difference. This is perhaps due to the fact that they are either tempered by the data as in the case of a_τ and a_τ, are deeply embedded

in the model as in the case of a_ν and b_ν or their effects are nullified by the burn-in period of the MCMC sampling as with the initial values for the region specific indicators.

3.6 Analysis of PPA Data with a simulated response

In this section, we consider the clinical application described in Section 3.2. As described in Section 3.2, in the course of our data analysis we regress the language deficiency score on the GM map and the brain connectivity network for the 26 subjects, where we construct 19×19 symmetric matrix $\boldsymbol{A}$ representing the brain connectivity network. The second modality on GM object is unstructured and it represents information on gray matters for 8 voxels for each of the 19 regions of interest.

3.6.1 Data Exploration

Even though our preliminary analysis suggests limited signal with only 26 subjects, we would like to show evidence that our modeling framework and assumptions are realistic for this data. In particular, we will empirically investigate the validity of (a) homoscedastic error assumption; and (b) the assumption of fixed predictor coefficient across subjects by fitting the model on the real data with 26 subjects. We would also like to check the predictive performance of our approach with the ordinary horseshoe for the real data.

We proceed to fit BOOM and ordinary horseshoe for this data by running 10,000 MCMC iterations of which we use 5000 for our burn-in period as in the simulations. Figure 3.4 shows the box-plots of the residuals for both models. We

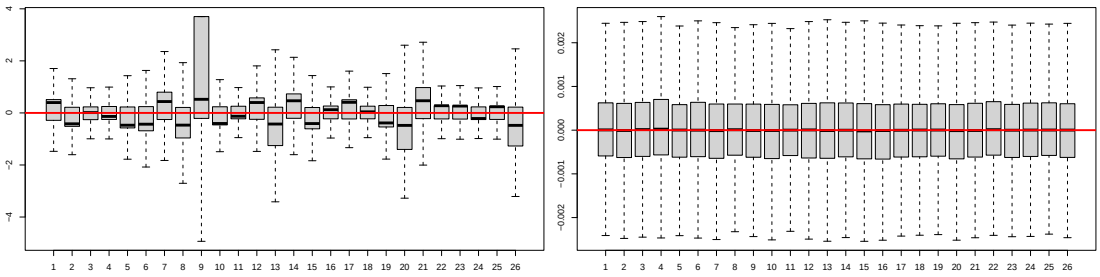
Table 3.5: Sensitivity analysis for the MSE of case 7 varying initialization settings and hyper-parameters.

	$\boldsymbol{\Theta}$	$\boldsymbol{B}$	*All*
$g_p^{(0)} = 1$			
$a_\nu = b_\nu = 0.1$			
$a_\tau = b_\tau = 0.1$	0.002 (0.00077)	0.002 (0.00029)	0.002 (0.00052)
$a_\tau = b_\tau = 1$	0.002 (0.00085)	0.002 (0.00031)	0.002 (0.00057)
$a_\tau = b_\tau = 10$	0.002 (0.00081)	0.002 (0.00029)	0.002 (0.00054)
$a_\nu = b_\nu = 1$			
$a_\tau = b_\tau = 0.1$	0.002 (0.00078)	0.002 (0.0003)	0.002 (0.00053)
$a_\tau = b_\tau = 1$	0.002 (0.00082)	0.0019 (0.0003)	0.002 (0.00055)
$a_\tau = b_\tau = 10$	0.002 (0.00082)	0.002 (0.00029)	0.002 (0.00054)
$a_\nu = b_\nu = 10$			
$a_\tau = b_\tau = 0.1$	0.002 (0.00081)	0.002 (0.00029)	0.002 (0.00054)
$a_\tau = b_\tau = 1$	0.002 (0.00083)	0.002 (0.00029)	0.002 (0.00055)
$a_\tau = b_\tau = 10$	0.002 (0.00082)	0.002 (0.0003)	0.002 (0.00055)
$g_p^{(0)} = 0$			
$a_\nu = b_\nu = 0.1$			
$a_\tau = b_\tau = 0.1$	0.002 (0.00081)	0.002 (0.00027)	0.002 (0.00053)
$a_\tau = b_\tau = 1$	0.002 (0.0008)	0.002 (0.00027)	0.002 (0.00052)
$a_\tau = b_\tau = 10$	0.002 (0.00082)	0.002 (0.0003)	0.002 (0.00055)
$a_\nu = b_\nu = 1$			
$a_\tau = b_\tau = 0.1$	0.002 (0.00081)	0.002 (0.0003)	0.002 (0.00055)
$a_\tau = b_\tau = 1$	0.002 (0.00083)	0.002 (0.00026)	0.002 (0.00054)
$a_\tau = b_\tau = 10$	0.002 (0.00079)	0.002 (0.00028)	0.002 (0.00053)
$a_\nu = b_\nu = 10$			
$a_\tau = b_\tau = 0.1$	0.002 (0.00082)	0.002 (0.00029)	0.002 (0.00054)
$a_\tau = b_\tau = 1$	0.002 (0.00083)	0.002 (0.00028)	0.002 (0.00055)
$a_\tau = b_\tau = 10$	0.002 (0.00082)	0.002 (0.00028)	0.002 (0.00054)

do not observe any outlying residual for BOOM, while there are clear signals of overfitting corresponding to the residuals for horseshoe. Importantly, this figure provides evidence supporting homoscedastic error assumption for the data.

Figure 3.4: Distribution of the Residuals for the BOOM and Horseshoe models

(a) Residuals BOOM **(b)** Residuals Horseshoe



We also perform leave one out cross-validation, and compute the mean square prediction error for BOOM and horseshoe. Table 3.6 shows significantly better point prediction offered by BOOM compared to the ordinary horseshoe prior on coefficients.

Table 3.6: Mean square predictive error of leave one out cross validation for BOOM and Horseshoe. Best result is presented in bold. Standard deviations are given in parentheses.

Method	MSPE
BOOM	**0.058**
	(0.244)
Horseshoe	0.316
	(1.160)

Lastly, we proceed to check the assumption on fixed predictor coefficient across subjects. In particular, the patients might not share the same coefficient structure and it might be necessary to add flexibility to the coefficient structure to allow different coefficients for different patients. In order to explore this issue, we separately regress response on every voxel level predictor from GM and every network

edge from $\boldsymbol{A}$, and use a subject-specific regression coefficient. In particular, we run the following regression model with the language score separately on each voxel level data from GM, given by,

$$y_i = \beta_{i,p,v} G_{i,p,v} + \epsilon_{\beta,i,p,v}, \;\; \beta_{i,p,v} \sim N(\beta_{0,p,v}, \sigma^2_{\beta,p,v}), \; \forall i, p, v \tag{3.7}$$

Similarly, we regress the language score separately on every network edge, given by,

$$y_i = \theta_{i,p,p'} A_{i,p,p'} + \epsilon_{\theta,i,p,p'}, \;\; \theta_{i,p,p'} \sim N(\theta_{0,p,p'}, \sigma^2_{\theta,p,p'}), \; \forall i, p \leq p' \tag{3.8}$$

In this analysis we compare the posterior distributions of $\beta_{1,p,v}, ..., \beta_{26,p,v}$ with the prior mean $\beta_{0,p,v}$ for every region p and every voxel v within the p-th region. Similarly, the posterior distributions of $\theta_{1,p,p'}, ..., \theta_{26,p,p'}$ with the prior mean $\theta_{0,p,p'}$, for every $1 \leq p < p' \leq P$. Figure 3.5(a) shows the posterior distributions of GM coefficient corresponding to all 26 individuals in region $p = 1$ and voxel $v = 4$. Similarly, Figure 3.5(b) shows the posterior distributions of coefficient for the network edge connecting regions 2 and 3 corresponding to all 26 individuals. According to the figure, the prior mean falls within the inter-quartile range for almost all the subject-specific coefficients (except for subject 9). Further, Figure 3.6 presents the proportion of subjects for which the prior mean parameter falls in the interquartile range for every network cell coefficient and GM voxel coefficient. All values appear to be close to 1 indicating no significant difference of prior mean from every subject-specific coefficient. Hence, the assumption of identical regression coefficient appears to be a reasonable assumption.

Figure 3.5: Distribution of the coefficients for every subject for the region 1 voxel 4 and network coefficient corresponding to regions 2 and 3. In red the mean of the prior mean for the coefficients.

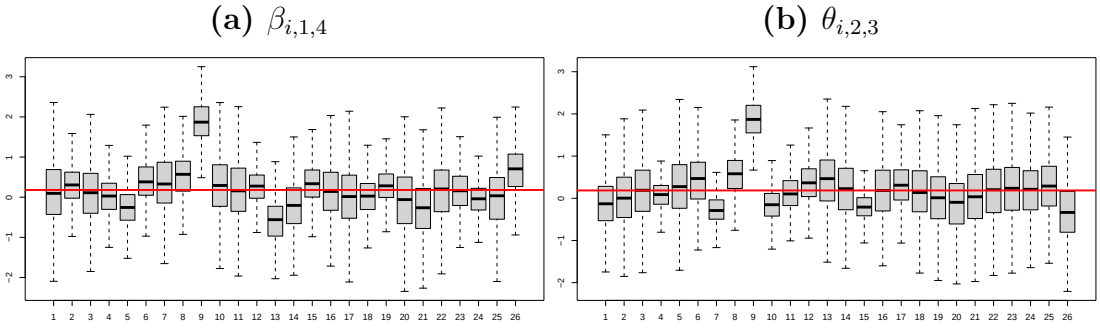
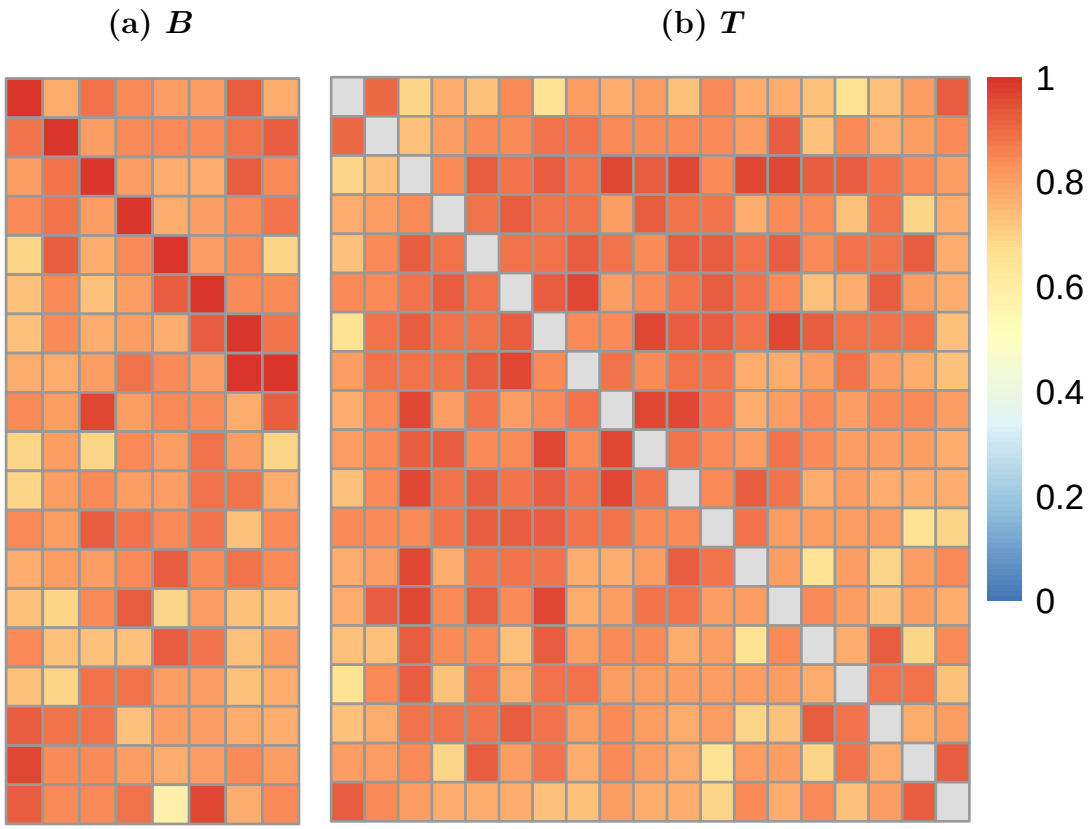


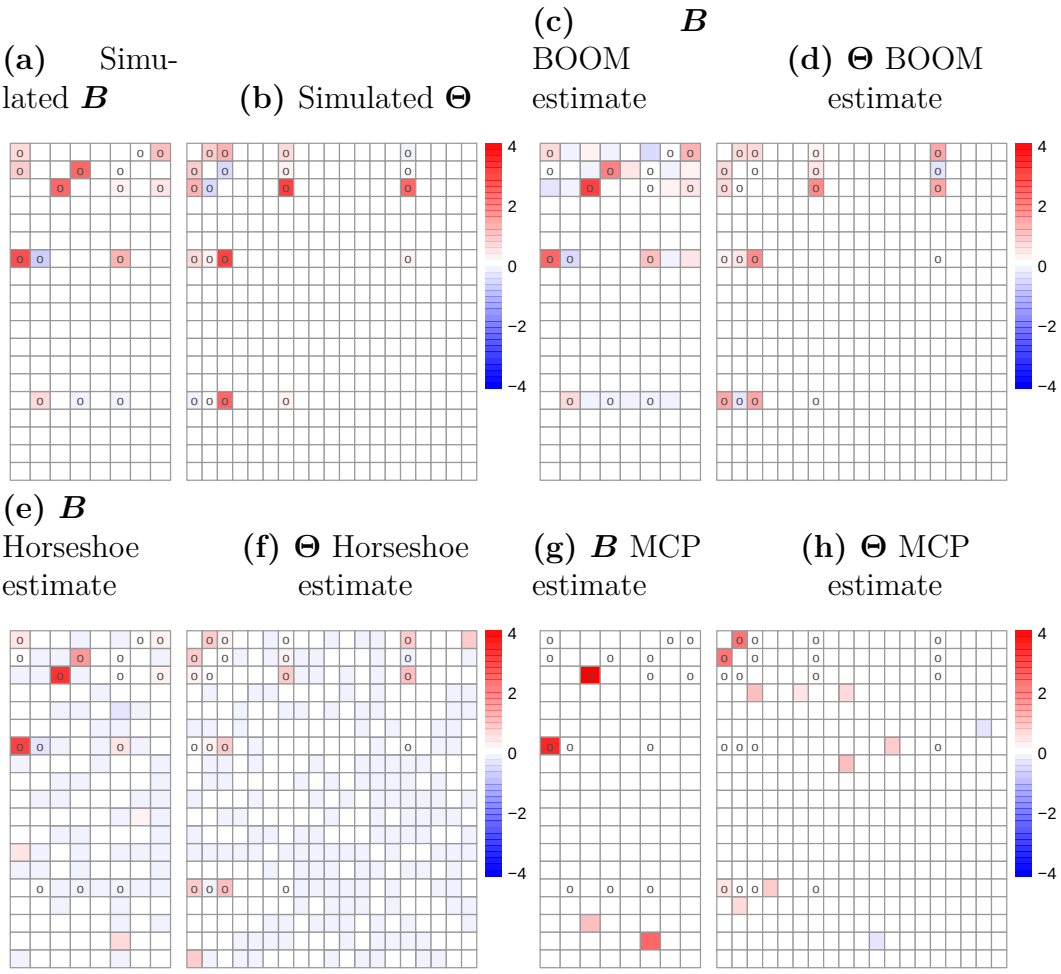
Figure 3.6: Heatmap with the percentage the prior falls within the interquartile range of every coefficient.



While the aforementioned analysis suggests (a) the dataset being supportive of our model assumptions; and (b) we offer better predictive inference than ordinary horseshoe, the small sample size deters BOOM from identifying activated regions reliably. Therefore we simulate 150 new observations from the original 26 observations. To simulate each predictor for a newly generated observation, we first randomly select a predictor matrix $\boldsymbol{A}_{i'}$ from the original 26 observations. The (p, p')-th upper triangular entry of the predictor matrix $\boldsymbol{A}$ for a new observation is simulated from a mixture model $0.8\delta_{a_{i',(p,p')}} + 0.2N(0, 1)$, where δ_x represents the Dirac-delta function. GM predictor for the new observation is simulated similarly from the GM predictor corresponding to the i'-th $\boldsymbol{g}_{i'}$ in the original data. We then construct sparse predictor coefficients $\boldsymbol{\Theta}$ and $\boldsymbol{B}$ keeping 5 randomly selected regions as activated. For this illustration, we choose the first, second, third, seventh and fifteenth regions to be activated with 3 voxels in each region randomly selected to be influential. The simulated coefficients $\boldsymbol{B}$ and $\boldsymbol{\Theta}$ can be observed in Figures 3.7a and 3.7b. Here circles indicate the non-zero entries of each matrix and colors represent the magnitude of each simulated coefficient.

Similar to simulation studies, we implement BOOM along with Horseshoe and MCP on this dataset. Both BOOM and Horseshoe run for 10,000 MCMC iterations out of which the last 5,000 are used to draw inference. Figures 3.7c and 3.7d show that BOOM accurately estimates the sparsity pattern as well as the coefficients in each cell of $\boldsymbol{\Theta}$. This is also corroborated by the TPR and TNR (TNR is defined as $1-$ FPR) values of identifying influential ROIs, and the MSE values in estimating the coefficients, as shown in Table 3.7 for BOOM. Since BOOM employs continuous Horseshoe shrinkage prior to estimate coefficients corresponding to each voxel of GM, it is not expected to recover the sparsity pattern of voxels within each ROI of GM coefficient. Nonetheless, it provides accurate estimation of

Figure 3.7: Randomly generated $\boldsymbol{B}$ and $\boldsymbol{\Theta}$ matrices. Circles indicate the non-zero coefficients.



GM coefficients as demonstrated in Table 3.7. Importantly, both its competitors yield less accurate point estimation of $\boldsymbol{B}$ and $\boldsymbol{\Theta}$. Additionally, they offer significantly lower values for either TPR or TNR in identifying the influential ROIs. We also compare the uncertainty in estimating the coefficients $\boldsymbol{B}$ and $\boldsymbol{\Theta}$ by the Bayesian competitors. While the average coverage is the same for both BOOM and Horseshoe, the length of the intervals is considerably smaller for BOOM, indicating more precise uncertainty quantification by BOOM.

Finally, as in our simulation studies, the predictive performance of the com-

Table 3.7: Mean Squared Error (MSE), average coverage and average length of 95% credible intervals for BOOM, Horseshoe (HS) and MCP are presented for the perturbed PPA data. Best results are presented in boldface.

			MSE			Avg. Coverage of 95% CI			Avg. length of 95% CI		
Method	TPR	TNR	$\boldsymbol{B}$	$\boldsymbol{\Theta}$	All	$\boldsymbol{B}$	$\boldsymbol{\Theta}$	All	$\boldsymbol{B}$	$\boldsymbol{\Theta}$	All
BOOM	**1**	**1**	**0.018**	**0.035**	**0.027**	**1**	**0.976**	**0.987**	**0.288**	**0.078**	**0.177**
Horseshoe	**1**	0.285	0.042	0.063	0.053	**1**	**0.976**	**0.987**	0.732	0.684	0.707
MCP	0.2	0.928	0.146	0.139	0.142	–	–	–	–	–	–

Table 3.8: Mean Squared Predictive Error (MSPE), average coverage and average length of 95% predictive intervals for BOOM, Horseshoe (HS) and MCP are presented for the perturbed PPA data. Best results are presented in boldface.

Method	MSPE	Coverage of the 95% PI	Length of the 95% PI
BOOM	**7.807**	0.8	**7.584**
Horseshoe	15.024	**0.91**	12.339
MCP	35.068	–	–

petitors are assessed based on 100 out of sample observations. Table 3.8 presents the MSPE, coverage and length of 95% predictive intervals for all competitors. Table 3.8 shows significantly smaller MSPE for BOOM than its competitors. While Horseshoe has little better predictive coverage than BOOM, it achieves so with twice the length of the 95% predictive interval. Since MCP is a frequentist competitor, we do not provide estimate of predictive uncertainty from MCP.

3.7 Conclusion

Recent emergence of multi-modal imaging data mandates development of new models which can simultaneously exploit topology of multiple objects and the linked information between the objects in a regression framework. Motivated by such multi-modal data, we develop BOOM approach in this chapter which is among the first Bayesian regression approaches with a scalar response and multi-object predictors, including a network predictor. The framework is illustrated

with extensive simulation studies which allows drawing Bayesian inference on important ROIs significantly associated with the response and offers predictive inference on the same.

Chapter 4

A Bayesian Covariance Based Clustering for High Dimensional Tensors

4.1 Introduction

Unlike Chapters 2 and 3 which develop regression framework for structured data, Chapter 4 addresses the problem of unsupervised clustering of multidimensional arrays or tensors. 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 psychotropic's. 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, 2021; 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 chapter 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. Loss-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 (2002a); 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 chapter.

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., 2021; 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 chapter 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 the 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 differences 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 multivariate 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 a more 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 works by identifying clusters through the 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.

The rest of the chapter evolves as follows. In section 4.2 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 4.3. Empirical evaluations with simulation studies and a real data analysis are presented in Sections 4.4 and 4.5, respectively. Finally, we

conclude in Section 4.6.

4.2 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.

4.2.1 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) $\boldsymbol{T} \in \mathbb{R}^{p_1 \times ... \times p_K}$ with its $(i_1, ..., i_K)$-th element denoted by $T_{i_1,...,i_K}$. When $K = 1$, the tensor reduces to a vector and when $K = 2$, the tensor is a matrix. The vec($\boldsymbol{T}$) operator applied to a tensor $\boldsymbol{T}$ stacks elements into a column vector of dimension $p = \prod_{k=1}^{K} p_k$ with $T_{i_1,...,i_K}$ mapped to the j-th entry of $vec(\boldsymbol{T})$, 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 $\boldsymbol{T}$ each with dimension $p_k \times 1$. The k-mode matricization of a tensor transforms a tensor into a matrix $\boldsymbol{T}_{(k)} \in \mathbb{R}^{p_k \times \frac{p}{p_k}}$, 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 $\boldsymbol{T} \in \mathbb{R}^{p_1 \times \ldots \times p_K}$ and a compatible matrix $\boldsymbol{A} \in \mathbb{R}^{J \times p_k}$, will result in a tensor $\boldsymbol{T} \times_k \boldsymbol{A} \in \mathbb{R}^{p_1 \times \ldots \times p_{k-1} \times J \times p_{k+1} \times \ldots p_K}$, where each element is the product of mode-k fiber of $\boldsymbol{T}$ multiplied by $\boldsymbol{A}$. 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. For a list of matrices $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_K$ with compatible sizes $A_k \in \mathbb{R}^{J_k \times p_k}$ we define the product $\boldsymbol{T} \times [\boldsymbol{A}_1, \ldots, \boldsymbol{A}_K] = \boldsymbol{T} \times_1 \boldsymbol{A}_1 \times_2 \ldots \times_K \boldsymbol{A}_K \in \mathbb{R}^{J_1 \times \ldots \times J_K}$. Thus, when $\boldsymbol{A}_1, \ldots, \boldsymbol{A}_K$ are square matrices, the resulting tensor is of the same dimension as $\boldsymbol{T}$. In what follows, we will use $||\cdot||_F$ to denote the Frobenius norm of the tensor $\boldsymbol{T}$ given by $||\boldsymbol{T}||_F := \sqrt{\sum_{i_1,\ldots,i_K} T^2_{i_1,\ldots,i_K}}$. Finally, we denote as $\otimes_{k=1}^K \boldsymbol{A}_k$ as the sequential Kronecker product of the matrices, that is $\otimes_{k=1}^K \boldsymbol{A}_k = \boldsymbol{A}_1 \otimes \ldots \otimes \boldsymbol{A}_K$.

4.2.2 Bayesian Model-based Tensor Clustering Approach

Let $\boldsymbol{T}_i$ 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

$$\begin{aligned} \pi(\mathcal{C}|\boldsymbol{T}_1, .., \boldsymbol{T}_n) &\propto \pi(\mathcal{C}) \prod_{h=1}^{n(\mathcal{C})} \left[\int \prod_{i \in \mathcal{C}_h} f(\boldsymbol{T}_i|\boldsymbol{\Theta}_h) \pi(\boldsymbol{\Theta}_h) d\boldsymbol{\Theta}_h \right] \\ &= \pi(\mathcal{C}) \prod_{h=1}^{n(\mathcal{C})} m(\{\boldsymbol{T}_i : i \in \mathcal{C}_h\}), \end{aligned} \tag{4.1}$$

where $f(\boldsymbol{T}_i|\boldsymbol{\Theta}_h)$ denotes the likelihood for a tensor observation belonging to the h-th cluster with the cluster-specific model parameter $\boldsymbol{\Theta}_h$ and $\pi(\boldsymbol{\Theta}_h)$ corresponds to the prior distribution on the parameter $\boldsymbol{\Theta}_h$. The quantity $m(\{\boldsymbol{T}_i : i \in \mathcal{C}_h\}) = \int \prod_{i \in \mathcal{C}_h} f(\boldsymbol{T}_i|\boldsymbol{\Theta}_h) \pi(\boldsymbol{\Theta}_h) d\boldsymbol{\Theta}_h$ 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 $\boldsymbol{c} = (c_1, ..., c_n)'$, 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 $\boldsymbol{c} = (c_1, ..., c_n)'$.

A natural choice for the likelihood $f(\boldsymbol{T}_i|\boldsymbol{\Theta}_h)$ 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

$$f(\boldsymbol{T}_i|\boldsymbol{M}_h, \boldsymbol{\Sigma}_{1,h}, \ldots, \boldsymbol{\Sigma}_{K,h}) = (2\pi)^{-\frac{p}{2}} \left\{ \prod_{k=1}^{K} |\boldsymbol{\Sigma}_{k,h}|^{-\frac{p}{2p_k}} \right\} \times \exp\left(-\frac{1}{2} \left\| (\boldsymbol{T}_i - \boldsymbol{M}_h) \times [\boldsymbol{\Sigma}_{1,h}^{-\frac{1}{2}}, \ldots, \boldsymbol{\Sigma}_{K,h}^{-\frac{1}{2}}] \right\|_F^2 \right), \tag{4.2}$$

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,* $\boldsymbol{M}_h = \boldsymbol{0}$ *for all* $h = 1, .., n(\mathcal{C})$.

According to the likelihood specification in (4.2) 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 $f(\boldsymbol{T}_i|\boldsymbol{\Theta}_h)$, as given in (4.2), does not yield a closed form integral for the marginal distribution in (4.1). The common practice is to begin with the distribution $(\boldsymbol{T}_i|\boldsymbol{\Theta}_h, c_i = h) \sim f(\boldsymbol{T}_i|\boldsymbol{\Theta}_h)$ and develop a Gibbs sampler to draw posterior samples of $\boldsymbol{c}$ 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 4.4. Next section develops an approximate Bayesian clustering algorithm that offers remedies to both these challenges simultaneously.

4.2.3 A Covariance-Based Bayesian Tensor Clustering Approach

To avoid complications due to model based clustering of high-dimensional tensor observations, we propose a two-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 $\mathcal{A}(\boldsymbol{T}_i)$ be the set of extracted features from tensor $\boldsymbol{T}_i$ 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 4.2.4 details out a specific choice of such transformed features. While the exact distribution of $\mathcal{A}(\boldsymbol{T}_i)$ is determined by the tensor normal specification given in (4.2), we focus on a reasonable approximation of the distribution for $\mathcal{A}(\boldsymbol{T}_i)$ in our goal to cluster these transformed features. Let $\tilde{f}(\mathcal{A}(\boldsymbol{T}_i)|\tilde{\boldsymbol{\Theta}}_h, \tilde{\boldsymbol{\Theta}}_a)$ be the approximated distribution of $\mathcal{A}(\boldsymbol{T}_i)$ in the h-th cluster, with $\tilde{\boldsymbol{\Theta}}_h$ as its h-th cluster-specific parameter and $\tilde{\boldsymbol{\Theta}}_a$ an auxiliary lower dimensional parameter common across all clusters. Let $\tilde{\pi}_h(\tilde{\boldsymbol{\Theta}}_h)$ and $\tilde{\pi}_a(\tilde{\boldsymbol{\Theta}}_a)$ denote the prior distribution of $\tilde{\boldsymbol{\Theta}}_h$ and $\tilde{\boldsymbol{\Theta}}_a$, respectively, for $h = 1, ..., H$. We choose $\tilde{f}(\cdot)$ and $\tilde{\pi}_h(\cdot)$ to ensure closed form marginal distribution of $\tilde{m}(\{\mathcal{A}(\boldsymbol{T}_i) : i \in \mathcal{C}_h\}|\tilde{\boldsymbol{\Theta}}_a) = \int \prod_{i \in \mathcal{C}_h} \tilde{f}(\mathcal{A}(\boldsymbol{T}_i)|\tilde{\boldsymbol{\Theta}}_h, \tilde{\boldsymbol{\Theta}}_a)\tilde{\pi}_h(\tilde{\boldsymbol{\Theta}}_h)d\tilde{\boldsymbol{\Theta}}_h$.

With closed form marginals for TFs in each cluster, the posterior distribution

of clusters and the auxiliary parameters is given by,

$$\pi(\mathcal{C}, \tilde{\boldsymbol{\Theta}}_a \mid \mathcal{A}(\boldsymbol{T}_1), ..., \mathcal{A}(\boldsymbol{T}_n)) = \pi(\mathcal{C})\tilde{\pi}_a(\tilde{\boldsymbol{\Theta}}_a) \prod_{h=1}^{n(\mathcal{C})} \tilde{m}(\{\mathcal{A}(\boldsymbol{T}_i) : i \in \mathcal{C}_h\}|\tilde{\boldsymbol{\Theta}}_a), \quad (4.3)$$

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, 1973a; 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 $\mathcal{A}(\boldsymbol{T}_i)$ after integrating out the weights of the mixing components. With the posterior distribution of partitions given in (4.3), the computation proceeds through a Chinese restaurant sampler described below (Lau and Green, 2007a).

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 $\mathcal{A}(\boldsymbol{T}_i)$ 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:

$$\tilde{m}(\mathcal{A}(\boldsymbol{T}_i)|\tilde{\boldsymbol{\Theta}}_a) \times \frac{\pi(\{\mathcal{C}_{1,-i}, ..., \mathcal{C}_{n(\mathcal{C}_{-i}),-i}, \{i\}\})}{\pi(\{\mathcal{C}_{1,-i}, ..., \mathcal{C}_{n(\mathcal{C}_{-i}),-i}\})}, \tag{4.4}$$

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:

$$\begin{aligned} &\frac{\tilde{m}(\{\mathcal{A}(\boldsymbol{T}_s) : s \in \{\{i\} \cup \mathcal{C}_{j,-i}\}\}|\tilde{\boldsymbol{\Theta}}_a)}{\tilde{m}(\{\mathcal{A}(\boldsymbol{T}_s) : s \in \mathcal{C}_{j,-i}\}|\tilde{\boldsymbol{\Theta}}_a)} \\ &\times \frac{\pi(\{\mathcal{C}_{1,-i}, ..., \mathcal{C}_{j,-i} \cup \{i\}, \ldots, \mathcal{C}_{n(\mathcal{C}_{-i}),-i}\})}{\pi(\{\mathcal{C}_{1,-i}, ..., \mathcal{C}_{n(\mathcal{C}_{-i}),-i}\})} \end{aligned} \tag{4.5}$$

(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 $\tilde{\boldsymbol{\Theta}}_a$:* Draw s-th iterate of $\tilde{\boldsymbol{\Theta}}_a$ from its full conditional distribution derived from (4.3).

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 $\mathcal{A}(\boldsymbol{T}_i)$ of the tensor $\boldsymbol{T}_i$, 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.

4.2.4 Transformed Features and Their Distributions

This section discusses the specific choice of transformed features $\mathcal{A}(\boldsymbol{T})$ and the approximate distribution $\tilde{f}(\mathcal{A}(\boldsymbol{T})|\tilde{\boldsymbol{\Theta}}_h, \tilde{\boldsymbol{\Theta}}_a)$ of the transformed features used in this chapter. For clustering of high dimensional tensors, we propose to work with the collection of transformed features given by $\mathcal{A}(\boldsymbol{T}_i) = \{\frac{p_k}{p}\boldsymbol{T}_{i,(k)}\boldsymbol{T}'_{i,(k)} : k = 1, ..., K\}$, where $\boldsymbol{T}_{i,(k)}$ is the k-th mode matrix of the tensor $\boldsymbol{T}_i$. Therefore, given a k-way tensor observation $\boldsymbol{T}_i$ 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 4.2.1. *Let* $\boldsymbol{T}_i \sim TN(\boldsymbol{0}, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K)$ *and* $\mathcal{A}(\boldsymbol{T}_i)^{(k)} = \frac{p_k}{p}\boldsymbol{T}_{i,(k)}\boldsymbol{T}'_{i,(k)}$. *Assume that for all* $k = 1, ..., K$, *(i)* $\frac{p_k}{p} \to 0$ *(ii)* $\frac{p_k}{p} tr(\otimes_{k' \neq k}\boldsymbol{\Sigma}_{k'}) \to w_k$ *and (iii)* $\frac{p_k^2}{p^2}\sum_{l,r}\{\otimes_{k' \neq k}\boldsymbol{\Sigma}_{k'}\}_{l,r} \to 0$, *for all* $l, r = 1, ..., p/p_k$, *where* $\{\otimes_{k' \neq k}\boldsymbol{\Sigma}_{k'}\}_{l,r}$ *denotes the* (l, r)*th entry of the matrix* $\otimes_{k' \neq k}\boldsymbol{\Sigma}_{k'}$. *(i)-(iii) together imply that* $\{\mathcal{A}(\boldsymbol{T}_i)^{(k)}\}_{l,r} \to \{\boldsymbol{\Sigma}_k\}_{l,r}w_k$, *where* w_k *is a constant.*

The proof of Lemma 4.2.1 is provided in appendix C. 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 4.2.1. 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 $\boldsymbol{\Sigma}_k$ is an identity

matrix of dimension $p_k \times p_k$, (ii) and (iii) are trivially satisfied with $w_k = 1$ for all $k = 1, ..., K$. They are also found to hold in other non-trivial cases such as in the Toeplitz structured covariance matrices. Broadly, the conditions (ii) and (iii) assumes sparsity in the mode-specific covariance matrices which turn out to be crucial in dictating the clustering performance of the approach.

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 $\mathcal{A}(\boldsymbol{T}_i)^{(k)}$ in all clusters and for all modes $k = 1, ..., K$. More specifically, the (l, r)-th entry of $\mathcal{A}(\boldsymbol{T}_i)^{(k)}$ is modeled as

$$\{\mathcal{A}(\boldsymbol{T}_i)^{(k)}\}_{l,r} \overset{ind.}{\sim} N(\theta^{(k)}_{l,r,h}, \sigma^2), \text{ for } i \in \mathcal{C}_h, \ \theta^{(k)}_{l,r,h} \sim N(\theta_0, \sigma^2/\phi), \ l < r. \tag{4.6}$$

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

The specification of (4.6) leads to a closed form marginal distribution of $\mathcal{A}(\boldsymbol{T}_i)$ in each cluster conditional on the auxiliary parameters $\tilde{\boldsymbol{\Theta}}_a = (\sigma^2, \phi)'$ by integrating out cluster specific parameters $\tilde{\boldsymbol{\Theta}}_h = (\theta^{(k)}_{l,r,h} : l < r)'$. More specifically,

$$\tilde{m}\left(\{\{\mathcal{A}(\boldsymbol{T}_i)^{(k)}\}_{l,r} : i \in \mathcal{C}_h\}|\phi, \sigma^2\right) = \left(2\pi\sigma^2\right)^{\frac{-n_h}{2}} \left[\frac{\phi}{n_h + \phi}\right]^{\frac{1}{2}}$$
$$\times \exp\left\{-\frac{1}{2\sigma^2}\left(\left[\sum_{i \in \mathcal{C}_h}\left(\{\mathcal{A}(\boldsymbol{T}_i)^{(k)}\}_{l,r} - \{\bar{\mathcal{A}}(\boldsymbol{T})^{(k)}_{\mathcal{C}_h}\}_{l,r}\right)^2\right] + \phi\left(\{\bar{\mathcal{A}(\boldsymbol{T})}^{(k)}_{\mathcal{C}_h}\}_{l,r} - \theta_0\right)^2\right)\right\}, \tag{4.7}$$

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

$$\tilde{m}\left(\mathcal{A}(\boldsymbol{T}_1), ..., \mathcal{A}(\boldsymbol{T}_n)|\phi, \sigma^2\right) = \prod_{h=1}^{n(\mathcal{C})}\prod_{i\in\mathcal{C}_h}\prod_{k=1}^{K}\prod_{1\le l<r\le p_k}\tilde{m}\left(\{\{\mathcal{A}(\boldsymbol{T}_i)^{(k)}\}_{l,r} : i\in\mathcal{C}_h\}|\phi, \sigma^2\right), \tag{4.8}$$

where the form of $\tilde{m}\left(\{\{\mathcal{A}(\boldsymbol{T}_i)^{(k)}\}_{l,r} : i\in\mathcal{C}_h\}|\phi, \sigma^2\right)$ is obtained from (4.7).

While Section 4.2.3 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 (2007a), the prior distribution on the partition $\mathcal{C}$ under such a specification assumes the form,

$$\pi(\mathcal{C}|\phi) = \phi^{n(\mathcal{C})+1}\frac{\Gamma(\phi)}{\Gamma(n+\phi)}\prod_{h=1}^{n(\mathcal{C})}\Gamma(n_h), \tag{4.9}$$

with the prior being dependent on the auxiliary parameter ϕ. Following the Chinese Restaurant analogy, (4.9) 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 $1/F$.

4.2.5 Point Estimation and Uncertainty Quantification in Clustering

While we will explore the posterior distribution of partitions through MCMC-based sampling algorithms (see Section 4.3 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 (2007a); 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 (4.8), 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.

$$\sum_{h=1}^{n(\mathcal{C})} \sum_{k=1}^{K} \sum_{1 \le l < r \le p_k} \left\{ \sum_{i \in \mathcal{C}_h} ||\{\mathcal{A}(\boldsymbol{T}_i)^{(k)}\}_{l,r} - \{\bar{\mathcal{A}}(\boldsymbol{T})^{(k)}_{\mathcal{C}_h}\}_{l,r}||^2 + \phi \left(\theta_0 - \{\bar{\mathcal{A}}(\boldsymbol{T})^{(k)}_{\mathcal{C}_h}\}_{l,r}\right)^2 \right\} \tag{4.10}$$

With little algebra, equation (4.10) can be rewritten as

$$\sum_{h=1}^{n(\mathcal{C})} \sum_{k=1}^{K} \sum_{1 \le l < r \le p_k} \left\{ \sum_{i \in \mathcal{C}_h} ||\{\mathcal{A}(\boldsymbol{T}_i)^{(k)}\}_{l,r} - \{\bar{\mathcal{A}}(\boldsymbol{T})^{(k)}_{\mathcal{C}_h,0}\}_{l,r}||^2 + \frac{n_h \phi}{n_h + \phi} \left(\theta_0 - \{\bar{\mathcal{A}}(\boldsymbol{T})^{(k)}_{\mathcal{C}_h,0}\}_{l,r}\right)^2 \right\}, \tag{4.11}$$

where $\{\bar{\mathcal{A}}(\boldsymbol{T})^{(k)}_{\mathcal{C}_h,0}\}_{l,r} = \frac{1}{|\mathcal{C}_h|} \sum_{i \in \mathcal{C}_h} \{\mathcal{A}(\boldsymbol{T}_i)^{(k)}\}_{l,r}$. Notably, the objective function in (4.11) 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 $\mathcal{A}(\boldsymbol{T}_i)$ 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 $\boldsymbol{c} = (c_1, ..., c_n)'$ 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 $\boldsymbol{G}$ (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 $\boldsymbol{G}$ matrix can be used to identify which pair of units are more certain/uncertain to belong to the same cluster.

4.3 Posterior Computation

With likelihood and prior distributions specified as in Section 4.2.4, the full posterior distribution of partitions and auxiliary variables is given by,

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

The posterior computation proceeds following the general algorithm described in Section 4.2.3 with simplifications due to the prior structure. Specifically, the

probability of assigning the i-th observation to a new cluster, described in (4.4), reduces to

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

On the other hand, the probability of being assigned to the existing j-th cluster $\mathcal{C}_{j,-i}$, described in (4.5), takes the form

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

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

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

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

4.4 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 $\boldsymbol{T}_1$,...,$\boldsymbol{T}_n$ from a finite mixture of tensor normal models with H mixing components given by

$$\boldsymbol{T}_i \sim \sum_{h=1}^{H} \pi_h TN(\boldsymbol{0}, \boldsymbol{\Sigma}_{1,h}, ..., \boldsymbol{\Sigma}_{K,h}), \ \sum_{h=1}^{H} \pi_h = 1. \tag{4.12}$$

The data generation scheme ensures that the tensors in different clusters 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 $\boldsymbol{\Sigma}_{k,h}^{-1}$ 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 $\boldsymbol{E}$ 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 $\boldsymbol{D} = \boldsymbol{E}/2 + \delta \boldsymbol{I}$ where $\boldsymbol{I}$ is the identity and δ is chosen so that $\boldsymbol{D}$ 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 $\boldsymbol{D}$.

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.

4.4.1 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. Further, we acknowledge the fact that the DEEM algorithm does not necessarily fix the mean

of every cluster at zero. To be fair to the DEEM competitor, we implement a modified version of DEEM which simplifies the enhanced expectation step of the algorithm by setting the means as known and equal to zero. We then implement the enhanced Maximization step considering different within cluster covariances as necessary for our set-up. Although the implementation of DEEM in (Mai et al., 2021b) has the same within cluster covariance, their method is not restricted to this set up. We then initialize the cluster memberships by performing k-means clustering on the vectorized tensors as suggested in (Mai et al., 2021b). We run the algorithm for 100 iterations, noticing that cluster membership stabilizes around 20 iterations. For DTC we use the code shared by the authors in its default setting, running a first step to determine the tuning parameters, followed by running the second step of clustering. We fix the number of clusters at their true values.

Finally, we also employ (4.12) after fixing the true number of clusters and the true values of $\boldsymbol{\Sigma}_{k,h}$'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 $\hat{\boldsymbol{c}}$, and (ii) a heatmap of the posterior probability of any two samples belonging to the same cluster, or the co-clustering matrix $\boldsymbol{G}$ 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 $\boldsymbol{G}$ can be obtained from the post burn-in MCMC samples of the cluster membership indices $\boldsymbol{c}$.

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.

4.4.2 Simulation Results

Table 4.1 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 4.1 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

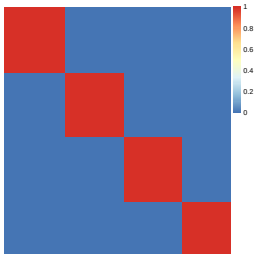
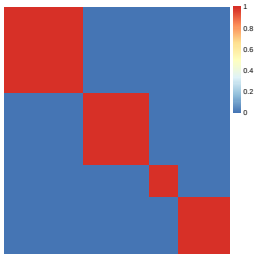
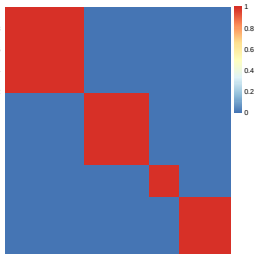
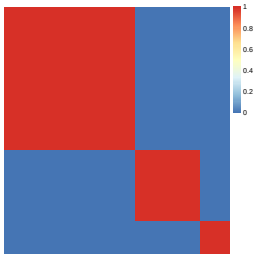
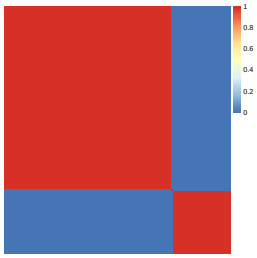
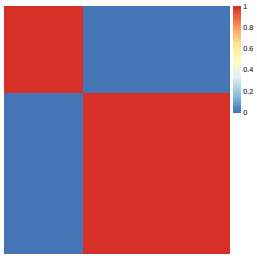
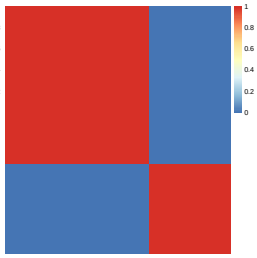
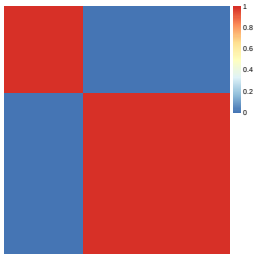
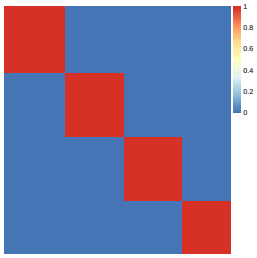
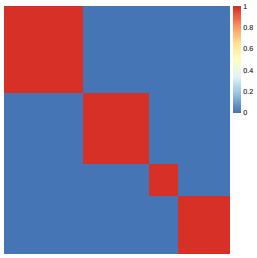
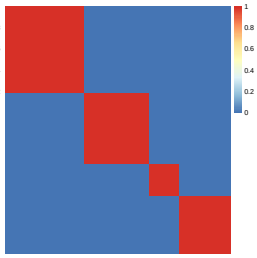
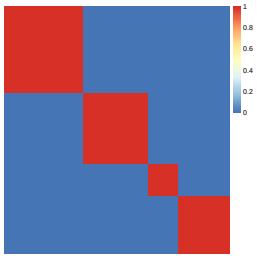
clustering performance of DEEM. Note that DTC clusters based on the low-rank decomposition of the mean structure of each tensor which is not conducive in capturing in the present scenario, since data generating clusters mainly differ in terms of their variability. In fact, the tensors simulated from (4.12) 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 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.

Table 4.1: 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

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

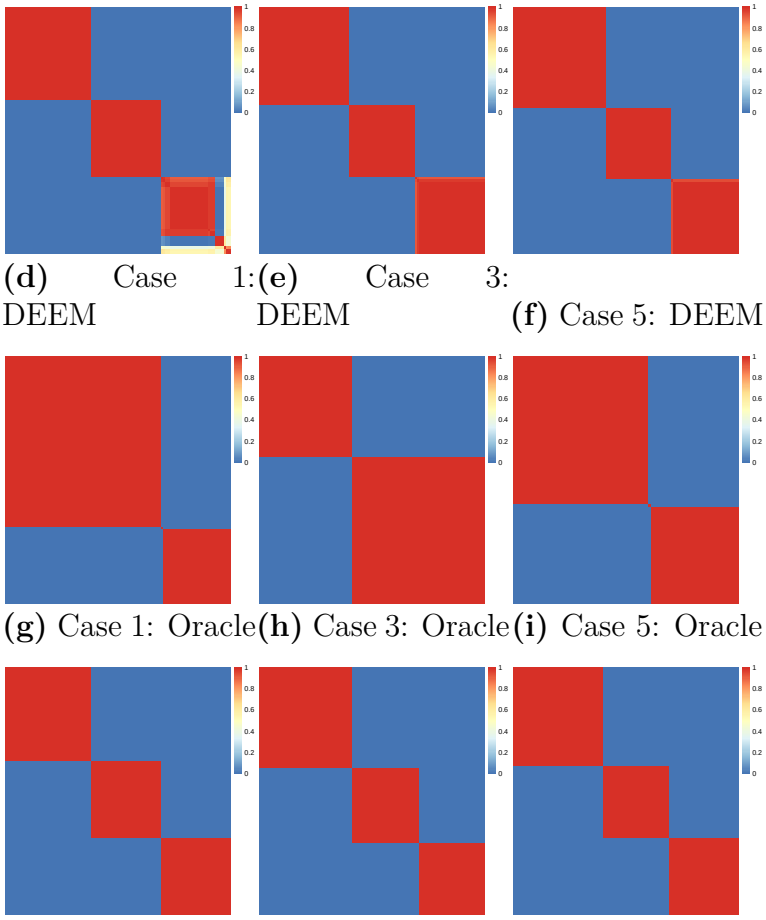
Figure 4.1: Heatmap of the posterior probability of any two samples belonging to the same cluster. For the cases with $H = 4$.

$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$	$\alpha = 0.4$
(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
			

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

Figure 4.2: Heatmap of the posterior probability of any two samples belonging to the same cluster. For the cases with $H = 3$.

$\alpha = 0.1$	$\alpha = 0.2$	$\alpha = 0.3$



(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

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.

4.5 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. (2020). The EEG data for 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 EGG 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.

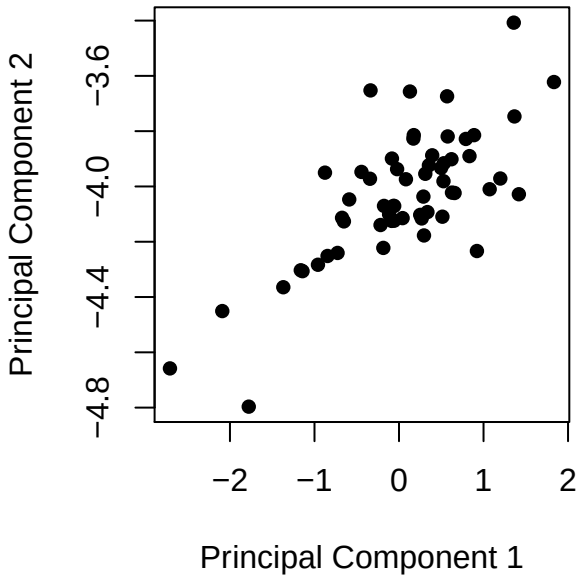
Prior evidence suggests patients with autism spectrum disorder (ASD) can be clustered based on EEG recordings with substantial heterogeneity in cluster-specific mean and covariance structures Hasenstab et al. (2016). In a previous analysis of our motivating alpha spectral density EEG data, Scheffler et al. (2020) 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 encourages such a structure. While it is hard to verify such an assumption in high dimensional objects, we present two separate exploratory analyses to investigate this issue on this dataset. We conduct an exploratory analysis by performing Principal Component Analysis (PCA) of the data matrices. Figure 4.3 presents a plot for the first two principal components, which account for 42.39% of the total variability in the observations, here we can also see that no clustering is apparent, with the first two principal components for observations being 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 each coordinate clustering by computing the ARI of every coordinate clustering against every other coordinate clustering, resulting in a possible $\binom{825}{2}$ ARI values. We perform this analysis for k-means with $2, 3, 4$ and 5 clusters. Table 4.2 presents the 5th, 25th, 50th, 75th and 95th

Figure 4.3: Observations visualizations: we present the first two principal components after performing Principal Component Analysis.



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. In fact, even for the 95th percentile, the level of similarity is still very low, and it becomes even lower as we increase the number of clusters.

Table 4.2: 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 was run for 400 iterations total, of which we used 100 as burn-in period. The posterior distribution of the number of clusters in Figure 4.4b shows a clear mode at 3, indicating three clusters among subjects. The co-clustering matrix shown in Figure 4.4a indicates four clusters with a high degree of uncertainty among the elements in the second and in the first cluster. Indeed, the result indicates that the elements in the second cluster are often included as part of the first cluster, which is consistent with the posterior mode of the number of clusters being identified as three. In Figure 4.4c 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 4.5) 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 very close to 1. 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 some 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.

Figure 4.4: Cluster structure for EEG data on 58 ASD children.

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

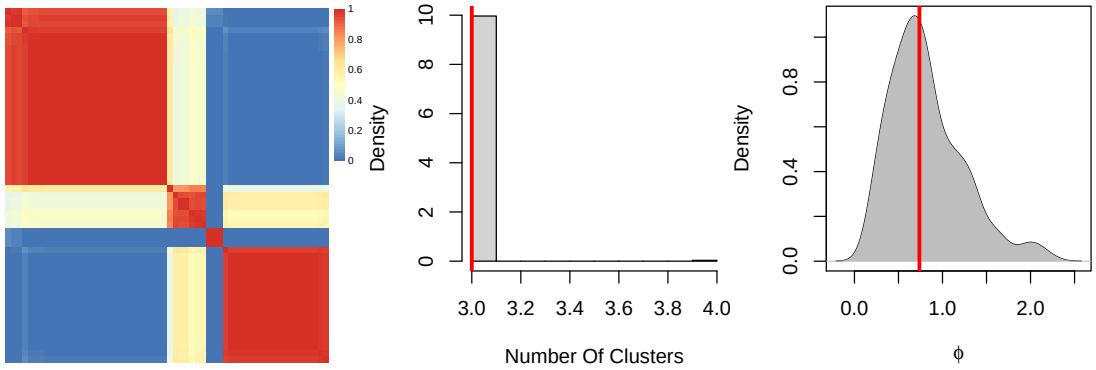
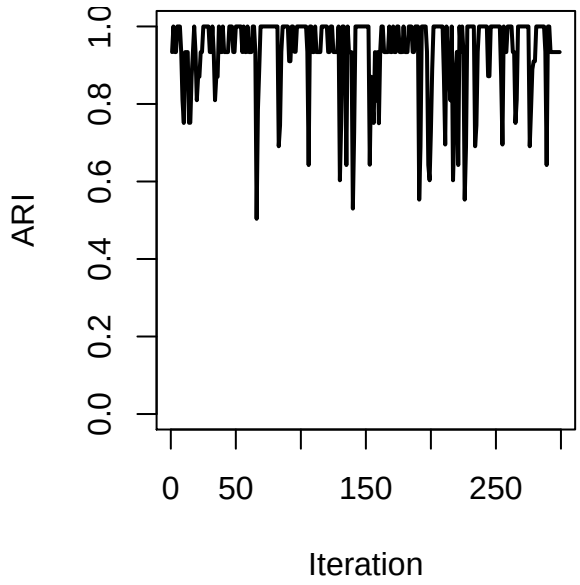


Figure 4.5: ARI of each partition with respect to the previous partition throughout the 300 MCMC iterations sequentially.



We also perform a full Bayesian mixture model analysis of the data using matrix normal distributions with zero mean as mixture components. This approach also attempts to cluster tensors based on their variability, but without

any approximation as in BTC. This approach should ideally offer better clustering performance than DTC, since DTC is essentially is 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 4.6 presents co-clustering matrices for the full Bayesian implementation for a mixture of $K = 3, 4, 5$ matrix normal distributions. The figure demonstrates unsatisfactory performance of the full Bayesian clustering approach, showing only one cluster. This is somewhat expected based on the performance of our competing approach 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 4.3.

Figure 4.6: Cluster structure for EEG data on 58 ASD children using a full Bayesian mixture of matrix normals.

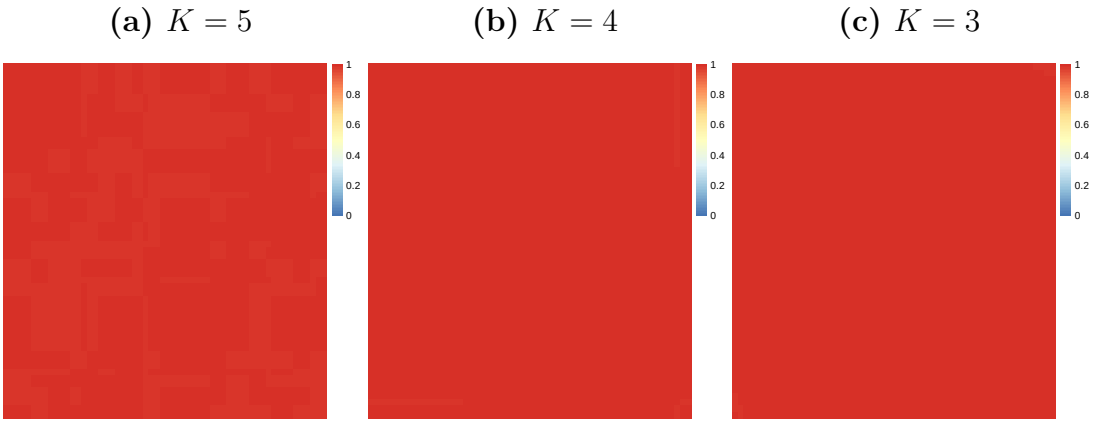


Table 4.3: Runtime (in seconds) for 400 iterations for BTC and Full Bayesian implementation for $K = 5, 4, 3$ for ASD data.

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

4.6 Conclusion

This chapter has studied the problem of clustering high dimensional tensors into subgroups where subgroups differ mainly in their variabilities, rather than in their means. Algorithmic clustering approaches mainly cluster objects differing in their centers, where as the Bayesian model based clustering approaches are too heavy computationally for high dimensional tensors. Moreover, their performance is less than satisfactory when the sample size is also moderate. To overcome these issues, we have proposed a novel approach where we first start with an estimator of the variability for each tensor and then develop model-based clustering of these estimators. The approach is computationally simple, easily scales with big tensors and offers accurate clusters, as well as clustering uncertainties. While there is a plenty of literature on Bayesian mixture model based clustering or algorithmic clustering for scalar, vector or functional data, very few articles have addressed the issue of clustering tensors with high dimensions. This chapter proposes a novel solution to this important problem.

Chapter 5

Future Work

Many future directions emerge from Chapters 2-4 of the thesis. The DFP algorithm in Chapter 2 develops an online approximate Bayesian framework for high-dimensional linear regression with Gaussian errors. However, the scope of the DFP algorithm can be extended well beyond the realm of Gaussian errors. For example, DFP will be employed for high dimensional logistic and probit regressions as part of our future work. While data augmentation schemes (Albert and Chib, 1993; Polson et al., 2013) in high-dimensional binary regression allow Gibbs sampling for parameter blocks, making the DFP formulation natural, they also violate assumptions (1) and (2) in the formulation of DFP in Section 2.3 of Chapter 2, which needs further research to be implemented. We also propose to extend the DFP formulation for high dimensional linear regression with heavy-tailed error distributions. Notably, a heavy-tailed error distribution can often be expressed as a scale mixture of Gaussian errors. Thus, upon using a data augmentation scheme, developing DFP under this model will require extending the DFP framework when the number of parameters increases with the onset of a new data shard. We would also like to extend our theoretical results on the convergence of the DFP kernel to the full posterior from a fixed partitioning set up to an adaptive

dynamic partitioning set up. Finally, this article constructs $\widehat{\boldsymbol{\Theta}}^{(t)}$ as the average of samples of $\boldsymbol{\Theta}$ drawn from the DFP algorithm at time t. It is mentioned that the theory allows alternative constructions of $\widehat{\boldsymbol{\Theta}}^{(t)}$, as long as the sequence converges to the true data generating parameter as $t \to \infty$. As a future exploration, we plan to develop a hybrid DFP algorithm where $\widehat{\boldsymbol{\Theta}}^{(t)}$ is constructed separately by implementing a frequentist high dimensional regression technique (e.g., lasso) at the onset of a new data shard at every time. This will guarantee consistency of $\widehat{\boldsymbol{\Theta}}^{(t)}$ and the purpose of fitting the DFP algorithm then becomes quantifying uncertainty in the posterior distribution of parameters.

Similar to Chapter 2, several future research directions emerge from our multi-object regression framework in Chapter 3. In the first step to our multi-object regression, we only exploit the hierarchical structure between ROIs and voxels in the GM image but did not consider the spatial neighborhood information of voxels within an ROI. In future work, we plan to incorporate spatial information for voxels within every ROI of GM. Such a framework will presumably present computational challenges due to employing Gaussian processes for spatial smoothing. As part of this work, we will employ computationally efficient Gaussian processes to overcome such challenges. Another line of future work may incorporate images from additional sources (e.g., white matter) as predictors to draw more robust inference on influential ROIs.

Finally, the tensor clustering framework in Chapter 4 also opens the door to a number of future research directions. We want to develop the tensor clustering framework when tensors are symmetric as part of future work. As another future work, we will develop clustering of multi-object data. Some of these constitute our ongoing work.

Appendix A

Bayesian Dynamic Feature Partitioning in High-Dimensional Regression with Big Data

A.1 Convergence Behavior of Approximate Samplers

We study convergence behavior for the DFP algorithm provided in Section 2.3 of chapter 2. Since developing results with dynamic partitioning is challenging given that any partitioning scheme exploits specifics of model and prior distributions, the results developed here establish convergence of the DFP algorithm with the assumption that the partitioning of the parameter set is fixed over time. Although this is a restrictive assumption, DFP seems to enjoy desirable asymptotic behavior even under this assumption. With dynamic partitioning of subsets, we expect to witness stronger theoretical results for DFP, which needs separate attention in a future work.

The theoretical development proceeds in a few steps. DFP algorithm being a Markov chain framework assumes a transition kernel (denoted by $T_t(\cdot,\cdot)$ at time t) and a stationary distribution of the transition kernel at each time t (referred to as the DFP stationary distribution and denoted by π_t). At first, we establish the general form of the DFP stationary distribution π_t at each time t. Next, we develop sufficient conditions on the transition kernel ($T_t(\cdot,\cdot)$), no. of samples (S) drawn from the transition kernel at each time t, dynamic evolution of the DFP stationary distributions over time and conditions on the point estimates ($\widehat{\boldsymbol{\Theta}}^{(t)}$) to ensure convergence of the DFP transition kernel to the full posterior distribution asymptotically. Some of these conditions are verified for the specific cases of high dimensional linear regression with shrinkage priors and spike and slab priors. To begin with, we define a few quantities.

A.1.1 Notation and Framework

For the sake of simplicity denote $\boldsymbol{\Theta}_{G_l^t} = \boldsymbol{\Theta}_{l,t} \in \mathbb{R}^{q_l}$ for $l = 1, ..., k_t$ a partition of $\boldsymbol{\Theta}$ into k_t subsets at time t. Since our theoretical exposition fixes partitions over time t, $k_t = k$ and q_l's are not functions of time t and $\sum_{l=1}^{k} q_l = q = dim(\boldsymbol{\Theta})$, which is also fixed across time by Assumption (1) of Section 2.3.1 in chapter 2. Assume $\boldsymbol{\Theta}_{l,t} = (\theta_{l,t,1}, ..., \theta_{l,t,q_l})'$. The full posterior distribution of $\boldsymbol{\Theta}$ at time t, denoted by $f(\boldsymbol{\Theta}|\boldsymbol{S}^{(t)})$ in Section 2.3, is also shortened as $f_t(\boldsymbol{\Theta})$. Assume that the density $f_t(\boldsymbol{\Theta})$ is admitted with respect to the Lebesgue measure ν. $T_t : \mathbb{R}^q \times \mathbb{R}^q \to \mathbb{R}^+$ is a transition kernel at time t having the property that $T_t(\boldsymbol{z}, \cdot)$ is a probability measure for all $\boldsymbol{z} \in \mathbb{R}^q$ and $T_t(\cdot, \boldsymbol{A})$ is a measurable function for all $\boldsymbol{A}$ in the Borel sigma algebra of $\mathbb{R}^q$.

Finally, we denote $\widehat{\boldsymbol{\Theta}}^{(t)}_{-G_l^t}$ and $\widehat{\boldsymbol{\Theta}}^{(t)}_{G_l^t}$ as $\widehat{\boldsymbol{\Theta}}^{(t)}_{-l}$ and $\widehat{\boldsymbol{\Theta}}^{(t)}_{l}$ respectively, $l = 1, ..., k$, for the simplicity of notation.

A.1.2 The DFP transition kernel

It follows from the DFP algorithm that the DFP transition kernel $T_t : \mathbb{R}^{q_1} \times \cdots \times \mathbb{R}^{q_k} \to \mathbb{R}^+$ at time t is given by:

$$T_t(\boldsymbol{\Theta}, \boldsymbol{\Theta}') = \prod_{l=1}^{k} \prod_{i=1}^{q_l} f_t\Big(\theta'_{l,t,i} | \widehat{\boldsymbol{\Theta}}_{-l}^{(t-1)}, \{\theta'_{l,t,j}\}_{j<i}, \{\theta_{l,t,j}\}_{j>i}\Big), \tag{A.1}$$

which represents sequential updating of parameters within subsets. The unique stationary distribution $\pi_t : \mathbb{R}^q \to \mathbb{R}^+$ of the transition kernel T_t at time t is given in the following lemma.

Lemma A.1.1. *DFP approximate kernel T_t has a unique stationary distribution $\pi_t(\boldsymbol{\Theta}) = \prod_{l=1}^{k} f_t(\boldsymbol{\Theta}_l | \widehat{\boldsymbol{\Theta}}_{-l}^{(t-1)})$.*

Proof. In order to prove the lemma, we will simply show that π_t given by the equation above satisfies $\int T_t(\boldsymbol{\Theta}, \boldsymbol{\Theta}')\pi_t(\boldsymbol{\Theta})\, d\boldsymbol{\Theta} = \pi_t(\boldsymbol{\Theta}')$. Note that

$$\begin{aligned}
&\int T_t(\boldsymbol{\Theta}, \boldsymbol{\Theta}')\pi_t(\boldsymbol{\Theta}) d\boldsymbol{\Theta} \\
&\qquad = \int \prod_{l=1}^{k} \Big[\prod_{i=1}^{q_l} f_t(\theta'_{l,t,i} | \widehat{\boldsymbol{\Theta}}_{-l}^{(t-1)}, \{\theta'_{l,t,j}\}_{j<i}, \{\theta_{l,t,j}\}_{j>i}) f_t(\boldsymbol{\Theta}_l | \widehat{\boldsymbol{\Theta}}_{-l}^{(t-1)}) \Big] d\boldsymbol{\Theta} \\
&\qquad = \prod_{l=1}^{k} \int \Big[\prod_{i=1}^{q_l} f_t(\theta'_{l,t,i} | \widehat{\boldsymbol{\Theta}}_{-l}^{(t-1)}, \{\theta'_{l,t,j}\}_{j<i}, \{\theta_{l,t,j}\}_{j>i}) f_t(\boldsymbol{\Theta}_l | \widehat{\boldsymbol{\Theta}}_{-l}^{(t-1)}) \Big] d\boldsymbol{\Theta} \\
&\qquad = \prod_{l=1}^{k} f_t(\boldsymbol{\Theta}'_l | \widehat{\boldsymbol{\Theta}}_{-l}^{(t-1)}).
\end{aligned}$$

Here the last step follows by recognizing that the kernel T_t is a product of various independent Gibbs sampler (or Metropolis Hastings) kernels in different parameter partitions. □

A.1.3 Main convergence results

We will now state a theorem and a corollary. The theorem states reasonable assumptions to ensure decay of the total variation distance between DFP transition kernel and its stationary distribution as t increases. The corollary then adds a few more sufficient conditions to ensure that DFP kernel becomes close to the full posterior distribution as t increases. Let π_0 denote the initial distribution from which parameters are drawn. Suppose T_t^S denotes the kernel corresponding to S draws from the DFP kernel T_t. We use $||\cdot||_{TV}$ to denote the total variation distance and $d_H(\cdot,\cdot)$ to denote the Hellinger distance between two densities. The statement of the theorem is given below.

Theorem A.1.2. *Let $\epsilon \in (0,1)$. Assume $\exists$ a constant $C > 0$, a positive integer S and a function $V : \mathbb{R}^q \to [1,\infty)$ s.t. for all large t,*

(i) $E_{\pi_t}(V^2) \leq C$

(ii) $||T_t(\boldsymbol{\Theta},\cdot)^S - \pi_t||_{TV} \leq V(\boldsymbol{\Theta})\alpha_t^S < 1-\epsilon \;\forall\; \boldsymbol{\Theta}$ *and for some* $\alpha_t \in (0,1)$.

Then,

$$||T_t^S \cdots T_1^S - \pi_t||_{TV} \leq \sum_{s=1}^{t} \epsilon^{t+1-s}\rho_s, \tag{A.2}$$

where $\rho_t = 2\sqrt{C}d_H(\pi_t, \pi_{t-1})$.

The proof of Theorem A.1.2 follows along the same line of the proof of Theorem 3.6 in (Yang and Dunson, 2013) and is thus omitted.

Corollary A.1.2.1. *If conditions (i) and (ii) of Theorem A.1.2 are satisfied and additionally we assume (iii)* $\rho_t \to 0$ *and (iv)* $||\pi_t - f_t||_{TV} \to 0$, *as* $t \to \infty$. *Then* $||T_t^S \cdots T_1^S - f_t||_{TV} \to 0$.

Proof: Using conditions (i), (ii) and (iii), as $t \to \infty$, $||T_t^S \cdots T_1^S - \pi_t||_{TV} \to 0$, following Theorem 0.2. Now we use (iv) to deduce that $||T_t^S \cdots T_1^S - f_t||_{TV} \leq ||T_t^S \cdots T_1^S - \pi_t||_{TV} + ||\pi_t - f_t||_{TV} \to 0$, as $t \to \infty$.

Remark Corollary A.1.2.1 shows that the DFP transition kernel after S draws each at time $1, ..., t$ becomes close to the full posterior distribution f_t at time t. This implies that as time t increases, samples drawn from the DFP full conditional distributions can be taken as the draws from the un-approximated full posterior distribution f_t.

Next, we argue that the assumptions in Theorem A.1.2 and Corollary A.1.2.1 are reasonable. Note that conditions (i) and (ii) refer to the assumption that the DFP transition kernel at time t converges to the DFP stationary distribution at time t at a geometric rate. This assumption is also referred to as the *Geometric Ergodicity* assumption. We first prove that this assumption holds for shrinkage and spike and lasso priors used in this article in Theorems 0.4 and 0.5, respectively. Condition (iii) ensures that the stationary distribution of the approximating kernel changes slowly as time proceeds. This is a mild condition satisfied by any regular parametric model by applying the Bernstein-Von Mises theorem. Finally, we prove condition (iv) under a few regularity assumptions in Lemma 0.6.

We will now proceed to verify Geometric Ergodicty for the DFP kernel with some of the Gaussian scale mixture priors and spike and lasso prior. The theorem below shows conditions for geometric ergodicity under Bayesian lasso prior. The proof uses some of the techniques outlined in Pal et al. (2014).

Theorem A.1.3. *Assume there exists $m_0 > 0$ s.t. $e_{min}(\boldsymbol{S}_{1,\nabla}^{(t)}) \geq m_0$, for any set $\nabla \subseteq \{1, ..., p\}$ and any $t = 1, ..., T$, where $\boldsymbol{S}_{1,\nabla}^{(t)}$ is a submatrix of $\boldsymbol{S}_1^{(t)}$ with columns corresponding to the indices ∇. Then the DFP Bayesian lasso transition kernel is geometrically ergodic.*

Proof. If $T_t((\boldsymbol{\beta}, \boldsymbol{\tau}^2, \sigma^2, \lambda^2), (\boldsymbol{\beta}', (\boldsymbol{\tau}^2)', (\sigma^2)', (\lambda^2)'))$ is the transition kernel of the DFP and $\pi_t(\boldsymbol{\beta}, \boldsymbol{\tau}^2, \sigma^2, \lambda^2)$ is the stationary distribution of the transition kernel, then $T_t(\cdot, \cdot)$ and $\pi_t(\cdot)$ for the Bayesian lasso are given by

$$
\begin{aligned}
&T_t((\boldsymbol{\beta}, \boldsymbol{\tau}^2, \sigma^2, \lambda^2), ((\boldsymbol{\beta})', (\boldsymbol{\tau}^2)', (\sigma^2)', (\lambda^2)') \\
&\quad = \prod_l \prod_{j \in G_l^t} \Big\{ f_t((\beta_j)' | (\tau_j^2), \widehat{\boldsymbol{\beta}}_{-l}^{(t-1)}, \widehat{\boldsymbol{\tau}}_{-l}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) \\
&\qquad \times f_t((\tau_j^2)' | (\beta_j)', \widehat{\boldsymbol{\beta}}_{-l}^{(t-1)}, \widehat{\boldsymbol{\tau}}_{-l}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) \Big\} \\
&\qquad \times f_t((\sigma^2)' | \widehat{\boldsymbol{\beta}}^{(t-1)}, \widehat{\boldsymbol{\tau}}^{2(t-1)}) f_t((\lambda^2)' | \widehat{\boldsymbol{\beta}}^{(t-1)}, \widehat{\boldsymbol{\tau}}^{2(t-1)}) \qquad \text{(A.3)} \\
&\pi_t(\boldsymbol{\beta}, \boldsymbol{\tau}^2, \sigma^2, \lambda^2) \\
&\quad = \prod_{l=1}^{k} \Big\{ f_t(\boldsymbol{\beta}_l, \boldsymbol{\tau}_l^2 | \widehat{\boldsymbol{\beta}}_{-l}^{(t-1)}, \widehat{\boldsymbol{\tau}}_{-l}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) \Big\} \\
&\qquad \times f_t(\sigma^2 | \widehat{\boldsymbol{\beta}}^{(t-1)}, \widehat{\boldsymbol{\tau}}^{2(t-1)}) f_t(\lambda^2 | \widehat{\boldsymbol{\beta}}^{(t-1)}, \widehat{\boldsymbol{\tau}}^{2(t-1)}). \qquad \text{(A.4)}
\end{aligned}
$$

Hence, $||T_t - \pi_t||_{TV} = ||\tilde{T}_{t,1} - \tilde{\pi}_{t,1}||_{TV}$, where

$$
\begin{aligned}
\tilde{T}_{t,1} &= \prod_{j \in G_l^t} \Big\{ f_t((\beta_j)' | (\tau_j^2), \widehat{\boldsymbol{\beta}}_{-l}^{(t-1)}, \widehat{\boldsymbol{\tau}}_{-l}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) \\
&\qquad \times f_t((\tau_j^2)' | (\beta_j)', \widehat{\boldsymbol{\beta}}_{-l}^{(t-1)}, \widehat{\boldsymbol{\tau}}_{-l}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) \Big\} \\
\tilde{\pi}_{t,1} &= \prod_{l=1}^{k} \Big\{ f_t(\boldsymbol{\beta}_l, \boldsymbol{\tau}_l^2 | \widehat{\boldsymbol{\beta}}_{-l}^{(t-1)}, \widehat{\boldsymbol{\tau}}_{-l}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) \Big\}.
\end{aligned}
$$

Thus it is enough to show the geometric ergodicity of the chain by establishing a geometric drift condition and a geometric minorization condition for the $(\boldsymbol{\beta}, \boldsymbol{\tau}^2)$ chain.

<u>*Minorization condition.*</u>

Define, $\tilde{V}_t(\boldsymbol{\beta}, \boldsymbol{\tau}^2) = \frac{\sum_{j=1}^{p} \beta_j^2}{\frac{1}{m_0} \sum_{l=1}^{k} \boldsymbol{H}_l' \boldsymbol{H}_l + 1} + \sum_{j=1}^{p} \tau_j^2$, where $\boldsymbol{H}_l = \boldsymbol{S}_{2,l}^{(t)} - \boldsymbol{S}_{1,l,-l}^{(t)} \widehat{\boldsymbol{\beta}}_{-l}^{(t-1)}$. Let $\mathcal{S}_{\tilde{V}_t, d} = \{(\boldsymbol{\beta}, \boldsymbol{\tau}^2) : \tilde{V}_t(\boldsymbol{\beta}, \boldsymbol{\tau}^2) \leq d\}$. While showing minorization condition, we

will establish that there exists a constant $0 < c(\tilde{V}_t, d) < 1$ depending on $\tilde{V}_t$ and d such that $\tilde{T}_{t,1}((\boldsymbol{\beta}_0, \boldsymbol{\tau}_0^2), (\boldsymbol{\beta}, \boldsymbol{\tau}^2)) \geq c(\tilde{V}_t, d) g(\boldsymbol{\beta}, \boldsymbol{\tau}^2)$ for some density function $g(\cdot)$ for every $(\boldsymbol{\beta}_0, \boldsymbol{\tau}_0^2) \in \mathcal{S}_{\tilde{V}_t, d}$. Denote $\tilde{\lambda} = \widehat{\lambda}^{2(t-1)}$ and $\tilde{\mu}_j = \sqrt{\frac{\hat{\sigma}^{2(t-1)} \widehat{\lambda}^{2(t-1)}}{\beta_{0j}^2}}$. Then

$$
\begin{aligned}
f_t(\tau_j^2 | \boldsymbol{\beta}_0) &= \sqrt{\frac{\tilde{\lambda}}{2\pi}} (\tau_j^2)^{-1/2} \exp\left\{ -\tilde{\lambda} \frac{(1 - \tau_j^2 \tilde{\mu}_j)^2}{2\tilde{\mu}_j^2 \tau_j^2} \right\} \\
&= \sqrt{\frac{\tilde{\lambda}}{2\pi}} (\tau_j^2)^{-1/2} \exp\left\{ -\frac{\tilde{\lambda} \tau_j^2}{2} - \frac{\tilde{\lambda}}{2\tilde{\mu}_j^2 \tau_j^2} + \frac{\tilde{\lambda}}{\tau_j} \right\} \\
&\geq \sqrt{\frac{\tilde{\lambda}}{2\pi}} (\tau_j^2)^{-1/2} \exp\left\{ -\frac{\tilde{\lambda} \tau_j^2}{2} - \frac{\tilde{\lambda}}{2\tilde{\mu}_j^2 \tau_j^2} \right\}. \qquad \text{(A.5)}
\end{aligned}
$$

Note that $\tilde{V}_t(\boldsymbol{\beta}_0, \tau_0^2) \leq d$ implies $\frac{\boldsymbol{\beta}_0' \boldsymbol{\beta}_0}{\frac{1}{m_0} \sum_{l=1}^k \boldsymbol{H}_l' \boldsymbol{H}_l + 1} + \sum_{j=1}^p \tau_j^2 \leq d$ when $(\boldsymbol{\beta}_0, \boldsymbol{\tau}_0^2) \in \mathcal{S}_{\tilde{V}_t, d}$. Thus $\boldsymbol{\beta}_0' \boldsymbol{\beta}_0 \leq d \left[\frac{1}{m_0} \sum_{l=1}^k \boldsymbol{H}_l' \boldsymbol{H}_l + 1 \right]$ when $(\boldsymbol{\beta}_0, \boldsymbol{\tau}_0^2) \in \mathcal{S}_{\tilde{V}_t, d}$. Using this, and the previous inequality (A.5), we have

$$
\begin{aligned}
f_t(\tau_j^2 | \boldsymbol{\beta}_0) &\geq \sqrt{\frac{\widehat{\lambda}^{2(t-1)}}{2\pi}} (\tau_j^2)^{-1/2} \exp\left\{ -\frac{\widehat{\lambda}^{2(t-1)} \tau_j^2}{2} - \frac{d \left[\frac{1}{m_0} \sum_{l=1}^k \boldsymbol{H}_l' \boldsymbol{H}_l + 1 \right]}{2\tau_j^2 \widehat{\sigma}^{2(t-1)}} \right\} \\
&\geq \sqrt{\frac{\widehat{\lambda}^{2(t-1)}}{2\pi}} (\tau_j^2)^{-1/2} \\
&\quad \exp\left\{ -\frac{1}{2} \left(\sqrt{\widehat{\lambda}^{2(t-1)} \tau_j^2} - \sqrt{\frac{d \left[\frac{1}{m_0} \sum_{l=1}^k \boldsymbol{H}_l' \boldsymbol{H}_l + 1 \right]}{\tau_j^2 \widehat{\sigma}^{2(t-1)}}} \right)^2 \right\} \\
&\quad \exp\left\{ -\sqrt{\frac{\widehat{\lambda}^{2(t-1)} d \left[\frac{1}{m_0} \sum_{l=1}^k \boldsymbol{H}_l' \boldsymbol{H}_l + 1 \right]}{\widehat{\sigma}^{2(t-1)}}} \right\}
\end{aligned}
$$

Let $c(\tilde{V}_t, d) = \exp\left\{-\sqrt{\frac{\widehat{\lambda}^{2(t-1)} d\left[\frac{1}{m_0}\sum_{l=1}^{k} \boldsymbol{H}_l' \boldsymbol{H}_l + 1\right]}{\widehat{\sigma}^{2(t-1)}}}\right\}$. Thus

$$
\begin{aligned}
&\tilde{T}_{t,1}((\boldsymbol{\beta}_0, \boldsymbol{\tau}_0^2), (\boldsymbol{\beta}, \boldsymbol{\tau}^2)) \\
&\quad = \prod_{l=1}^{k} \prod_{j \in G_l^t} f_t(\beta_j | \tau_j^2, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) f_t(\tau_j^2 | \beta_{0j}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) \\
&\quad \geq c(\tilde{V}_t, d) \prod_{l=1}^{k} \prod_{j \in G_l^t} \prod_{j=1}^{p} f_t(\beta_j | \tau_j^2, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) g_t(\tau_j^2 | \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}),
\end{aligned}
$$

where

$$
\begin{aligned}
g_t(\tau_j^2 | \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) = &\sqrt{\frac{\widehat{\lambda}^{2(t-1)}}{2\pi}} (\tau_j^2)^{-1/2} \\
&\exp\left\{-\frac{1}{2}\left(\sqrt{\widehat{\lambda}^{2(t-1)} \tau_j^2} - \sqrt{\frac{d\left[\frac{1}{m_0}\sum_{l=1}^{k} \boldsymbol{H}_l' \boldsymbol{H}_l + 1\right]}{\tau_j^2 \widehat{\sigma}^{2(t-1)}}}\right)^2\right\}
\end{aligned}
$$

is a density function. Hence the minorization condition is established.

Geometric drift condition.

$$
E[\sum_{l=1}^{k} \tilde{V}_t(\boldsymbol{\beta}_l, \boldsymbol{\tau}_l^2) | \boldsymbol{\beta}_0, \boldsymbol{\tau}_0^2] = E_2[E_1[\sum_{l=1}^{k} \tilde{V}_t(\boldsymbol{\beta}_l, \boldsymbol{\tau}_l^2) | \boldsymbol{\beta}_0, \boldsymbol{\tau}_0^2]],
$$

where the inner expectation is w.r.t conditional distribution of $\boldsymbol{\beta}|\boldsymbol{\tau}_0^2$ and the outer

expectation is w.r.t. $\boldsymbol{\tau}^2|\boldsymbol{\beta}_0$.

$$
\begin{aligned}
&E_1[\sum_{l=1}^{k}\tilde{V}_t(\boldsymbol{\beta}_l,\boldsymbol{\tau}_l^2)|\boldsymbol{\beta}_0,\boldsymbol{\tau}_0^2] \\
&= \frac{\sum_{l=1}^{k}\boldsymbol{H}_l'(\boldsymbol{S}_{1,l}^{(t)}+\boldsymbol{M}_{\boldsymbol{\tau}_{0,l}}^{-1})^{-1}(\boldsymbol{S}_{1,l}^{(t)}+\boldsymbol{M}_{\boldsymbol{\tau}_{0,l}}^{-1})^{-1}\boldsymbol{H}_l + tr(\widehat{\sigma}^{2(t-1)}(\boldsymbol{S}_{1,l}^{(t)}+\boldsymbol{M}_{\boldsymbol{\tau}_{0,l}}^{-1})^{-1})}{\left[\frac{1}{m_0}\sum_{l=1}^{k}\boldsymbol{H}_l'\boldsymbol{H}_l+1\right]} \\
&\quad + \sum_{j=1}^{p}\tau_j^2 \\
&\leq \frac{(\sum_{j=1}^{p}\tau_{0j}^2)\frac{1}{m_0}\sum_{l=1}^{k}\boldsymbol{H}_l'\boldsymbol{H}_l + \widehat{\sigma}^{2(t-1)}\frac{p}{m_0}}{\left[\frac{1}{m_0}\sum_{l=1}^{k}\boldsymbol{H}_l'\boldsymbol{H}_l+1\right]} + \sum_{j=1}^{p}\tau_j^2 \\
&\leq (\sum_{j=1}^{p}\tau_{0j}^2)\frac{\frac{1}{m_0}\sum_{l=1}^{k}\boldsymbol{H}_l'\boldsymbol{H}_l}{\left[\frac{1}{m_0}\sum_{l=1}^{k}\boldsymbol{H}_l'\boldsymbol{H}_l+1\right]} + \frac{\widehat{\sigma}^{2(t-1)}\frac{p}{m_0}}{\left[\frac{1}{m_0}\sum_{l=1}^{k}\boldsymbol{H}_l'\boldsymbol{H}_l+1\right]} + \sum_{j=1}^{p}\tau_j^2, \qquad \text{(A.6)}
\end{aligned}
$$

where the second step follows

$$
(\boldsymbol{S}_{1,l}^{(t)}+\boldsymbol{M}_{\boldsymbol{\tau}_{0,l}}^{-1})^{-1} \leq \frac{1}{m_0}\boldsymbol{I},\ \ (\boldsymbol{S}_{1,l}^{(t)}+\boldsymbol{M}_{\boldsymbol{\tau}_{0,l}}^{-1})^{-1} \leq \sum_{j=1}^{p}\tau_j^2. \qquad \text{(A.7)}
$$

$$
\begin{aligned}
E_2[\sum_{j=1}^{p}\tau_j^2] &= \sum_{j=1}^{p}\left[\sqrt{\frac{\beta_{0j}^2}{\widehat{\sigma}^{2(t-1)}\widehat{\lambda}^{2(t-1)}}}+\frac{1}{\widehat{\lambda}^{2(t-1)}}\right] \\
&= \sum_{j=1}^{p}\sqrt{\frac{\beta_{0j}^2}{\left[\frac{1}{m_0}\sum_{l=1}^{k}\boldsymbol{H}_l'\boldsymbol{H}_l+1\right]}\frac{\left[\frac{1}{m_0}\sum_{l=1}^{k}\boldsymbol{H}_l'\boldsymbol{H}_l+1\right]}{\widehat{\sigma}^{2(t-1)}\widehat{\lambda}^{2(t-1)}}}+\frac{p}{\widehat{\lambda}^{2(t-1)}} \\
&\leq \frac{\boldsymbol{\beta}_0'\boldsymbol{\beta}_0}{2\left[\frac{1}{m_0}\sum_{l=1}^{k}\boldsymbol{H}_l'\boldsymbol{H}_l+1\right]} + \frac{p\left[\frac{1}{m_0}\sum_{l=1}^{k}\boldsymbol{H}_l'\boldsymbol{H}_l+1\right]}{2\widehat{\sigma}^{2(t-1)}\widehat{\lambda}^{2(t-1)}} + \frac{p}{\widehat{\lambda}^{2(t-1)}}, \qquad \text{(A.8)}
\end{aligned}
$$

where the last inequality follows by the Cauchy-Schwartz inequality. Using (A.6)

and (A.8),

$$
\begin{aligned}
E[\sum_{l=1}^{k} V(\boldsymbol{\beta}_l, \boldsymbol{\tau}_l^2)|\boldsymbol{\beta}_0, \boldsymbol{\tau}_0^2] &\leq (\sum_{j=1}^{p} \tau_{0j}^2) \frac{\frac{1}{m_0}\sum_{l=1}^{k} \boldsymbol{H}_l'\boldsymbol{H}_l}{\left[\frac{1}{m_0}\sum_{l=1}^{k} \boldsymbol{H}_l'\boldsymbol{H}_l + 1\right]} + \frac{\widehat{\sigma}^{2(t-1)}\frac{p}{m_0}}{\left[\frac{1}{m_0}\sum_{l=1}^{k} \boldsymbol{H}_l'\boldsymbol{H}_l + 1\right]} \\
&\quad + \frac{\boldsymbol{\beta}_0'\boldsymbol{\beta}_0}{2\left[\frac{1}{m_0}\sum_{l=1}^{k} \boldsymbol{H}_l'\boldsymbol{H}_l + 1\right]} + \frac{p\left[\frac{1}{m_0}\sum_{l=1}^{k} \boldsymbol{H}_l'\boldsymbol{H}_l + 1\right]}{2\widehat{\sigma}^{2(t-1)}\widehat{\lambda}^{2(t-1)}} \\
&\quad + \frac{p}{\widehat{\lambda}^{2(t-1)}} \\
&\leq \gamma V(\boldsymbol{\beta}_0, \boldsymbol{\tau}_0^2) + b,
\end{aligned}
$$

where $0 < \gamma = \max\left\{\frac{1}{2}, \frac{\frac{1}{m_0}\sum_{l=1}^{k} \boldsymbol{H}_l'\boldsymbol{H}_l}{\left[\frac{1}{m_0}\sum_{l=1}^{k} \boldsymbol{H}_l'\boldsymbol{H}_l + 1\right]}\right\} < 1$ and $b = \frac{\hat{\sigma}^{2(t-1)}\frac{p}{m_0}}{\left[\frac{1}{m_0}\sum_{l=1}^{k} \boldsymbol{H}_l'\boldsymbol{H}_l + 1\right]} + \frac{p}{\hat{\lambda}^{2(t-1)}} + \frac{p\left[\frac{1}{m_0}\sum_{l=1}^{k} \boldsymbol{H}_l'\boldsymbol{H}_l + 1\right]}{2\hat{\sigma}^{2(t-1)}\hat{\lambda}^{2(t-1)}} > 0$. Hence the geometric drift condition is satisfied. Geometric drift and minorization condition together implies geometric ergodicity of the chain.

□

We will now prove a similar result for the spike and lasso model. Indeed,

Theorem A.1.4. *Assume there exists $m_0 > 0$ s.t. $e_{min}(\boldsymbol{S}_{1,\nabla}^{(t)}) \geq m_0$, for any set $\nabla \subseteq \{1, ..., p\}$ and any $t = 1, ..., T$, where $\boldsymbol{S}_{1,\nabla}^{(t)}$ is a submatrix of $\boldsymbol{S}_1^{(t)}$ with columns corresponding to the indices ∇. Then the DFP Bayesian spike and lasso transition kernel is geometrically ergodic.*

Proof. If $T_t((\boldsymbol{\beta}, \boldsymbol{\tau}^2, \sigma^2, \lambda^2, \theta, \boldsymbol{\gamma}), (\boldsymbol{\beta}', (\boldsymbol{\tau}^2)', (\sigma^2)', (\lambda^2)', (\theta)', (\boldsymbol{\gamma})'))$ is the transition kernel of the DFP and $\pi_t(\boldsymbol{\beta}, \boldsymbol{\tau}^2, \sigma^2, \lambda^2, \theta, \boldsymbol{\gamma})$ is the stationary distribution of the transition kernel, then $T_t(\cdot, \cdot)$ and $\pi_t(\cdot)$ for the Bayesian spike and lasso model are

given by

$$
\begin{aligned}
&T_t((\boldsymbol{\beta}, \boldsymbol{\tau}^2, \sigma^2, \lambda^2, \theta, \boldsymbol{\gamma}), ((\boldsymbol{\beta})', (\boldsymbol{\tau}^2)', (\sigma^2)', (\lambda^2)', (\theta)', (\boldsymbol{\gamma})')) \\
&\quad = \Big\{ f_t((\boldsymbol{\beta}_1)' | (\boldsymbol{\tau}_1^2), \widehat{\boldsymbol{\beta}}_{-1}^{(t-1)}, \widehat{\boldsymbol{\tau}}_{-1}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) \\
&\qquad f_t((\boldsymbol{\tau}_1^2)' | (\boldsymbol{\beta}_1)', \widehat{\boldsymbol{\beta}}_{-1}^{(t-1)}, \widehat{\boldsymbol{\tau}}_{-1}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) \Big\} \\
&\qquad \prod_l \prod_{j \in G_l^t} \Big\{ f_t((\beta_j)' | (\tau_j^2), \widehat{\boldsymbol{\beta}}_{-l}^{(t-1)}, \widehat{\boldsymbol{\tau}}_{-l}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) \\
&\qquad f_t((\tau_j^2)' | (\beta_j)', \hat{\beta}_{-l}^{(t-1)}, \widehat{\tau}_{-l}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) \Big\} \\
&\qquad f_t((\sigma^2)' | \widehat{\boldsymbol{\beta}}^{(t-1)}, \widehat{\boldsymbol{\tau}}^{2(t-1)}) f_t((\lambda^2)' | \widehat{\boldsymbol{\beta}}^{(t-1)}, \widehat{\boldsymbol{\tau}}^{2(t-1)}) \\
&\qquad f_t((\theta)' | \widehat{\boldsymbol{\beta}}^{(t-1)}, \widehat{\boldsymbol{\tau}}^{2(t-1)}) \prod_{j=1}^{p} f_t((\boldsymbol{\gamma})' | \widehat{\boldsymbol{\beta}}^{(t-1)}, \widehat{\boldsymbol{\tau}}^{2(t-1)}) \qquad \text{(A.9)}
\end{aligned}
$$

$$
\begin{aligned}
&\pi_t(\boldsymbol{\beta}, \boldsymbol{\tau}^2, \sigma^2, \lambda^2, \theta, \boldsymbol{\gamma}) \\
&\quad = \prod_{l=1}^{k} \Big\{ f_t(\boldsymbol{\beta}_l, \boldsymbol{\tau}_l^2 | \widehat{\boldsymbol{\beta}}_{-l}^{(t-1)}, \widehat{\boldsymbol{\tau}}_{-l}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}) \Big\} \\
&\qquad f_t(\sigma^2 | \widehat{\boldsymbol{\beta}}^{(t-1)}, \widehat{\boldsymbol{\tau}}^{2(t-1)}) f_t(\lambda^2 | \widehat{\boldsymbol{\beta}}^{(t-1)}, \widehat{\boldsymbol{\tau}}^{2(t-1)}) \\
&\qquad f_t(\theta | \widehat{\boldsymbol{\beta}}^{(t-1)}, \widehat{\boldsymbol{\tau}}^{2(t-1)}) f_t(\boldsymbol{\gamma} | \widehat{\boldsymbol{\beta}}^{(t-1)}, \widehat{\boldsymbol{\tau}}^{2(t-1)}). \qquad \text{(A.10)}
\end{aligned}
$$

where $\boldsymbol{\beta}_1 = \{\beta_j : \beta_j \in \boldsymbol{\Theta}_{1,t}\}, \boldsymbol{\tau}_1^2 = \{\tau_j^2 : \tau_j^2 \in \boldsymbol{\Theta}_{1,t}\}$. Hence, $||T_t - \pi_t||_{TV} = ||\tilde{T}_{t,1} - \tilde{\pi}_{t,1}||_{TV}$, where

$$
\begin{aligned}
\tilde{T}_{t,1} &= \Big\{ f_t((\boldsymbol{\beta}_1)' | (\boldsymbol{\tau}^2), \widehat{\boldsymbol{\beta}}_{-1}^{(t-1)}, \widehat{\boldsymbol{\tau}}_{-1}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}, \widehat{\theta}^{(t-1)}, \widehat{\boldsymbol{\gamma}}^{(t-1)}) \\
&\qquad f_t((\boldsymbol{\tau}_1^2)' | (\boldsymbol{\beta}_1)', \widehat{\boldsymbol{\beta}}_{-1}^{(t-1)}, \widehat{\boldsymbol{\tau}}_{-1}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}, \widehat{\theta}^{(t-1)}, \widehat{\boldsymbol{\gamma}}^{(t-1)}) \Big\} \\
\tilde{\pi}_{t,1} &= \Big\{ f_t(\boldsymbol{\beta}_1, \boldsymbol{\tau}_1^2 | \widehat{\boldsymbol{\beta}}_{-1}^{(t-1)}, \widehat{\boldsymbol{\tau}}_{-1}^{2(t-1)}, \widehat{\sigma}^{2(t-1)}, \widehat{\lambda}^{2(t-1)}, \widehat{\theta}^{(t-1)}, \widehat{\boldsymbol{\gamma}}^{(t-1)}) \Big\}.
\end{aligned}
$$

Thus it is enough to show the geometric ergodicity of the chain by establishing a geometric drift condition and a geometric minorization condition for the $(\boldsymbol{\beta}_1, \boldsymbol{\tau}_1^2)$

chain. Define, $\tilde{V}_t(\boldsymbol{\beta}_1, \boldsymbol{\tau}_1^2) = \frac{\boldsymbol{\beta}_1'\boldsymbol{\beta}_1}{\frac{1}{m_0}\boldsymbol{H}_1'\boldsymbol{H}_1+1} + \mathbf{1}'\boldsymbol{\tau}_1^2\mathbf{1}$, where $\boldsymbol{H}_l = \boldsymbol{S}_{2,l}^{(t)} - \boldsymbol{S}_{1,l,-l}^{(t)}\widehat{\boldsymbol{\beta}}_{-l}^{(t-1)}$. Using similar calculations as in Theorem A.1.3, the proof of minorization and geometric drift conditions follow. □

It remains to show (iv) in Corollary 0.3. The lemma presented below develops sufficient conditions to derive (iv). The lemma is presented for a general likelihood function $p_{\boldsymbol{\Theta}}(\cdot)$.

Lemma A.1.5. *Assume the following,*
(B1) the likelihood function $p_{\boldsymbol{\Theta}}(\cdot)$ is continuous as a function of $\boldsymbol{\Theta}$ at $\boldsymbol{\Theta}^0 = (\boldsymbol{\Theta}_1^0, .., \boldsymbol{\Theta}_k^0)$ and $\sqrt{t}p_{\boldsymbol{\Theta}^0}(\boldsymbol{D}^{(t)})$ in limit is bounded away from 0 and ∞;
(B2) $\boldsymbol{\Theta}^0$ is an interior point in the domain and prior distribution $\pi_0(\boldsymbol{\Theta})$ is positive and continuous at $\boldsymbol{\Theta}^0$;
(B3) $\widehat{\boldsymbol{\Theta}}^{(t)} \to \boldsymbol{\Theta}^0$ a.s. under the data generating law at $\boldsymbol{\Theta}^0$;
(B4) for a neighborhood $\mathcal{N}_\vartheta = \{\boldsymbol{\Theta} : ||\boldsymbol{\Theta} - \boldsymbol{\Theta}_0|| < \vartheta\Delta_t\}$, $f_t(\mathcal{N}_\vartheta) \to 1$ and $\pi_t(\mathcal{N}_\vartheta) \to 1$, as $t \to \infty$, under the data generating law at $\boldsymbol{\Theta}_0$, for any $\vartheta > 0$. Thus, the distributions f_t and π_t both concentrate around $\boldsymbol{\Theta}^0$ at a rate Δ_t (with $\Delta_t \downarrow 0$ as $t \to \infty$).
Under (B1)-(B4), $\exists\ \kappa_t$ depending on Δ_t, s.t. $\kappa_t \to 0$ and $||f_t - \pi_t||_{TV} = 2\int \left|\pi_t(\boldsymbol{\Theta}) - f_t(\boldsymbol{\Theta})\right| d\boldsymbol{\Theta} \leq 2\kappa_t$ for large t, a.s. under the data generating model at $\boldsymbol{\Theta}^0$.

Proof. Stationary distribution π_t of the DFP transition kernel T_t is the approximate posterior distribution to π_t obtained at time t, and by Lemma A.1.1 is given by

$$\pi_t(\boldsymbol{\Theta}_1, ..., \boldsymbol{\Theta}_k) = \prod_{l=1}^{k} f_t(\boldsymbol{\Theta}_l|\widehat{\boldsymbol{\Theta}}_{-l}^{(t-1)}) = \frac{\prod_{l=1}^{k}\prod_{s=1}^{t}\left\{p_{\boldsymbol{\Theta}_l,\widehat{\boldsymbol{\Theta}}_{-l}^{(t-1)}}(\boldsymbol{D}_s)\pi_0(\widehat{\boldsymbol{\Theta}}_{-l}^{(t-1)}, \boldsymbol{\Theta}_l)\right\}}{\int \prod_{l=1}^{k}\prod_{s=1}^{t}\left\{p_{\boldsymbol{\Theta}_l,\widehat{\boldsymbol{\Theta}}_{-l}^{(t-1)}}(\boldsymbol{D}_s)\pi_0(\widehat{\boldsymbol{\Theta}}_{-l}^{(t-1)}, \boldsymbol{\Theta}_l)\right\}}.$$

By assumption, $\widehat{\boldsymbol{\Theta}}_l^{(t)} \to \boldsymbol{\Theta}_l^0$ a.s. under $\boldsymbol{\Theta}^0$, there exists Ω_0 which has probability 1

under the data generating law s.t. for all $\omega \in \Omega_0$, $\widehat{\boldsymbol{\Theta}}_l^{(t)}(\omega)$ is in an arbitrarily small neighborhood of $\boldsymbol{\Theta}_l^0$, $l = 1, ..., k$. Also by assumption, prior π_0 is continuous at $\boldsymbol{\Theta}^0$. That is, given $\epsilon_t > 0$ and $\eta_{1,t}, \eta_{2,t} > 0$, there exists a neighborhood $N_{\epsilon_t,\eta_{1,t},\eta_{2,t}} = \{\boldsymbol{\Theta} : ||\boldsymbol{\Theta} - \boldsymbol{\Theta}^0|| \leq M\Delta_t\}$ s.t. for all $\boldsymbol{\Theta} \in N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}$ one has $|\pi_0(\boldsymbol{\Theta}_1, ..., \boldsymbol{\Theta}_k) - \pi_0(\boldsymbol{\Theta}_1^0, ..., \boldsymbol{\Theta}_k^0)| < \epsilon_t$. Using this and the consistency of $\widehat{\boldsymbol{\Theta}}_l^{(t)}$, $l = 1, ..., k$ as above, one obtains for all $t > t_0$ and $\omega \in \Omega_0$

$$|\pi_0(\boldsymbol{\Theta}_l, \widehat{\boldsymbol{\Theta}}_{-l}^{(t-1)}) - \pi_0(\boldsymbol{\Theta}^0)| < \epsilon_t, \tag{A.11}$$

Similarly, continuity of $p_{\boldsymbol{\Theta}}(\cdot)$ at $\boldsymbol{\Theta}^0$ leads to the condition that for all $t > t_0$,

$$|p_{\boldsymbol{\Theta}_1,...,\boldsymbol{\Theta}_k}(\boldsymbol{D}_t) - p_{\boldsymbol{\Theta}_1^0,...,\boldsymbol{\Theta}_k^0}(\boldsymbol{D}_t)| < \epsilon_t. \tag{A.12}$$

Further, convergence assumptions on f_t and π_t yield that for all $t > t_1$ and $\omega \in \Omega_1$ $f_t(N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}|\boldsymbol{D}^{(t)}(\omega)) > 1 - \eta_{1,t}, \pi_t(N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}|\boldsymbol{D}^{(t)}(\omega)) > 1 - \eta_{2,t}$, where Ω_1 has probability 1 under the data generating law. Considering $\Omega = \Omega_0 \cap \Omega_1$ and $t_2 = \max\{t_1, t_0\}$ it is evident that Ω has probability 1 under the true data generating law and all of the above conditions hold for $t > t_2$ and $\omega \in \Omega$. Simple algebraic manipulations yield

$$\frac{\pi_t(\boldsymbol{\Theta}|\boldsymbol{D}^{(t)}(\omega))}{f_t(\boldsymbol{\Theta}|\boldsymbol{D}^{(t)}(\omega))} = \frac{\pi_t(N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}|\boldsymbol{D}^{(t)}(\omega))}{f_t(N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}|\boldsymbol{D}^{(t)}(\omega))} \frac{\int_{N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}} \prod_{s=1}^t p_{\boldsymbol{\Theta}}(\boldsymbol{D}_s)\pi_0(\boldsymbol{\Theta})}{\prod_{s=1}^t p_{\boldsymbol{\Theta}}(\boldsymbol{D}_s)\pi_0(\boldsymbol{\Theta})}$$
$$\frac{\left[\prod_{s=1}^t \prod_{l=1}^k p_{\boldsymbol{\Theta}_l,\widehat{\boldsymbol{\Theta}}_{-l}^{(t)}}(\boldsymbol{D}_s)\pi_0(\widehat{\boldsymbol{\Theta}}_{-l}^{(t)}, \boldsymbol{\Theta}_l)\right]}{\int_{N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}} \left[\prod_{s=1}^t \prod_{l=1}^k p_{\boldsymbol{\Theta}_l,\widehat{\boldsymbol{\Theta}}_{-l}^{(t)}}(\boldsymbol{D}_s)\pi_0(\widehat{\boldsymbol{\Theta}}_{-l}^{(t)}, \boldsymbol{\Theta}_l)\right]}.$$

Using (A.11) we have

$$(\pi_0(\boldsymbol{\Theta}^0)-\epsilon)\int\limits_{N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}}\prod_{s=1}^{t}p_{\boldsymbol{\Theta}}(\boldsymbol{D}_s)\leq\int\limits_{N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}}\Big[\prod_{s=1}^{t}p_{\boldsymbol{\Theta}}(\boldsymbol{D}_s)\Big]\pi_0(\boldsymbol{\Theta})$$
$$\leq(\pi_0(\boldsymbol{\Theta}^0)+\epsilon)\int\limits_{N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}}\prod_{s=1}^{t}p_{\boldsymbol{\Theta}}(\boldsymbol{D}_s).$$

Thus,

$$\frac{\pi_t(\boldsymbol{\Theta}|\boldsymbol{D}^{(t)}(\omega))}{f_t(\boldsymbol{\Theta}|\boldsymbol{D}^{(t)}(\omega))}\leq\frac{(1-\eta_{1,t})^{-1}\prod_{s=1}^{t}\prod_{l=1}^{k}p_{\boldsymbol{\Theta}_l,\widehat{\boldsymbol{\Theta}}_{-l}^{(t)}}(\boldsymbol{D}_s)}{\int_{N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}}\prod_{s=1}^{t}\prod_{l=1}^{k}p_{\boldsymbol{\Theta}_l,\widehat{\boldsymbol{\Theta}}_{-l}^{(t)}}(\boldsymbol{D}_s)}\frac{\int_{N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}}\prod_{s=1}^{t}p_{\boldsymbol{\Theta}}(\boldsymbol{D}_s)}{\prod_{s=1}^{t}p_{\boldsymbol{\Theta}}(\boldsymbol{D}_s)}$$
$$\frac{(\pi_0(\boldsymbol{\Theta}^0)+\epsilon)^3}{(\pi_0(\boldsymbol{\Theta}^0)-\epsilon)^3}.$$

Using similar calculations we have

$$\frac{\pi_t(\boldsymbol{\Theta}|\boldsymbol{D}^{(t)}(\omega))}{f_t(\boldsymbol{\Theta}|\boldsymbol{D}^{(t)}(\omega))}\geq\frac{(1-\eta_{2,t})\prod_{s=1}^{t}\prod_{l=1}^{k}p_{\boldsymbol{\Theta}_l,\widehat{\boldsymbol{\Theta}}_{-l}^{(t)}}(\boldsymbol{D}_s)}{\int_{N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}}\prod_{s=1}^{t}\prod_{l=1}^{k}p_{\boldsymbol{\Theta}_l,\widehat{\boldsymbol{\Theta}}_{-l}^{(t)}}(\boldsymbol{D}_s)}\frac{\int_{N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}}\prod_{s=1}^{t}p_{\boldsymbol{\Theta}}(\boldsymbol{D}_s)}{\prod_{s=1}^{t}p_{\boldsymbol{\Theta}}(\boldsymbol{D}_s)}$$
$$\frac{(\pi_0(\boldsymbol{\Theta}^0)-\epsilon)^3}{(\pi_0(\boldsymbol{\Theta}^0)+\epsilon)^3}.$$

Condition (A.12) now gives us

$$\frac{\prod_{s=1}^{t}(p_{\boldsymbol{\Theta}^0}(\boldsymbol{D}_s)-\epsilon)^3}{\prod_{s=1}^{t}(p_{\boldsymbol{\Theta}^0}(\boldsymbol{D}_s)+\epsilon)^3}\leq\frac{\prod_{s=1}^{t}\prod_{l=1}^{k}p_{\boldsymbol{\Theta}_l,\hat{\boldsymbol{\Theta}}_{-l}^{t}}(\boldsymbol{D}_s)}{\int_{N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}}\prod_{s=1}^{t}\prod_{l=1}^{k}p_{\boldsymbol{\Theta}_l,\widehat{\boldsymbol{\Theta}}_{-l}^{(t)}}(\boldsymbol{D}_s)}$$
$$\frac{\int_{N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}}\prod_{s=1}^{t}p_{\boldsymbol{\Theta}}(\boldsymbol{D}_s)}{\prod_{s=1}^{t}p_{\boldsymbol{\Theta}}(\boldsymbol{D}_s)}$$
$$\leq\frac{\prod_{s=1}^{t}(p_{\boldsymbol{\Theta}^0}(\boldsymbol{D}_s)+\epsilon)^3}{\prod_{s=1}^{t}(p_{\boldsymbol{\Theta}^0}(\boldsymbol{D}_s)-\epsilon)^3}.$$

Using the condition that $\lim_{t\to\infty}\sqrt{t}p_{\boldsymbol{\Theta}^0}(\boldsymbol{D}^{(t)})$ is bounded away from 0 and ∞ and choosing ϵ,η sufficiently small, we have $\left|\frac{\pi_t(\boldsymbol{\Theta}|\boldsymbol{D}^{(t)}(\omega))}{f_t(\boldsymbol{\Theta}|\boldsymbol{D}^{(t)}(\omega))}-1\right|<\upsilon_t$ for all $t>$

t_2 and $\omega \in \Omega$. Finally,

$$\begin{aligned}\int |\pi_t(\boldsymbol{\Theta}) - f_t(\boldsymbol{\Theta})| &\leq \int_{N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}} |\pi_t(\boldsymbol{\Theta}) - f_t(\boldsymbol{\Theta})| + \int_{N^c_{\epsilon_t,\eta_{1,t},\eta_{2,t}}} |\pi_t(\boldsymbol{\Theta}) - f_t(\boldsymbol{\Theta})| \\ &\leq \int_{N_{\epsilon_t,\eta_{1,t},\eta_{2,t}}} |\pi_t(\boldsymbol{\Theta}) - f_t(\boldsymbol{\Theta})| + \eta_{1,t} + \eta_{2,t} \\ &\leq f_t(N_{\epsilon_t,\eta_{1,t},\eta_{2,t}})\upsilon_t + \eta_{1,t} + \eta_{2,t} < \upsilon_t + \eta_{1,t} + \eta_{2,t} = \kappa_t.\end{aligned}$$

□

Remark: Lemma A.1.5 outlines sufficient conditions (B1)-(B4) for the DFP stationary distribution to be close to the full posterior distribution as t increases. One of the important sufficient conditions pertains to the consistency of the sequence of estimators $\widehat{\boldsymbol{\Theta}}^{(t)}$ (assumption (B3)). Referring to Section 2.3.2 of chapter 2, we construct $\widehat{\boldsymbol{\Theta}}^{(t)}$ as the average of samples of $\boldsymbol{\Theta}$ drawn from the DFP algorithm at time t. While consistency of $\widehat{\boldsymbol{\Theta}}^{(t)}$ constructed in this way is difficult to prove theoretically, we have empirically demonstrated that $\widehat{\boldsymbol{\Theta}}^{(t)}$ concentrates around the true $\boldsymbol{\Theta}_0$ in the simulation examples presented in Section 2.4 of chapter 2. It is mentioned that the theory allows alternative constructions of $\widehat{\boldsymbol{\Theta}}^{(t)}$, as long as the consistency condition is met. As a future exploration, we plan to develop a hybrid DFP algorithm where $\widehat{\boldsymbol{\Theta}}^{(t)}$ is constructed rapidly by implementing a separate frequentist high dimensional regression technique (e.g., lasso) at the onset of a new data shard at every time. This will guarantee consistency of $\widehat{\boldsymbol{\Theta}}^{(t)}$ and the purpose of the DFP algorithm then becomes quantifying uncertainty on the parameters.

We also emphasize that the assumption (B4) on the concentration of posterior f_t around the truth $\boldsymbol{\Theta}_0$ is reasonable as $t \to \infty$, as there is a fairly well developed literature that shows posterior of parameters concentrating around the true parameter $\boldsymbol{\Theta}_0$ at a rate close to $t^{-1/2}$ in the high dimensional regression with variable selection or shrinkage priors, see e.g., Song and Liang (2017). Similar proof

techniques can be adapted to show the concentration of π_t around $\boldsymbol{\Theta}_0$ as $t \to \infty$. We will take Δ_t to be the smallest among the rates of convergence for π_t and f_t.

A.2 Algorithms

This section details out specifics of the DFP algorithm for high dimensional regression with the Bayesian Lasso prior (Section 2.4.1 of chapter 2), Horseshoe prior (Section 2.4.2 of chapter 2) and Spike-and-Lasso prior (Section 2.4.3 of chapter 2).

A.2.1 Bayesian Lasso Prior

1. *Initialize:* Initialize variables $\boldsymbol{\beta}$, $\boldsymbol{\tau}^2$, σ^2 and λ. Set $\widehat{\boldsymbol{\beta}}^{(0)}, \widehat{\sigma}^{2(0)}, \widehat{\boldsymbol{\tau}}^{2(0)}, \widehat{\lambda}^{2(0)}$ at their initial values.

2. *Observe data and partition parameter space at time t:* Observe data $\boldsymbol{D}_t = \{\boldsymbol{y}_t, \boldsymbol{X}_t\}$ at time t. Update the partitions of the parameters based on the iterates of the parameters at time $(t-1)$. The parameter partitioning algorithm at time t for the shrinkage priors is given in Section 2.3.

3. *Update sufficient statistics:* Update sufficient statistics $\boldsymbol{S}_1^{(t)}, \boldsymbol{S}_2^{(t)}, \boldsymbol{S}_3^{(t)}$ based on
   $\boldsymbol{S}_1^{(t-1)}, \boldsymbol{S}_2^{(t-1)}, \boldsymbol{S}_3^{(t-1)}$ and $\boldsymbol{D}_t$ with the equations given in Section 2.2.1.

4. *Drawing approximate posterior samples:* Draw S samples from the DFP full

conditional posterior distributions of $\boldsymbol{\beta}_l$ and $\boldsymbol{\tau}_l^2$ given by

$$\frac{1}{\tau_j^2}|\cdot \sim Inv-Gaussian\left(\sqrt{\frac{\widehat{\lambda}^{2(t-1)}\widehat{\sigma}^{2(t-1)}}{\beta_j^2}}, \widehat{\lambda}^{2(t-1)}\right) \forall \tau_j^2 \in \boldsymbol{\tau}_l^2,$$
$$\boldsymbol{\beta}_l \sim N\left(\boldsymbol{\mu}_{\boldsymbol{\beta}_l^{(t)}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_l^{(t)}}\right) \boldsymbol{\mu}_{\boldsymbol{\beta}_l^{(t)}} = \left(\boldsymbol{S}_{1,l}^{(t)} + \boldsymbol{M}_{\tau,l}^{-1}\right)^{-1}\left(\boldsymbol{S}_{2,l}^{(t)} - \boldsymbol{S}_{1,l,-l}^{(t)}\widehat{\boldsymbol{\beta}_{-l}}^{(t-1)}\right),$$
$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}_l^{(t)}} = \widehat{\sigma}^{2(t-1)}\left(\boldsymbol{S}_{1,l}^{(t)} + \boldsymbol{M}_{\tau,l}^{-1}\right)^{-1}.$$

The conditional distributions of the parameters in the lth server depends on the lower dimensional functions of sufficient statistics, point estimates from time $(t-1)$ and the other parameters from the same partition. This is conceptualized in the notation $\boldsymbol{J}_{l,j}^{(t)}$ in Section 2.3. Sampling from the DFP full conditionals of $\{\boldsymbol{\beta}_l, \boldsymbol{\tau}_l^2\}$ $(l = 1, ..., b_t)$ is performed on b_t servers in parallel. In the (b_t+1)-th server, draw S samples from the DFP conditional distributions of λ^2 and σ^2 given by $\lambda^2 \sim Gamma\left(p+r, \frac{\sum_{j=1}^{p}\widehat{\tau}_j^{2(t-1)}}{2} + d\right)$, $\sigma^2 \sim IG\left(\frac{nt+p}{2}, \frac{\left(\boldsymbol{S}_3^{(t)} + \widehat{\boldsymbol{\beta}}^{(t-1)'}\boldsymbol{S}_1^{(t)}\widehat{\boldsymbol{\beta}}^{(t-1)} - 2\widehat{\boldsymbol{\beta}}^{(t-1)'}\boldsymbol{S}_2^{(t)}\right) + \widehat{\boldsymbol{\beta}}^{(t-1)'}(\widehat{\boldsymbol{M}}_{\tau}^{(t-1)})^{-1}\widehat{\boldsymbol{\beta}}^{(t-1)}}{2}\right)$.

5. <u>*Compute the sequence of estimators at time t:*</u> Set $\widehat{\boldsymbol{\beta}}^{(t)}$, $\widehat{\boldsymbol{\tau}}^{2(t)}$, $\widehat{\sigma}^{2(t)}$, $\widehat{\lambda}^{2(t)}$ from their respective sample averages from S MCMC samples.

A.2.2 Horseshoe Prior

1. Set $\widehat{\boldsymbol{\beta}}^{(0)}, \widehat{\sigma}^{2(0)}, \widehat{\boldsymbol{\lambda}}^{2(0)}, \widehat{\boldsymbol{\nu}}^{2(0)}, \widehat{\tau}^{2(0)}$ and $\widehat{\xi}^{(0)}$ at their initial values.

2. Observe data $\boldsymbol{D}_t = \{\boldsymbol{y}_t, \boldsymbol{X}_t\}$ at time t. Update the partitions of the parameters based on the iterates of the parameters at time $(t-1)$. The dynamic partitioning scheme for parameters for shrinkage priors described in Section 2.3 is employed. Throughout, the partitions $G_{b_t+1}^{(t)}$ and $G_{b_t+2}^{(t)}$ are kept fixed.

3. Update sufficient statistics $\boldsymbol{S}_1^{(t)}, \boldsymbol{S}_2^{(t)}, \boldsymbol{S}_3^{(t)}$ based on $\boldsymbol{S}_1^{(t-1)}, \boldsymbol{S}_2^{(t-1)}, \boldsymbol{S}_3^{(t-1)}$ and

$\boldsymbol{D}_t$ with the equations given in Section 2.2.2.

4. Draw S samples from the DFP conditional distributions of $\boldsymbol{\beta}_l$ and $\boldsymbol{\lambda}_l$ given by

$$\lambda_j^2 \sim IG\left(1, \left[\frac{1}{\widehat{\nu}_j^{(t-1)}} + \frac{\beta_j^2}{2\widehat{\tau}^{2(t-1)}\widehat{\sigma}^{2(t-1)}}\right]\right),\ \lambda_j^2 \in \boldsymbol{\lambda}_l^2,\quad \boldsymbol{\beta}_l \sim N\left(\boldsymbol{\mu}_{\boldsymbol{\beta}_l^{(t)}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}_l^{(t)}}\right)$$
$$\boldsymbol{\mu}_{\boldsymbol{\beta}_l^{(t)}} = \left(\boldsymbol{S}_{1,l}^{(t)} + \frac{\boldsymbol{M}_{\lambda,l}^{-1}}{\tau^2}\right)^{-1}\left(\boldsymbol{S}_{2,l}^{(t)} - \boldsymbol{S}_{1,l,-l}^{(t)}\widehat{\boldsymbol{\beta}_{-l}}^{(t-1)}\right),$$
$$\boldsymbol{\Sigma}_{\boldsymbol{\beta}_l^{(t)}} = \widehat{\sigma}^{2(t-1)}\left(\boldsymbol{S}_{1,l}^{(t)} + \frac{\boldsymbol{M}_{\lambda,l}^{-1}}{\tau^2}\right)^{-1},$$

Sampling from the DFP full conditionals of $\{\boldsymbol{\beta}_l, \boldsymbol{\lambda}_l\}$ $(l = 1, .., b_t)$ are performed on b_t servers in parallel with the number of flops at most M^3 at every server. Draw S samples from the DFP full conditionals of $\boldsymbol{\nu}$ given by $\nu_j \sim IG\left(1, \left(1 + \frac{1}{\widehat{\lambda}_j^{2(t-1)}}\right)\right)$, $j = 1, ..., p$, in the $(b_t + 1)$-th server. Finally, in the $(b_t + 2)$-th server, draw S samples from the DFP full conditional posterior distributions of τ^2, ξ, σ^2 given by $\xi \sim IG\left(1, 1 + \frac{1}{\tau^2}\right)$,
$\tau^2 \sim IG\left(\frac{p+1}{2}, \frac{1}{\xi} + \frac{\hat{\boldsymbol{\beta}}^{(t-1)'}(\widehat{\boldsymbol{M}}_{\lambda}^{(t-1)})^{-1}\hat{\boldsymbol{\beta}}^{(t-1)}}{2\sigma^2}\right)$,
$\sigma^2 \sim IG\left(\frac{nt+p}{2}, \frac{\boldsymbol{S}_3^{(t)} + \hat{\boldsymbol{\beta}}^{(t-1)'}\boldsymbol{S}_1^{(t)}\hat{\boldsymbol{\beta}}^{(t-1)} - 2\hat{\boldsymbol{\beta}}^{(t-1)'}\boldsymbol{S}_2^{(t)}}{2} + \frac{\hat{\boldsymbol{\beta}}^{(t-1)'}(\widehat{\boldsymbol{M}}_{\lambda}^{(t-1)})^{-1}\hat{\boldsymbol{\beta}}^{(t-1)}}{2\tau^2}\right)$.

5. Set $\widehat{\boldsymbol{\beta}}^{(t)}$, $\widehat{\boldsymbol{\lambda}}^{2(t)}$, $\widehat{\boldsymbol{\nu}}^{(t)}$, $\widehat{\tau}^{2(t)}$, $\widehat{\sigma}^{2(t)}$ and $\widehat{\xi}^{(t)}$ as their respective sample averages from S MCMC samples.

A.2.3 Spike & Lasso Prior

1. *Initialize:* Set $\widehat{\boldsymbol{\beta}}^{(0)}, \widehat{\sigma}^{2(0)}, \widehat{\lambda}^{2(0)}, \widehat{\boldsymbol{\gamma}}^{2(0)}, \widehat{\boldsymbol{\tau}}^{2(0)}$ and $\widehat{\theta}^{(0)}$ at their initial values.

2. *Parameter space partitioning at time t:* Observe data $\boldsymbol{D}_t = \{\boldsymbol{y}_t, \boldsymbol{X}_t\}$ at time t. Update the partitions of the parameters based on the iterates of the parameters at time $(t - 1)$. As discussed in the partitioning scheme for the

Spike and Lasso prior in Section 2.3, the number of partitions is $k_t = 1+|\{j : (\beta_j, \tau_j^2) \in \boldsymbol{\Theta}_{2t}\}|$, where $|\cdot|$ denotes the cardinality of the set.

3. *Update sufficient statistics:* Update sufficient statistics $\boldsymbol{S}_1^{(t)}, \boldsymbol{S}_2^{(t)}$ and $\boldsymbol{S}_3^{(t)}$ based on $\boldsymbol{S}_1^{(t)} = \boldsymbol{S}_1^{(t-1)} + \boldsymbol{X}_t'\boldsymbol{X}_t$, $\boldsymbol{S}_2^{(t)} = \boldsymbol{S}_2^{(t-1)} + \boldsymbol{X}_t'\boldsymbol{y}_t$ and $\boldsymbol{S}_3^{(t)} = \boldsymbol{S}_3^{(t-1)} + \boldsymbol{y}_t'\boldsymbol{y}_t$.

4. *Draw approximate posterior samples at time t:* Define $\mathcal{I}_{1t} = \{j : (\boldsymbol{\beta}_j, \tau_j^2) \in \boldsymbol{\Theta}_{1t}\}$, where $\boldsymbol{\Theta}_{1t} = \{(\beta_j, \tau_j^2) : \hat{\gamma}_j^{(t-1)} = 1\}$. In a server, draw S samples from the DFP full conditional posterior distributions of $\boldsymbol{\beta}_{\mathcal{I}_{1t}} = (\beta_j : j \in \mathcal{I}_{1t})'$ and $\boldsymbol{\tau}^2_{\mathcal{I}_{1t}} = (\tau_j^2 : j \in \mathcal{I}_{1t})'$ given by

$$\begin{aligned}
&\frac{1}{\tau_j^2}|\cdot \sim Inv-Gaussian\left(\sqrt{\frac{\widehat{\lambda}^{2(t-1)}\widehat{\sigma}^{2(t-1)}}{\beta_j^2}}, \widehat{\lambda}^{2(t-1)}\right) \ \forall\, j \in \mathcal{I}_{1t},\\
&\boldsymbol{\beta}_{\mathcal{I}_{1t}} \sim N\left(\boldsymbol{\mu}_{\boldsymbol{\beta}^{(t)}_{\mathcal{I}_{1t}}}, \boldsymbol{\Sigma}_{\boldsymbol{\beta}^{(t)}_{\mathcal{I}_{1t}}}\right),\ \boldsymbol{\Sigma}_{\boldsymbol{\beta}^{(t)}_{\mathcal{I}_{1t}}} = \widehat{\sigma}^{2(t-1)}\left(\boldsymbol{S}^{(t)}_{1,\mathcal{I}_{1t}} + \boldsymbol{M}^{-1}_{\mathcal{I}_{1t}}\right)^{-1}\\
&\boldsymbol{\mu}_{\boldsymbol{\beta}^{(t)}_{\mathcal{I}_{1t}}} = \left(\boldsymbol{S}^{(t)}_{1,\mathcal{I}_{1t}} + \boldsymbol{M}^{-1}_{\mathcal{I}_{1t}}\right)^{-1}\left(\boldsymbol{S}^{(t)}_{2,\mathcal{I}_{1t}} - \boldsymbol{S}^{(t)}_{1,\mathcal{I}_{1t},-\mathcal{I}_{1t}}\widehat{\boldsymbol{\beta}_{-\mathcal{I}_{1t}}}^{(t-1)}\right),
\end{aligned}$$

where $\boldsymbol{M}_{\mathcal{I}_{1t}}$ is a sub-matrix of $\boldsymbol{M}$ corresponding to the indices of $\mathcal{I}_{1t}$, $\boldsymbol{S}^{(t)}_{1,\mathcal{I}_{1t}}, \boldsymbol{S}^{(t)}_{1,\mathcal{I}_{1t},-\mathcal{I}_{1t}}$ and $\boldsymbol{S}^{(t)}_{2,\mathcal{I}_{1t}}$ are defined analogous to the last section. Similarly draw (β_j, τ_j^2) for $j \in \mathcal{I}_{2t} = \{j : (\beta_j, \tau_j^2) \in \boldsymbol{\Theta}_{2t}\}$, $\boldsymbol{\Theta}_{2t} = \{(\beta_j, \tau_j^2) : \hat{\gamma}_j^{(t-1)} = 0\}$ in different processors from their DFP full conditional distributions. Draw S samples from the DFP full conditional posterior distributions of $\boldsymbol{\gamma}$ given in (4) with $\sigma^2, \boldsymbol{\beta}, \boldsymbol{\tau}$ replaced by their point estimates from time $(t-1)$. Finally, draw S samples from the DFP full conditional posterior distributions of λ^2, σ^2 and θ in a server.

5. *Compute the sequence of estimators at time t:* Set $\widehat{\boldsymbol{\beta}}^{(t)}$, $\widehat{\boldsymbol{\tau}}^{2(t)}$, $\widehat{\lambda}^{2(t)}$, $\widehat{\sigma}^{2(t)}$ and $\widehat{\theta}^{(t)}$ as their respective sample averages from S MCMC samples. Set $\widehat{\gamma}_j^{(t)} = 1$ if out of S approximate posterior samples of γ_j at time $(t-1)$, at least $S/2$ have resulted in $\gamma_j = 1$.

Appendix B

Bayesian Multi-Object Regression

This chapter discusses posterior computation in BOOM. Efficient sampling of the posterior is done through Gibbs sampling. Here, we exploit the combination of the two frameworks for variable selection to perform efficient computation of the posterior. In a first instance the focus on the spike and slab prior is used, and in a second step we use the global-local scale of mixtures for improved mixing. As we will see, the combination of this two steps has as a result an efficient sampling procedure.

The full posterior distribution is given by:

$$
\begin{aligned}
p(\cdot|\cdot) \propto & \prod_{i=1}^{n} N\left(y_i \middle| \beta_0 + \boldsymbol{x}_i^T \boldsymbol{\beta}_{\boldsymbol{x}} + \sum_{p=1}^{P} \boldsymbol{g}_{i,p}^T \boldsymbol{\beta}_p + \langle \boldsymbol{A}_i, \boldsymbol{\Theta} \rangle / 2, \tau^2\right) \\
& \times \prod_{p<p'} \left[N\left(\theta_{p,p'} \middle| 0, \tau^2 \sigma_\theta^2 \lambda_{p,p'}^2\right)\right]^{\xi_p \xi_{p'}} [\delta_0]^{1-\xi_p \xi_{p'}} \\
& \times \prod_{p=1}^{P} \prod_{j=1}^{V} \left[N\left(\beta_p \middle| 0, \tau^2 \Delta_p^2 \eta_{p,j}\right)\right]^{\xi_p} [\delta_0]^{1-\xi_p} \prod_{p=1}^{P} Ber(\xi_p|\nu) IG(\tau^2|a_\tau, b_\tau) \\
& \times N\left(\beta_x|a_x, b_x\right) C^+(\sigma^2|0,1) \times C^+(\Delta^2|0,1) \prod_{p'<p} C^+(\lambda_{p,p'}|0,1) \\
& \times \prod_{p=1}^{P} \prod_{j=1}^{V} C^+(\lambda_{p,p'}|0,1) \qquad \text{(B.1)}
\end{aligned}
$$

We explore this full posterior by Gibbs sampling in three main steps, of each Gibbs iteration, as follows:

- *First step*: Joint sampling, of each triplet: $\left\{\boldsymbol{\beta}_p, \boldsymbol{\theta}_p, \xi_p\right\}$ for $p \in \{1 \ldots, P\}$, given all the other parameters. Here $\boldsymbol{\theta}_p$ is either column p or row p of matrix $\boldsymbol{\Theta}$. Notice that sampling each triplet requires, at most, operations of the order $\mathcal{O}\left((P+V)^3\right)$.

- *Second step*: Joint sampling of $\boldsymbol{\Theta}, \boldsymbol{\beta}_0, \boldsymbol{B} = [\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_p], \boldsymbol{\beta}_x$ given all the other parameters, most importantly the $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_P)'$ indicators. Since the sampling of each triplets separately has as a consequence chains with poor mixing, we perform an extra sampling step of all the coefficients involved in the regression to improve mixing. This not only improves mixing, but makes the algorithm more stable. For this step, we follow Bhattacharya et al. (2016a), since usually for this applications $n << p$, resulting in computations of the order $\mathcal{O}\left(n^3\right)$ instead of $\mathcal{O}\left(Q^3\right)$.

- *Third step*: In the final step, we sample the remainder of the parameters.

Here, we introduce the following notation. For a matrix $\boldsymbol{M}$ let $\boldsymbol{M}[-i,-j]$ be the sub-matrix of $\boldsymbol{M}$ without row i and column j. In the same way $\boldsymbol{M}[-i,\cdot]$ and $\boldsymbol{M}[\cdot,-j]$ are the sub-matrices of $\boldsymbol{M}$ without row i and column j respectively. In the same way for a vector $\boldsymbol{z}$ let $\boldsymbol{z}[-i]$ the sub-vector of $\boldsymbol{z}$ without entry i. For two boolean vectors $\boldsymbol{\phi}$ and $\boldsymbol{\phi}'$ and a matrix $\boldsymbol{M}$ of adequate sizes, we denote $\boldsymbol{M}[\boldsymbol{\phi}=1,\boldsymbol{\phi}'=1]$ as the sub-Matrix of $\boldsymbol{M}$ composed of the columns and rows for which $\boldsymbol{\phi}=1$ and $\boldsymbol{\phi}'=1$ respectively. In a similar way, $\boldsymbol{M}[\boldsymbol{\phi}=1,\cdot]$ the sub-matrix of $\boldsymbol{M}$ that contains all the columns of $\boldsymbol{M}$ but only the rows for which $\boldsymbol{\phi}=1$, and equivalently $\boldsymbol{M}[\cdot,\boldsymbol{\phi}=1]$. For a boolean vector $\boldsymbol{\phi}$ and a vector $\boldsymbol{z}$ of the same size, we define $\boldsymbol{z}[\boldsymbol{\phi}=1]$ as the sub-vector of $\boldsymbol{z}$ that only contains the corresponding entries for which the entries of $\boldsymbol{\phi}$ are 1. In a similar fashion, let us define $\boldsymbol{z}[\boldsymbol{\phi}=1]$ as a sub-vector $\boldsymbol{z}$ that only contains the corresponding entries for which the entries of $\boldsymbol{\phi}$ are 0.

For the each triplet $\{\beta_p,\theta_p,\xi_p\}$, where $\theta_p=\{\theta_{p,1},\dots\theta_{p,P}\}$. We do so as follows:

$$\begin{aligned}
&p(\beta_p,\theta_p,\xi_p|\beta_0,\beta_x,\{\boldsymbol{\beta}'_p\}_{p'\neq p},\Theta[-p,-p],\Delta,\sigma^2,\eta,\Lambda,\tau^2)\\
&\qquad= p\left(\xi_p\middle|\beta_0,\beta_x,\{\boldsymbol{\beta}'_p\}_{p'\neq p},\Theta[-p,-p],\Delta,\sigma^2,\eta,\Lambda,\tau^2\right)\\
&\qquad\times p\left(\beta_p,\theta_p\middle|\beta_0,\beta_x,\{\boldsymbol{\beta}'_p\}_{p'\neq p},\Theta[-p,-p],\Delta,\sigma^2,\eta,\Lambda,\tau^2,\xi_p\right)
\end{aligned}\tag{B.2}$$

That is, first we integrate β_p,θ_p and sample ξ_p given the remaining parameters. Then we sample β_p,θ_p given all the parameters. Given that ξ_p is Bernoulli distributed, we only need the probability of success. To compute the probability, we find first the odds after integrating out β_p,θ_p, for which we introduce new notation. In this way, for each p, we can re-write the regression equation as follows:

$$r_{i,p} = \boldsymbol{x}_{i,p}^T \boldsymbol{\alpha}_p + \epsilon_i \tag{B.3}$$

where $r_{i,p} = y_i - \beta_0 - \boldsymbol{x}_i^T \boldsymbol{\beta}_{\boldsymbol{x}} - \sum_{p' \neq p} \boldsymbol{g}_{i,p'}^T \boldsymbol{\beta}_{p'} - \langle \boldsymbol{A}_i[-p,-p], \boldsymbol{\Theta}[-p,-p] \rangle / 2$, $\boldsymbol{\alpha} = (\boldsymbol{\beta}_p^T, \boldsymbol{\theta}_p[-p][\boldsymbol{\xi}[-p] = 1]^T)^T$, $\boldsymbol{x}_p = (\boldsymbol{g}_{i,p}^T, \boldsymbol{a}_{i,p}[-p][\boldsymbol{\xi}[-p] = 1]^T)^T$

which we can also express in matrix form.

$$R_p = X_p \boldsymbol{\alpha}_p + \boldsymbol{\epsilon} \tag{B.4}$$

where $R_p = (r_{1,p}, \ldots, r_{n,p})^T$, $X_p = (x_{1,p}, \ldots, x_{n,p})^T$ and $\boldsymbol{\epsilon} = (\epsilon_1, \ldots, \epsilon_n)^T$. Given this representation, we have that the odds are given by:

$$\begin{aligned} &\frac{p\left(\xi_p = 1 \middle| \beta_0, \beta_x, \{\boldsymbol{\beta}'_p\}_{p' \neq p}, \Theta[-p,-p], \Delta, \sigma^2, \eta, \Lambda, \tau^2\right)}{p\left(\xi_p = 0 \middle| \beta_0, \beta_x, \{\boldsymbol{\beta}'_p\}_{p' \neq p}, \Theta[-p,-p], \Delta, \sigma^2, \eta, \Lambda, \tau^2\right)} \\ &\quad = |\boldsymbol{\Psi}_p|^{-\frac{1}{2}} \left| \boldsymbol{X}_p^T \boldsymbol{X}_p + \boldsymbol{\Psi}_p^{-1} \right|^{-\frac{1}{2}} \exp\left\{ -\frac{1}{2\tau^2} \hat{\boldsymbol{\alpha}}_p^T \left(X_p^T X_p + \boldsymbol{\Psi}_p^{-1} \right) \hat{\boldsymbol{\alpha}}_p \right\} \end{aligned} \tag{B.5}$$

where $\hat{\boldsymbol{\alpha}}_p = \left(X_p^T X_p + \boldsymbol{\Psi}_p^{-1} \right)^{-1} X_p^T R_p$ and $\boldsymbol{\Psi}_p$ is a diagonal matrix with diagonal entries
$\left(\Delta_p^2 \eta_p^2, \sigma^2 \lambda_p^2[-p][\boldsymbol{\xi} = 1] \right)$. For $\boldsymbol{\beta}_p, \boldsymbol{\theta}_b$. If $\xi_p = 1$ we have that:

$$\begin{aligned} &\beta_p, \theta_p[-p][\boldsymbol{\xi}[-p] = 1] | \beta_0, \beta_x, \boldsymbol{B}_{-p}, \Theta_{-p,-p}, \Delta, \sigma^2, \eta, \Lambda, \tau^2, \xi_p \\ &\qquad \sim N_{P+V}\left(\hat{\boldsymbol{\alpha}}_p, \left(X_p^T X_p + \boldsymbol{\Psi}_p^{-1} \right)^{-1} \right) \end{aligned} \tag{B.6}$$

and if $\xi_p = 0$, we simply set $\beta_p = 0$ and $\theta_p = 0$. Finally, in every case,

$$\theta_p[-p][\boldsymbol{\xi}[-p]=0]|\beta_0,\beta_x,\boldsymbol{B}_{-p},\Theta_{-p,-p},\Delta,\sigma^2,\eta,\Lambda,\tau^2,\xi_p=0 \tag{B.7}$$

After the first step, we proceed to re-sample $\boldsymbol{B}$ and $\boldsymbol{\Theta}$, from it's full conditional. Notice that given $\boldsymbol{\xi}$ we have that, if $\xi_p = 0$ then $\boldsymbol{\beta}_p = 0$ and $\boldsymbol{\theta}_p[-p][\xi[-p] = 0] = 0$. So we only need to re-sample $\boldsymbol{B}[\cdot, \boldsymbol{\xi} = 1]$ and $\boldsymbol{\Theta}[\boldsymbol{\xi} = 1, \boldsymbol{\xi} = 1]$. Here we use, the vectorized version $\boldsymbol{\alpha}_{\boldsymbol{\xi}} = (\beta_0, \text{vec}(\boldsymbol{B}[\cdot, \boldsymbol{\xi} = 1]), \text{lower}(\boldsymbol{\Theta}[\boldsymbol{\xi} = 1, \boldsymbol{\xi} = 1]))$ of this parameters. Even though we have vectorized the parameters, this vectorized version still conserves some information from the object structure with the lasso structure. The full conditional of $\boldsymbol{\alpha}_{\boldsymbol{\xi}}$ is given by

$$\boldsymbol{\alpha}_{\boldsymbol{\xi}}|\Delta,\sigma^2,\eta,\Lambda,\tau^2 \sim N\left(\hat{\boldsymbol{\alpha}}_{\boldsymbol{\xi}}, (\boldsymbol{X}_{\boldsymbol{\xi}}^T\boldsymbol{X}_{\boldsymbol{\xi}} + \boldsymbol{\Psi}_{\boldsymbol{\xi}})^{-1}\right) \tag{B.8}$$

where

$$\hat{\boldsymbol{\alpha}}_{\boldsymbol{\xi}} = (\boldsymbol{X}_{\boldsymbol{\xi}}^T\boldsymbol{X}_{\boldsymbol{\xi}} + \boldsymbol{\Psi}_{\boldsymbol{\xi}})^{-1}\boldsymbol{X}_{\boldsymbol{\xi}}^T\boldsymbol{y}, \boldsymbol{X}_{\boldsymbol{\xi}} = (\boldsymbol{x}'_{1,\boldsymbol{\xi}}, \ldots, \boldsymbol{x}'_{n,\boldsymbol{\xi}})',$$
$$\boldsymbol{x}_{i,\boldsymbol{\xi}} = (1, \boldsymbol{x}_i, \text{vec}(G_i[\cdot, \boldsymbol{\xi} = 1]), \text{lower}(\boldsymbol{A}_i[\boldsymbol{\xi} = 1, \boldsymbol{\xi} = 1]))',$$
$$\boldsymbol{\Psi}_{\boldsymbol{\xi}} = \text{diag}\left(1, b_x, \text{vec}\left(\left(\Delta_1^2\eta_1, \ldots, \Delta_P^2\eta_P\right)[\cdot, \boldsymbol{\xi} = 1]\right), \text{lower}\left(\Theta[\boldsymbol{\xi} = 1, \boldsymbol{\xi} = 1]\right)\right).$$

Here the sampling follows Bhattacharya et al. (2016a) for efficient computing.

Finally, in the third step, we sample the Horseshoe structure parameters following Makalic and Schmidt (2016) and τ^2 from the usual Inverse Gamma distribution.

Since we double sample the regression coefficient parameters to improve mixing we only save the coefficients sampled after the mixing for the Markov Chain, but

we do not apply any thining to the resulting chain.

Appendix C

A Bayesian Covariance Based Clustering for High Dimensional Tensors

C.1 Convergence Behavior of the Transformed Features

C.1.1 Convergence of the Transformed Features

Lemma C.1.1. *Let* $\boldsymbol{T}_i \sim TN(\mathbf{0}, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K)$ *and* $\mathcal{A}(\boldsymbol{T}_i)^{(k)} = \frac{p_k}{p}\boldsymbol{T}_{i,(k)}\boldsymbol{T}'_{i,(k)}$. *Assume that for all* $k = 1, ..., K$, *(i)* $\frac{p_k}{p} \to 0$ *(ii)* $\frac{p_k}{p} tr(\otimes_{k' \neq k}\boldsymbol{\Sigma}_{k'}) \to w_k$ *and (iii)* $\frac{p_k^2}{p^2}\sum_{l,r}\{\otimes_{k' \neq k}\boldsymbol{\Sigma}_{k'}\}_{l,r}^2 \to 0$, *for all* $l, r = 1, ..., p/p_k$. *(i)-(iii) together imply that* $\{\mathcal{A}(\boldsymbol{T}_i)^{(k)}\}_{l,r} \to \{\boldsymbol{\Sigma}_k\}_{l,r} w_k$, *where* w_k *is a constant.*

Proof. In order to prove the lemma, we will simply show that as $\frac{p_k}{p} \to 0$ we have that $\mathbb{E}\left[\{\mathcal{A}(T_i)^{(k)}\}_{l,r}\right] \to \{\boldsymbol{\Sigma}_k\}_{l,r} w_k$ and $\mathbb{V}\left[\{\mathcal{A}(T_i)^{(k)}\}_{l,r}\right] \to 0$. Note that if $\boldsymbol{T}_i \sim \mathrm{TN}(\mathbf{0}, \boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_K)$ then $\mathrm{vec}(\boldsymbol{T}_{i,(k)}) \sim N(0, \boldsymbol{\Sigma}_k \otimes (\otimes_{k' \neq k}\boldsymbol{\Sigma}_{k'}))$ and $\{\boldsymbol{T}_{i,(k)}\}_{l,r}$

is the entry $(r-1)p_k + l$ of $\text{vec}(\boldsymbol{T}_{i,(k)})$. And also, we have that $\{\mathcal{A}(T_i)^{(k)}\}_{l,r} = \frac{p_k}{p}\sum_{j=1}^{\frac{p_k}{p}}\{\boldsymbol{T}_{i,(k)}\}_{j,l}\{\boldsymbol{T}_{i,(k)}\}_{j,r}$. Using the vectorized representation of $\{\boldsymbol{T}_{i,(k)}\}$ we have that:

$$\begin{aligned}\mathbb{E}\left[\{\boldsymbol{T}_{i,(k)}\}_{j,l}\{\boldsymbol{T}_{i,(k)}\}_{j,r}\right] &= \{\boldsymbol{\Sigma}_k \otimes (\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'})\}_{(l-1)p_k+j,(r-1)p_k+j} \\ &= \{\boldsymbol{\Sigma}_k\}_{l,r}\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,j}\end{aligned}$$

Then we have that:

$$\begin{aligned}\mathbb{E}\left[\{\mathcal{A}(T_i)^{(k)}\}_{l,r}\right] &= \mathbb{E}\left[\frac{p_k}{p}\sum_{j=1}^{\frac{p_k}{p}}\{\boldsymbol{T}_{i,(k)}\}_{j,l}\{\boldsymbol{T}_{i,(k)}\}_{j,r}\right] \\ &= \sum_{j=1}^{\frac{p_k}{p}}\frac{p_k}{p}\mathbb{E}\left[\{\boldsymbol{T}_{i,(k)}\}_{j,l}\{\boldsymbol{T}_{i,(k)}\}_{j,r}\right] \\ &= \sum_{j=1}^{\frac{p_k}{p}}\frac{p_k}{p}\{\boldsymbol{\Sigma}_k\}_{l,r}\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,j} \\ &= \{\boldsymbol{\Sigma}_k\}_{l,r}\frac{p_k}{p}\sum_{j=1}^{\frac{p_k}{p}}\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,j} \\ &= \{\boldsymbol{\Sigma}_k\}_{l,r}\frac{p_k}{p}\text{tr}(\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'})\end{aligned}$$

Then by condition (ii) as $\frac{p}{p_k} \to \infty$, we have that $\mathbb{E}\left[\{\mathcal{A}(T_i)^{(k)}\}_{l,r}\right] \to \{\boldsymbol{\Sigma}_k\}_{l,r}w_k$.

From here we have that the constant w_k is the weighted average of all the variance elements of the covariance matrices but for the ones corresponding to k. In particular if the covariance matrices have unit variance elements, w_k will be

equal to 1 for every k. In a similar way

$$
\begin{aligned}
&\mathbb{E}\left[\{\boldsymbol{T}_{i,(k)}\}_{j,l}\{\boldsymbol{T}_{i,(k)}\}_{j,r}\{\boldsymbol{T}_{i,(k)}\}_{m,l}\{\boldsymbol{T}_{i,(k)}\}_{m,r}\right] \\
&\quad= \{\boldsymbol{\Sigma}_k \otimes (\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'})\}_{(l-1)p_k+j,(r-1)p_k+j}\{\boldsymbol{\Sigma}_k \otimes (\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'})\}_{(l-1)p_k+m,(r-1)p_k+m} \\
&\qquad+ \{\boldsymbol{\Sigma}_k \otimes (\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'})\}_{(l-1)p_k+j,(l-1)p_k+m}\{\boldsymbol{\Sigma}_k \otimes (\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'})\}_{(r-1)p_k+j,(r-1)p_k+m} \\
&\qquad+ \{\boldsymbol{\Sigma}_k \otimes (\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'})\}_{(l-1)p_k+j,(r-1)p_k+m}\{\boldsymbol{\Sigma}_k \otimes (\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'})\}_{(r-1)p_k+j,(l-1)p_k+m} \\
&\quad= \{\boldsymbol{\Sigma}_k\}_{l,r}\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,j}\{\boldsymbol{\Sigma}_k\}_{l,r}\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{m,m} \\
&\qquad+ \{\boldsymbol{\Sigma}_k\}_{l,l}\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,m}\{\boldsymbol{\Sigma}_k\}_{r,r}\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,m} \\
&\qquad+ \{\boldsymbol{\Sigma}_k\}_{l,r}\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,m}\{\boldsymbol{\Sigma}_k\}_{r,l}\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,m} \\
&\quad= \{\boldsymbol{\Sigma}_k\}_{l,r}^2\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,j}\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{m,m} \\
&\qquad+ \left(\{\boldsymbol{\Sigma}_k\}_{l,l}\{\boldsymbol{\Sigma}_k\}_{r,r} + \{\boldsymbol{\Sigma}_k\}_{l,r}^2\right)\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,m}^2
\end{aligned}
$$

Which implies that:

$$
\begin{aligned}
&\mathbb{C}\left[\{\boldsymbol{T}_{i,(k)}\}_{j,l}\{\boldsymbol{T}_{i,(k)}\}_{j,r}, \{\boldsymbol{T}_{i,(k)}\}_{m,l}\{\boldsymbol{T}_{i,(k)}\}_{m,r}\right] \\
&\quad= \mathbb{E}\left[\{\boldsymbol{T}_{i,(k)}\}_{j,l}\{\boldsymbol{T}_{i,(k)}\}_{j,r}\{\boldsymbol{T}_{i,(k)}\}_{m,l}\{\boldsymbol{T}_{i,(k)}\}_{m,r}\right] \\
&\qquad- \mathbb{E}\left[\{\boldsymbol{T}_{i,(k)}\}_{j,l}\{\boldsymbol{T}_{i,(k)}\}_{j,r}\right]\mathbb{E}\left[\{\boldsymbol{T}_{i,(k)}\}_{m,l}\{\boldsymbol{T}_{i,(k)}\}_{m,r}\right] \\
&\quad= \{\boldsymbol{\Sigma}_k\}_{l,r}^2\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,j}\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{m,m} \\
&\qquad+ \left(\{\boldsymbol{\Sigma}_k\}_{l,l}\{\boldsymbol{\Sigma}_k\}_{r,r} + \{\boldsymbol{\Sigma}_k\}_{l,r}^2\right)\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,m}^2 \\
&\qquad- \{\boldsymbol{\Sigma}_k\}_{l,r}^2\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,j}\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{m,m} \\
&\quad= \left(\{\boldsymbol{\Sigma}_k\}_{l,l}\{\boldsymbol{\Sigma}_k\}_{r,r} + \{\boldsymbol{\Sigma}_k\}_{l,r}^2\right)\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,m}^2
\end{aligned}
$$

Then

$$\begin{aligned}
\mathbb{V}\left[\{\mathcal{A}(T_i)^{(k)}\}_{l,r}\right] &= \mathbb{V}\left[\frac{p_k}{p}\sum_{j=1}^{\frac{p_k}{p}}\{\boldsymbol{T}_{i,(k)}\}_{j,l}\{\boldsymbol{T}_{i,(k)}\}_{j,r}\right] \\
&= \frac{p_k^2}{p^2}\sum_{j,m}\mathbb{C}\left[\{\boldsymbol{T}_{i,(k)}\}_{j,l}\{\boldsymbol{T}_{i,(k)}\}_{j,r}, \{\boldsymbol{T}_{i,(k)}\}_{m,l}\{\boldsymbol{T}_{i,(k)}\}_{m,r}\right] \\
&= \frac{p_k^2}{p^2}\sum_{j,m}\left(\{\boldsymbol{\Sigma}_k\}_{l,l}\{\boldsymbol{\Sigma}_k\}_{r,r} + \{\boldsymbol{\Sigma}_k\}_{l,r}^2\right)\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,m}^2 \\
&= \left(\{\boldsymbol{\Sigma}_k\}_{l,l}\{\boldsymbol{\Sigma}_k\}_{r,r} + \{\boldsymbol{\Sigma}_k\}_{l,r}^2\right)\frac{p_k^2}{p^2}\sum_{j,m}\{\otimes_{k'\neq k}\boldsymbol{\Sigma}_{k'}\}_{j,m}^2
\end{aligned}$$

Then by condition (iii) we have that if $\frac{p}{p_k} \to \infty$ then $\mathbb{V}\left[\{\mathcal{A}(T_i)^{(k)}\}_{l,r}\right] \to 0$, concluding our proof. □

Bibliography

Abramovich, F. and V. Grinshtein (2019, May). High-Dimensional Classification by Sparse Logistic Regression. *IEEE Transactions on Information Theory 65*(5), 3068–3079. Conference Name: IEEE Transactions on Information Theory.

Akdemir, D. and A. K. Gupta (2011). Array variate random variables with multiway kronecker delta covariance matrix structure. *Journal of algebraic statistics 2*(1), 98–113.

Albert, J. H. and S. Chib (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association 88*(422), 669–679.

Aldous, D. J. (1981). Representations for partially exchangeable arrays of random variables. *Journal of Multivariate Analysis 11*(4), 581–598.

Aldous, D. J. (1985). Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII 1983*, pp. 1–198. Springer.

Amewou-Atisso, M., S. Ghosal, J. K. Ghosh, R. Ramamoorthi, et al. (2003). Posterior consistency for semi-parametric regression problems. *Bernoulli 9*(2), 291–312.

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 nonparametric problems. *The annals of statistics 2*, 1152–1174.

Armagan, A., D. B. Dunson, and J. Lee (2013). Generalized double pareto shrinkage. *Statistica Sinica 23*(1), 119.

Armagan, A., D. B. Dunson, J. Lee, W. U. Bajwa, and N. Strawn (2013). Posterior consistency in linear models under shrinkage priors. *Biometrika 100*(4), 1011–1018.

Arnaud Doucet, S. G. and C. Andreieu (2000). On sequential monte carlo sampling methods for bayesian filtering. *Statistics and Computing 10*, 197–208.

Arulampalam, M. S., S. Maskell, N. Gordon, and T. Clapp (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on signal processing 50*(2), 174–188.

Bailey, J. F., A. J. Fields, A. Ballatori, D. Cohen, D. Jain, D. Coughlin, C. O'Neill, Z. McCormick, M. Han, R. Krug, S. Demir-Deviren, and J. C. Lotz (2019). The relationship between endplate pathology and patient-reported symptoms for chronic low back pain depends on lumbar paraspinal muscle quality. *Spine (Phila Pa 1976) 44*(14), 1010–1017.

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.

Banerjee, S., B. P. Carlin, and A. E. Gelfand (2014). *Hierarchical modeling and analysis for spatial data*. CRC press.

Banfield, J. D. and A. E. Raftery (1993). Model-Based Gaussian and Non-Gaussian Clustering. *Biometrics 49*(3), 803–821. Publisher: [Wiley, International Biometric Society].

Barbieri, M. M. and J. O. Berger (2004). Optimal predictive model selection. *The annals of statistics 32*(3), 870–897.

Bayles, K. A., C. K. Tomoeda, and J. A. Rein (1996). Phrase repetition in alzheimer's disease: effect of meaning and length. *Brain Lang 54*(2), 246–261.

Belkin, M. and P. Niyogi (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pp. 585–591.

Berman, M. G., J. Jonides, and D. E. Nee (2006). Studying mind and brain with fmri. *Social cognitive and affective neuroscience 1*(2), 158–161.

Beskos, A., D. Crisan, A. Jasra, et al. (2014). On the stability of sequential monte carlo methods in high dimensions. *The Annals of Applied Probability 24*(4), 1396–1445.

Beskos, A., D. O. Crisan, A. Jasra, and N. Whiteley (2014). Error bounds and normalising constants for sequential monte carlo samplers in high dimensions. *Advances in Applied Probability 46*(1), 279–306.

Betancourt, M. (2018). A conceptual introduction to hamiltonian monte carlo. arxiv.org/pdf/1701.02434.pdf.

Bhattacharya, A., A. Chakraborty, and B. K. Mallick (2016a). Fast sampling with Gaussian scale mixture priors in high-dimensional regression. *Biometrika 103*(4), 985–991.

Bhattacharya, A., A. Chakraborty, and B. K. Mallick (2016b, December). Fast sampling with gaussian scale mixture priors in high-dimensional regression. *Biometrika 103*(4), 985–991.

Billio, M., R. Casarin, S. Kaufmann, and M. Iacopini (2018, May). Bayesian Dynamic Tensor Regression. SSRN Scholarly Paper ID 3192340, Social Science Research Network, Rochester, NY.

Bondell, H. D. and B. J. Reich (2012). Consistent high-dimensional Bayesian variable selection via penalized credible regions. *Journal of the American Statistical Association 107*(500), 1610–1624.

Bourotte, M., D. Allard, and E. Porcu (2016). A flexible class of non-separable cross-covariance functions for multivariate space–time data. *Spatial Statistics 18*(A), 125–146.

Breheny, P. and J. Huang (2011). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *Annals of Applied Statistics 5*(1), 232–253.

Brown, J. A., J. Deng, J. Neuhaus, I. J. Sible, A. C. Sias, S. E. Lee, J. Kornak, G. A. Marx, A. M. Karydas, S. Spina, L. T. Grinberg, G. Coppola, D. H. Geschwind, J. H. Kramer, M. L. Gorno-Tempini, B. L. Miller, H. J. Rosen, and W. W. Seeley (2019). Patient-tailored, connectivity-based forecasts of spreading brain atrophy. *Neuron 104*(5), 856–868.

Bullmore, E. and O. Sporns (2009). Complex brain networks: Graph theoretical analysis of structural and functional systems. *Nature Reviews. Neuroscience 10*(3), 186–198.

Butland, G., J. M. Peregrín-Alvarez, J. Li, W. Yang, X. Yang, V. Canadien, A. Starostine, D. Richards, B. Beattie, N. Krogan, et al. (2005). Interaction network containing conserved and essential protein complexes in escherichia coli. *Nature 433*(7025), 531–537.

Caffo, B. S., C. M. Crainiceanu, G. Verduzco, S. Joel, S. H. Mostofsky, S. S. Bassett, and J. J. Pekar (2010). Two-stage decompositions for the analysis of functional connectivity for fmri with application to alzheimer's disease risk. *NeuroImage 51*(3), 1140–1149.

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.

Calhoun, V. D. and J. Sui (2016a). Multimodal fusion of brain imaging data: A key to finding the missing link (s) in complex mental illness. *Biological psychiatry: cognitive neuroscience and neuroimaging 1*(3), 230–244.

Calhoun, V. D. and J. Sui (2016b). Multimodal fusion of brain imaging data: A key to finding the missing link(s) in complex mental illness. *Biol Psychiatry Cogn Neurosci Neuroimaging 1*(3), 230–244.

Campbell, T., J. Straub, J. W. Fisher III, and J. P. How (2015). Streaming, distributed variational inference for bayesian nonparametrics. In *Advances in Neural Information Processing Systems*, pp. 280–288.

Canova, F. and M. Ciccarelli (2004). Forecasting and turning point predictions in a bayesian panel var model. *Journal of Econometrics 120*(2), 327–359.

Canova, F. and M. Ciccarelli (2009). Estimating multicountry var models. *International economic review 50*(3), 929–959.

Canova, F., M. Ciccarelli, and E. Ortega (2007). Similarities and convergence in g-7 cycles. *Journal of Monetary economics 54*(3), 850–878.

Carlos M. Carvalho, N. G. P. and J. G. Scott (2009). Handling sparsity via the horseshoe. In *Proceedings of the 12 th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

Carlos M. Carvalho, Hedibert F. Lopes, N. G. P. and M. A. Taddy (2010). Particle learning for general mixtures. *International Society for Bayesian Analysis 5*(4), 619–650.

Caron, F. and A. Doucet (2008). Sparse bayesian nonparametric regression. In *Proceedings of the 25th international conference on Machine learning*, pp. 88–95. ACM.

Carvalho, C. M., H. F. Lopes, N. G. Polson, M. A. Taddy, et al. (2010). Particle learning for general mixtures. *Bayesian Analysis 5*(4), 709–740.

Carvalho, C. M., N. G. Polson, and J. G. Scott (2010). The horseshoe estimator for sparse signals. *Biometrika 97*(2), 465–480.

Caso, F., M. L. Mandelli, M. Henry, B. Gesierich, B. M. Bettcher, J. Ogar, M. Filippi, G. Comi, G. Magnani, M. Sidhu, J. Q. Trojanowski, E. J. Huang, L. T. Grinberg, B. L. Miller, N. Dronkers, W. W. Seeley, and M. L. Gorno-Tempini

(2014). In vivo signatures of nonfluent/agrammatic primary progressive aphasia caused by ftld pathology. *Neurology 82*(3), 239–247.

Castillo, I., J. Schmidt-Hieber, A. Van der Vaart, et al. (2015). Bayesian linear regression with sparse priors. *The Annals of Statistics 43*(5), 1986–2018.

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.

Chatterjee, A. and S. Lahiri (2010). Asymptotic properties of the residual bootstrap for lasso estimators. *Proceedings of the American Mathematical Society 138*(12), 4497–4509.

Chatterjee, A. and S. N. Lahiri (2011). Bootstrapping lasso estimators. *Journal of the American Statistical Association 106*(494), 608–625.

Chavez, R., A. Graff-Guerrero, J. Garcia-Reyna, V. Vaugier, and C. Cruz-Fuentes (2004). Neurobiology of creativity: Preliminary results from a brain activation study. *Salud Mental 27*(3), 38–46.

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.

Chopin, N. (2002a). A sequential particle filter method for static models. *Biometrika 89*(3), 539–552.

Chopin, N. (2002b). A sequential particle filter method for static models. *Biometrika 89*(3), 539–552.

Chopin, N. et al. (2004). Central limit theorem for sequential monte carlo methods and its application to bayesian inference. *The Annals of Statistics 32*(6), 2385–2411.

Christakis, N. A. and J. H. Fowler (2007). The spread of obesity in a large social network over 32 years. *n engl j med 2007*(357), 370–379.

Christakis, N. A. and J. H. Fowler (2008). The collective dynamics of smoking in a large social network. *New England Journal of Medicine 358*(21), 2249–2258.

Christidis, A.-A., L. Lakshmanan, E. Smucler, and R. Zamar (2020, July). Split Regularized Regression. *Technometrics 62*(3), 330–338. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/00401706.2019.1635533.

Cook, R. D., B. Li, and F. Chiaromonte (2010). ENVELOPE MODELS FOR PARSIMONIOUS AND EFFICIENT MULTIVARIATE LINEAR REGRESSION. *Statistica Sinica 20*(3), 927–960. Publisher: Institute of Statistical Science, Academia Sinica.

Craddock, R. C., P. E. Holtzheimer, X. P. Hu, and H. S. Mayberg (2009). Disease state prediction from resting state functional connectivity. *Magnetic Resonance in Medicine 62*(6), 1619–1628.

Cressie, N. and C. K. Wikle (2015). *Statistics for spatio-temporal data*. John Wiley & Sons.

Dai, X. and L. Li (2021). Orthogonal statistical inference for multimodal data analysis. arXiv preprint arXiv:2103.07088.

Datta, A., S. Banerjee, A. O. Finley, and A. E. Gelfand (2016). Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association 111*(514), 800–812.

De la Haye, K., G. Robins, P. Mohr, and C. Wilson (2010). Obesity-related behaviors in adolescent friendship networks. *Social Networks 32*(3), 161–167.

De Martino, F., A. W. De Borst, G. Valente, R. Goebel, and E. Formisano (2011). Predicting eeg single trial responses with simultaneous fMRI and relevance vector machine regression. *Neuroimage 56*(2), 826–836.

Desikan, R. S., F. Ségonne, B. Fischl, B. T. Quinn, B. C. Dickerson, D. Blacker, R. L. Buckner, A. M. Dale, R. P. Maguire, B. T. Hyman, et al. (2006). An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *Neuroimage 31*(3), 968–980.

Disease, G. ., I. Incidence, and C. Prevalence (2018). Global, regional, and national incidence, prevalence, and years lived with disability for 354 diseases and injuries for 195 countries and territories, 1990-2017: a systematic analysis for the global burden of disease study 2017. *Lancet 392*(10159), 1789–1858.

Doreian, P. (2001). Causality in social network analysis. *Sociological Methods & Research 30*(1), 81–114.

Doucet, A., N. De Freitas, and N. Gordon (2001). An introduction to sequential monte carlo methods. In *Sequential Monte Carlo methods in practice*, pp. 3–14. Springer.

Duan, J. A., M. Guindani, and A. E. Gelfand (2007). Generalized spatial dirichlet process models. *Biometrika 94*(4), 809–825.

Dunson, D. B., A. H. Herring, and S. M. Engel (2008). Bayesian selection and clustering of polymorphisms in functionally related genes. *Journal of the American Statistical Association 103*(482), 534–546.

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

Durante, D. and D. B. Dunson (2014). Nonparametric bayes dynamic modeling of relational data. *Biometrika 101*(4), 883–898.

Durante, D. and D. B. Dunson (2018, March). Bayesian Inference and Testing of Group Differences in Brain Networks. *Bayesian Analysis 13*(1), 29–58. Publisher: International Society for Bayesian Analysis.

Durante, D., D. B. Dunson, and J. T. Vogelstein (2017). Nonparametric Bayes modeling of populations of networks. *Journal of the American Statistical Association 112*(520), 1516–1530.

Eidsvik, J., A. O. Finley, S. Banerjee, and H. Rue (2012). Approximate Bayesian inference for large spatial datasets using predictive process models. *Computational Statistics & Data Analysis 56*(6), 1362–1380.

Engle, R. F. and C. W. Granger (1987). Co-integration and error correction: representation, estimation, and testing. *Econometrica: journal of the Econometric Society 55*, 251–276.

Erdos, P. and A. Rényi (1960). On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences 5*(1), 17–60.

Escobar, M. D. and M. West (1995). Bayesian density estimation and inference using mixtures. *Journal of the american statistical association 90*(430), 577–588.

Fan, J., W. Gong, and Z. Zhu (2019). Generalized high-dimensional trace regression via nuclear norm regularization. *Journal of econometrics 212*(1), 177–202.

Fan, J. and R. Li (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association 96*(456), 1348–1360.

Fan, L., H. Li, J. Zhuo, Y. Zhang, J. Wang, L. Chen, Z. Yang, C. Chu, S. Xie, A. R. Laird, P. T. Fox, S. B. Eickhoff, C. Yu, and T. Jiang (2016). The human brainnetome atlas: A new brain atlas based on connectional architecture. *Cereb Cortex 26*(8), 3508–3526.

Feng, X., T. Li, X. Song, and H. Zhu (2019). Bayesian scalar on image regression with nonignorable nonresponse. *Journal of the American Statistical Association 115*, 1–24.

Ferguson, T. S. (1973a). A bayesian analysis of some nonparametric problems. *The annals of statistics 1*, 209–230.

Ferguson, T. S. (1973b, March). A Bayesian Analysis of Some Nonparametric Problems. *Annals of Statistics 1*(2), 209–230. Publisher: Institute of Mathematical Statistics.

Finkelstein, Y., J. Vardi, and I. Hod (1991). Impulsive artistic creativity as a presentation of transient cognitive alterations. *Behavioral Medicine 17*(2), 91–94.

Finley, A. O., S. Banerjee, and D. W. MacFarlane (2011). A hierarchical model for quantifying forest variables over large heterogeneous landscapes with uncertain forest areas. *Journal of the American Statistical Association 106*(493), 31–48.

Finley, A. O., A. Datta, B. D. Cook, D. C. Morton, H. E. Andersen, and S. Banerjee (2019). Efficient algorithms for Bayesian nearest neighbor Gaussian processes. *Journal of Computational and Graphical Statistics 28*(2), 401–414.

Flaherty, A. W. (2005). Frontotemporal and dopaminergic control of idea generation and creative drive. *Journal of Comparative Neurology 493*(1), 147–153.

Fosdick, B. K. and P. D. Hoff (2015). Testing and modeling dependencies between a network and nodal attributes. *Journal of the American Statistical Association 110*(511), 1047–1056.

Fowler, J. H. and N. A. Christakis (2008). Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study. *British Medical Journal 337*, a2338.

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

Fraley, C. and A. E. Raftery (2002b, June). Model-Based Clustering, Discriminant Analysis, and Density Estimation. *Journal of the American Statistical Association 97*(458), 611–631. Publisher: Taylor & Francis _eprint: https://doi.org/10.1198/016214502760047131.

Frank, O. and D. Strauss (1986). Markov graphs. *Journal of the American Statistical Association 81*(395), 832–842.

Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software 33*(1), 1–22.

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 (2021). Regularized matrix data clustering and its application to image analysis. *Biometrics 77*(3), 890–902. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1111/biom.13354.

Gaspari, G. and S. E. Cohn (1999). Construction of correlation functions in two and three dimensions. *Quarterly Journal of the Royal Meteorological Society 125*(554), 723–757.

Gelfand, A. E. and S. K. Ghosh (1998). Model choice: a minimum posterior predictive loss approach. *Biometrika 85*(1), 1–11.

Gelfand, A. E., H.-J. Kim, C. Sirmans, and S. Banerjee (2003). Spatial modeling with spatially varying coefficient processes. *Journal of the American Statistical Association 98*(462), 387–396.

Gelfand, A. E., A. Kottas, and S. N. MacEachern (2005). Bayesian nonparametric spatial modeling with dirichlet process mixing. *Journal of the American Statistical Association 100*(471), 1021–1035.

Gelfand, A. E., A. M. Schmidt, S. Banerjee, and C. Sirmans (2004). Nonstationary multivariate process modeling through spatially varying coregionalization. *Test 13*(2), 263–312.

Gelman, A., J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin (2014). *Bayesian data analysis*, Volume 2. CRC press Boca Raton, FL.

Genton, M. G. and W. Kleiber (2015). Cross-covariance functions for multivariate geostatistics. *Statistical Science 30*(2), 147–163.

George, E. I. and R. E. McCulloch (1993). Variable selection via Gibbs sampling. *Journal of the American Statistical Association 88*(423), 881–889.

George, E. I. and R. E. McCulloch (1997). Approaches for bayesian variable selection. *Statistica sinica 7*(2), 339–373.

Gertheiss, J., A. Maity, and A.-M. Staicu (2013). Variable selection in generalized functional linear models. *Stat 2*(1), 86–101.

Ghosal, S., J. K. Ghosh, and R. Ramamoorthi (1999). Posterior consistency of dirichlet mixtures in density estimation. *Ann. Statist 27*(1), 143–158.

Ghosal, S., A. Roy, et al. (2006). Posterior consistency of gaussian process prior for nonparametric binary regression. *The Annals of Statistics 34*(5), 2413–2429.

Gilbert, E. N. (1959). Random graphs. *The Annals of Mathematical Statistics 30*(4), 1141–1144.

Gneiting, T., W. Kleiber, and M. Schlather (2010). Matérn cross-covariance functions for multivariate random fields. *Journal of the American Statistical Association 105*(491), 1167–1177.

Gneiting, T. and A. E. Raftery (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association 102*(477), 359–378.

Goldsmith, J., J. Bobb, C. M. Crainiceanu, B. Caffo, and D. Reich (2011). Penalized functional regression. *Journal of Computational and Graphical Statistics 20*(4), 830–851.

Goldsmith, J., L. Huang, and C. M. Crainiceanu (2014). Smooth scalar-on-image regression via spatial Bayesian variable selection. *Journal of Computational and Graphical Statistics 23*(1), 46–64.

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

Gorno-Tempini, M. L., S. M. Brambati, V. Ginex, J. Ogar, N. F. Dronkers, A. Marcone, D. Perani, V. Garibotto, S. F. Cappa, and B. L. Miller (2008). The logopenic/phonological variant of primary progressive aphasia. *Neurology 71*(16), 1227–1234.

Gorno-Tempini, M. L., A. E. Hillis, S. Weintraub, A. Kertesz, M. Mendez, S. F. Cappa, J. M. Ogar, J. D. Rohrer, S. Black, B. F. Boeve, F. Manes, N. F. Dronkers, R. Vandenberghe, K. Rascovsky, K. Patterson, B. L. Miller, D. S. Knopman, J. R. Hodges, M. M. Mesulam, and M. Grossman (2011). Classification of primary progressive aphasia and its variants. *Neurology 76*(11), 1006–1014.

Griffin, J. E., P. J. Brown, et al. (2010). Inference with normal-gamma prior distributions in regression problems. *Bayesian Analysis 5*(1), 171–188.

Guerrero, R., R. Wolz, and D. Rueckert (2011). Laplacian eigenmaps manifold learning for landmark localization in brain MR images. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 566–573. Springer.

Guha, S. and R. Guhaniyogi (2021, April). Bayesian Generalized Sparse Symmetric Tensor-on-Vector Regression. *Technometrics 63*(2), 160–170.

Guha, S. and A. Rodriguez (2020). High dimensional bayesian network classification with network global-local shrinkage priors. arXiv preprint arXiv:2009.11401.

Guhaniyogi, R. (2017). Convergence rate of Bayesian supervised tensor modeling with multiway shrinkage priors. *Journal of Multivariate Analysis 160*, 157–168.

Guhaniyogi, R. (2020). Bayesian methods for tensor regressions. Technical report, UCSC Technical Report.

Guhaniyogi, R. and D. B. Dunson (2016). Compressed Gaussian process for manifold regression. *The Journal of Machine Learning Research 17*(1), 2472–2497.

Guhaniyogi, R., A. O. Finley, S. Banerjee, and A. E. Gelfand (2011). Adaptive Gaussian predictive process models for large spatial datasets. *Environmetrics 22*(8), 997–1007.

Guhaniyogi, R., S. Qamar, and D. B. Dunson (2013). Bayesian conditional density filtering. *Journal of Computational and Graphical Statistics 27*(3), 657–672. Publisher: Taylor & Francis _eprint: https://doi.org/10.1080/10618600.2017.1422431.

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. 2, 13.

Guhaniyogi, R. and D. Spencer (2019). Bayesian tensor response regression with an application to brain activation studies. Technical report, UCSC Technical report: UCSC-SOE-18-15.

Guillas, S. and M.-J. Lai (2010). Bivariate splines for spatial functional regression models. *Journal of Nonparametric Statistics 22*(4), 477–497.

Gunawan, D., K.-D. Dang, M. Quiroz, R. Kohn, and M.-N. Tran (2018). Subsampling sequential monte carlo for static bayesian models. arxiv.org/pdf/1805.03317.pdf.

Guo, C. C., M. L. Gorno-Tempini, B. Gesierich, M. Henry, A. Trujillo, T. Shany-Ur, J. Jovicich, S. D. Robinson, J. H. Kramer, K. P. Rankin, B. L. Miller, and W. W. Seeley (2013). Anterior temporal lobe degeneration produces widespread network-driven dysfunction. *Brain 136*(Pt 10), 2979–2991.

Hackbusch, W. (2012). *Tensor spaces and numerical tensor calculus*, Volume 42. Springer Science & Business Media.

Hager, W. W. (1989). Updating the inverse of a matrix. *SIAM review 31*(2), 221–239.

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.

Hanneke, S., W. Fu, E. P. Xing, et al. (2010). Discrete temporal models of social networks. *Electronic Journal of Statistics 4*, 585–605.

Hans, C. (2009). Bayesian lasso regression. *Biometrika 96*(4), 835–845.

Harrison, L. M. and G. G. Green (2010). A Bayesian spatiotemporal model for very large data sets. *NeuroImage 50*(3), 1126–1141.

Harshman, R. A. and M. E. Lundy (1994). PARAFAC: Parallel factor analysis. *Computational Statistics & Data Analysis 18*(1), 39–72.

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].

Hasenstab, K., A. Scheffler, D. Telesca, C. A. Sugar, S. Jeste, C. DiStefano, and D. Şentürk (2017). A multi-dimensional functional principal components analysis of EEG data. *Biometrics 73*(3), 999–1009.

Hasenstab, K., C. Sugar, D. Telesca, S. Jeste, and D. Şentürk (2016). Robust functional clustering of erp data with application to a study of implicit learning in autism. *Biostatistics 17*(3), 484–498.

Hastie Trevor, J. and J. Tibshirani Robert (1990). Generalized additive models. vol. 43.

Henry, M. L., S. M. Wilson, M. C. Babiak, M. L. Mandelli, P. M. Beeson, Z. A. Miller, and M. L. Gorno-Tempini (2016). Phonological processing in primary progressive aphasia. *J Cogn Neurosci 28*(2), 210–222.

Hoff, P. (2008). Modeling homophily and stochastic equivalence in symmetric relational data. In *Advances in neural information processing systems*, pp. 657–664.

Hoff, P. D. (2005). Bilinear mixed-effects models for dyadic data. *Journal of the American Statistical Association 100*(469), 286–295.

Hoff, P. D. (2009). A hierarchical eigenmodel for pooled covariance estimation. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71*(5), 971–992.

Hoff, P. D. (2011). Hierarchical multilinear models for multiway data. *Computational Statistics & Data Analysis 55*(1), 530–543.

Hoff, P. D. (2015). Multilinear tensor regression for longitudinal relational data. *The Annals of Applied Statistics 9*(3), 1169.

Hoff, P. D., A. E. Raftery, and M. S. Handcock (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association 97*(460), 1090–1098.

Hoffman, M., F. R. Bach, and D. M. Blei (2010). Online learning for latent dirichlet allocation. In *advances in neural information processing systems*, pp. 856–864.

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.

Hou, M., Y. Wang, and B. Chaib-draa (2015). Online local Gaussian process for tensor-variate regression: Application to fast reconstruction of limb movements from brain signal. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5490–5494. IEEE.

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.

Huang, L., J. Goldsmith, P. T. Reiss, D. S. Reich, and C. M. Crainiceanu (2013). Bayesian scalar-on-image regression with application to association between intracranial dti and cognitive outcomes. *NeuroImage 83*, 210–223.

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

Hunt, M. K., D. R. Hopko, R. Bare, C. Lejuez, and E. Robinson (2005). Construct validity of the balloon analog risk task (BART) associations with psychopathy and impulsivity. *Assessment 12*(4), 416–428.

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.

Imaizumi, M. and K. Hayashi (2016). Doubly decomposing nonparametric tensor regression. In *International Conference on Machine Learning*, pp. 727–736.

Ishwaran, H. and L. F. James (2001). Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association 96*(453), 161–173.

Ishwaran, H. and L. F. James (2002). Approximate dirichlet process computing in finite normal mixtures: smoothing and prior information. *Journal of Computational and Graphical statistics 11*(3), 508–532.

Ishwaran, H. and J. S. Rao (2005). Spike and slab variable selection: Frequentist and Bayesian strategies. *Annals of Statistics 33*(2), 730–773.

James, G. M., J. Wang, J. Zhu, et al. (2009). Functional linear regression that?s interpretable. *The Annals of Statistics 37*(5A), 2083–2108.

Jansen, M., T. P. White, K. J. Mullinger, E. B. Liddle, P. A. Gowland, S. T. Francis, R. Bowtell, and P. F. Liddle (2012). Motion-related artefacts in EEG predict neuronally plausible patterns of activation in fMRI data. *Neuroimage 59*(1), 261–270.

Javanmard, A. and A. Montanari (2014). Confidence intervals and hypothesis testing for high-dimensional regression. *The Journal of Machine Learning Research 15*(1), 2869–2909.

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

Jung, R. E., J. M. Segall, H. Jeremy Bockholt, R. A. Flores, S. M. Smith, R. S. Chavez, and R. J. Haier (2010). Neuroanatomy of creativity. *Human Brain Mapping 31*(3), 398–409.

Kalus, S., P. G. Sämann, and L. Fahrmeir (2014). Classification of brain activation via spatial Bayesian variable selection in fMRI regression. *Advances in Data Analysis and Classification 8*(1), 63–83.

Kang, J., B. J. Reich, and A.-M. Staicu (2018). Scalar-on-image regression via the soft-thresholded Gaussian process. *Biometrika 105*(1), 165–184.

Kiar, G., K. J. Gorgolewski, D. Kleissas, W. G. Roncal, B. Litt, B. Wandell, R. A. Poldrack, M. Wiener, R. J. Vogelstein, R. Burns, et al. (2017). Science in the cloud (sic): A use case in MRI connectomics. *Giga Science 6*(5), 1–10.

Kiar, G., W. Gray Roncal, D. Mhembere, E. Bridgeford, R. Burns, and J. Vogelstein (2016). ndmg: Neurodata's MRI graphs pipeline.

Kim, S., P. Smyth, and H. Stern (2006). A nonparametric Bayesian approach to detecting spatial activation patterns in fMRI data. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 217–224. Springer.

Kiuru, N., W. J. Burk, B. Laursen, K. Salmela-Aro, and J.-E. Nurmi (2010). Pressure to drink but not to smoke: Disentangling selection and socialization in adolescent peer networks and peer groups. *Journal of adolescence 33*(6), 801–812.

Kolaczyk, E. D. and G. Csárdi (2014). *Statistical analysis of network data with R*, Volume 65. Springer.

Kolb, B. and B. Milner (1981). Performance of complex arm and facial movements after focal brain lesions. *Neuropsychologia 19*(4), 491–503.

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

Kottas, A., J. A. Duan, and A. E. Gelfand (2008). Modeling disease incidence data with spatial and spatio temporal dirichlet process mixtures. *Biometrical Journal: Journal of Mathematical Methods in Biosciences 50*(1), 29–42.

Krivitsky, P. N. and M. S. Handcock (2014). A separable model for dynamic networks. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76*(1), 29–46.

Kyung, M., J. Gill, M. Ghosh, G. Casella, et al. (2010). Penalized regression, standard errors, and bayesian lassos. *Bayesian Analysis 5*(2), 369–411.

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

Lau, J. W. and P. J. Green (2007b). Bayesian model-based clustering procedures. *Journal of Computational and Graphical Statistics 16*(3), 526–558.

Lazar, N. (2008). *The statistical analysis of functional MRI data*. Springer Science & Business Media.

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.

Lejuez, C., W. M. Aklin, H. A. Jones, J. B. Richards, D. R. Strong, C. W. Kahler, and J. P. Read (2003). The balloon analogue risk task (BART) differentiates smokers and nonsmokers. *Experimental and clinical psychopharmacology 11*(1), 26.

Lejuez, C. W., W. M. Aklin, M. J. Zvolensky, and C. M. Pedulla (2003). Evaluation of the balloon analogue risk task (BART) as a predictor of adolescent real-world risk-taking behaviours. *Journal of adolescence 26*(4), 475–479.

Lejuez, C. W., J. P. Read, C. W. Kahler, J. B. Richards, S. E. Ramsey, G. L. Stuart, D. R. Strong, and R. A. Brown (2002). Evaluation of a behavioral measure of risk taking: the balloon analogue risk task (bart). *Journal of Experimental Psychology: Applied 8*(2), 75.

Li, F., T. Zhang, Q. Wang, M. Z. Gonzalez, E. L. Maresh, and J. A. Coan (2015). Spatial Bayesian variable selection and grouping for high-dimensional scalar-on-image regression. *The Annals of Applied Statistics 9*(2), 687–713.

Li, H. and D. Pati (2017). Variable selection using shrinkage priors. *Computational Statistics & Data Analysis 107*, 107–119.

Li, L., J. Kang, S. N. Lockhart, J. Adams, and W. J. Jagust (2018). Spatially adaptive varying correlation analysis for multimodal neuroimaging data. *IEEE transactions on medical imaging 38*(1), 113–123.

Li, L. and X. Zhang (2017). Parsimonious tensor response regression. *Journal of the American Statistical Association 112*(519), 1131–1146.

Li, Q. and L. Li (2021). Integrative factor regression and its inference for multimodal data analysis. *Journal of the American Statistical Association 0*, 1–15.

Li, X., D. Xu, H. Zhou, and L. Li (2018). Tucker tensor regression and neuroimaging analysis. *Statistics in Biosciences 10*(3), 520–545.

Li, Y., Y. Qin, X. Chen, and W. Li (2013). Exploring the functional brain network of alzheimer's disease: based on the computational experiment. *PloS one 8*(9), e73186.

Lin, X. (2010). Identifying peer effects in student academic achievement by spatial autoregressive models with group unobservables. *Journal of Labor Economics 28*(4), 825–860.

Lindquist, M. A. (2008). The statistical analysis of fmri data. *Statistical science 23*(4), 439–464.

Lindsten, F., A. M. Johansen, C. A. Naesseth, B. Kirkpatrick, T. B. Schön, J. Aston, and A. Bouchard-Côté (2017). Divide-and-conquer with sequential monte carlo. *Journal of Computational and Graphical Statistics 26*(2), 445–458.

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.

Lock, E. F., K. A. Hoadley, J. S. Marron, and A. B. Nobel (2013). Joint and individual variation explained (jive) for integrated analysis of multiple data types. *The annals of applied statistics 7*(1), 523.

Lock, E. F. and G. Li (2018). Supervised multiway factorization. *Electronic journal of statistics 12*(1), 1150.

Lopes, H. F. and R. S. Tsay (2011). Particle filters and bayesian inference in financial econometrics. *Journal of Forecasting 30*(1), 168–209.

Lukic, S., M. L. Mandelli, A. Welch, K. Jordan, W. Shwe, J. Neuhaus, Z. Miller, H. I. Hubbard, M. Henry, B. L. Miller, N. F. Dronkers, and M. L. Gorno-Tempini (2019). Neurocognitive basis of repetition deficits in primary progressive aphasia. *Brain Lang 194*, 35–45.

Lusher, D., J. Koskinen, and G. Robins (2012). *Exponential random graph models for social networks: Theory, methods, and applications*. Cambridge University Press.

Luts, J., M. P. Wand, et al. (2015). Variational inference for count response semiparametric regression. *Bayesian Analysis 10*(4), 991–1023.

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 0*(0), 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.

Majumdar, A. and A. E. Gelfand (2007). Multivariate spatial modeling for geostatistical data using convolved covariance functions. *Mathematical Geology 39*(2), 225–245.

Makalic, E. and D. F. Schmidt (2016, January). A Simple Sampler for the Horseshoe Estimator. *IEEE Signal Processing Letters 23*(1), 179–182. Conference Name: IEEE Signal Processing Letters.

Mandelli, M. L., E. Caverzasi, R. J. Binney, M. L. Henry, I. Lobach, N. Block, B. Amirbekian, N. Dronkers, B. L. Miller, R. G. Henry, and M. L. Gorno-Tempini (2014). Frontal white matter tracts sustaining speech production in primary progressive aphasia. *J Neurosci 34*(29), 9754–9767.

Mandelli, M. L., E. Vilaplana, J. A. Brown, H. I. Hubbard, R. J. Binney, S. Attygalle, M. A. Santos-Santos, Z. A. Miller, M. Pakvasa, M. L. Henry, H. J. Rosen, R. G. Henry, G. D. Rabinovici, B. L. Miller, W. W. Seeley, and M. L. Gorno-Tempini (2016). Healthy brain connectivity predicts atrophy progression in non-fluent variant of primary progressive aphasia. *Brain 139*(Pt 10), 2778–2791.

Matias-Guiu, J. A., J. Díaz-Álvarez, J. L. Ayala, J. L. Risco-Martín, T. Moreno-Ramos, V. Pytel, J. Matias-Guiu, J. L. Carreras, and M. N. Cabrera-Martín (2018). Clustering analysis of FDG-PET imaging in primary progressive aphasia. *Frontiers in Aging Neuroscience 10*, 230.

McCulloch, E. I. G. R. E. (1993). Variable selection via gibbs sampling. *Journal of the American Statistical Association 88*(423), 881–889.

McMillan, C. T., D. J. Irwin, B. B. Avants, J. Powers, P. A. Cook, J. B. Toledo, E. McCarty Wood, V. M. Van Deerlin, V. M.-Y. Lee, J. Q. Trojanowski, and M. Grossman (2013). White matter imaging helps dissociate tau from TDP-43 in frontotemporal lobar degeneration. *J Neurol Neurosurg Psychiatry 84*(9), 949–955.

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.

Mesulam, M. (2008). Primary progressive aphasia pathology. *Annals of Neurology 63*(1), 124–125.

Miller, L. and B. Milner (1985). Cognitive risk-taking after frontal or temporal lobectomy-II. The synthesis of phonemic and semantic information. *Neuropsychologia 23*(3), 371–379.

Miller, Z. A., M. L. Mandelli, K. P. Rankin, M. L. Henry, M. C. Babiak, D. T. Frazier, I. V. Lobach, B. M. Bettcher, T. Q. Wu, G. D. Rabinovici, N. R. Graff-Radford, B. L. Miller, and M. L. Gorno-Tempini (2013). Handedness and language learning disability differentially distribute in progressive aphasia variants. *Brain 136*(Pt 11), 3461–3473.

Miranda, M. F., H. Zhu, and J. G. Ibrahim (2018). TPRM: Tensor partition regression models with applications in imaging biomarker detection. *The annals of applied statistics 12*(3), 1422.

Mo, Q., R. Shen, C. Guo, M. Vannucci, K. S. Chan, and S. G. Hilsenbeck (2018). A fully Bayesian latent variable model for integrative clustering analysis of multi-type omics data. *Biostatistics 19*(1), 71–86.

Moral, P. D., A. Jasra, and Y. Zhou (2017). Biased online parameter inference for state-space models. *Methodology and Computing in Applied Probability 19*(3), 727–749.

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

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo 2*(11), 2.

Newman, M. (2010). *Networks: An Introduction*. Oxford University Press.

Newman, M. E. (2003). The structure and function of complex networks. *SIAM review 45*(2), 167–256.

Nishihara, R., I. Murray, and R. P. Adams (2014). Parallel mcmc with generalized elliptical slice sampling. *The Journal of Machine Learning Research 15*(1), 2087–2112.

Nobile, A. and A. T. Fearnside (2007, May). Bayesian finite mixtures with an unknown number of components: The allocation sampler. *Statistics and Computing 17*(2), 147–162.

Nowicki, K. and T. A. B. Snijders (2001). Estimation and prediction for stochastic block structures. *Journal of the American Statistical Association 96*(455), 1077–1087.

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.

Ossenkoppele, R., N. D. Prins, Y. A. L. Pijnenburg, A. W. Lemstra, W. M. van der Flier, S. F. Adriaanse, A. D. Windhorst, R. L. H. Handels, C. A. G. Wolfs, P. Aalten, F. R. J. Verhey, M. M. Verbeek, M. A. van Buchem, O. S. Hoekstra, A. A. Lammertsma, P. Scheltens, and B. N. M. van Berckel (2013). Impact of molecular imaging on the diagnostic process in a memory clinic. *Alzheimers Dement 9*(4), 414–421.

Ossenkoppele, R., D. R. Schonhaut, S. L. Baker, J. P. O'Neil, M. Janabi, P. M. Ghosh, M. Santos, Z. A. Miller, B. M. Bettcher, M. L. Gorno-Tempini, B. L. Miller, W. J. Jagust, and G. D. Rabinovici (2015). Tau, amyloid, and hypometabolism in a patient with posterior cortical atrophy. *Ann Neurol 77*(2), 338–342.

Ossenkoppele, R., D. R. Schonhaut, M. Schöll, S. N. Lockhart, N. Ayakta, S. L. Baker, J. P. O'Neil, M. Janabi, A. Lazaris, A. Cantwell, J. Vogel, M. Santos, Z. A. Miller, B. M. Bettcher, K. A. Vossel, J. H. Kramer, M. L. Gorno-Tempini, B. L. Miller, W. J. Jagust, and G. D. Rabinovici (2016). Tau PET patterns mirror clinical and neuroanatomical variability in Alzheimer's disease. *Brain 139*(Pt 5), 1551–1567.

Pal, S., K. Khare, et al. (2014). Geometric ergodicity for bayesian shrinkage models. *Electronic Journal of Statistics 8*(1), 604–645.

Pan, W. and X. Shen (2007). Penalized model-based clustering with application to variable selection. *Journal of Machine Learning Research 8*(5), 1145–1164.

Park, T. and G. Casella (2008a). The bayesian lasso. *Journal of the American Statistical Association 103*(482), 681–686.

Park, T. and G. Casella (2008b). The Bayesian lasso. *Journal of the American Statistical Association 103*(482), 681–686.

Penny, W. D., N. J. Trujillo-Barreto, and K. J. Friston (2005). Bayesian fMRI time series analysis with spatial priors. *NeuroImage 24*(2), 350–362.

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.

Pierre Del Moral, A. D. and A. Jasra (2006). Sequential monte carlo samplers. *Journal Royal Statistical Society 68*(3), 411–436.

Pitman, J. (1995). Exchangeable and partially exchangeable random partitions. *Probability theory and related fields 102*(2), 145–158.

Polson, N. G. and J. G. Scott (2010). Shrink globally, act locally: Sparse bayesian regularization and prediction. *Bayesian statistics 9*, 501–538.

Polson, N. G., J. G. Scott, and J. Windle (2013). Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association 108*(504), 1339–1349.

Rabusseau, G. and H. Kadri (2016). Low-rank regression with tensor responses. In *Advances in Neural Information Processing Systems*, pp. 1867–1875.

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

Ramasamy, A., D. Trabzuni, S. Guelfi, V. Varghese, C. Smith, R. Walker, T. De, J. Hardy, M. Ryten, M. E. Weale, et al. (2014). Genetic variability in the regulation of gene expression in ten regions of the human brain. *Nature neuroscience 17*(10), 1418.

Raskutti, G., M. Yuan, H. Chen, et al. (2019). Convex regularization for high-dimensional multiresponse tensor regression. *The Annals of Statistics 47*(3), 1554–1584.

Ravikumar, P., J. Lafferty, H. Liu, and L. Wasserman (2009). Sparse additive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 71*(5), 1009–1030.

Ray, S. and B. Mallick (2006). Functional clustering by Bayesian wavelet methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68*(2), 305–332.

Razumnikova, O. M. (2007). Creativity related cortex activity in the remote associates task. *Brain Research Bulletin 73*(1), 96–102.

Rebeschini, P. and R. v. Handel (2015). Can local particle filters beat the curse of dimensionality? *Annals of Applied Probability 25*(5), 2809–2866. Publisher: Institute of Mathematical Statistics.

Reiss, P. T., L. Huo, Y. Zhao, C. Kelly, and R. T. Ogden (2015). Wavelet-domain regression and predictive inference in psychiatric neuroimaging. *The annals of applied statistics 9*(2), 1076.

Reiss, P. T. and R. T. Ogden (2010). Functional generalized linear models with images as predictors. *Biometrics 66*(1), 61–69.

Relión, J. D. A., D. Kessler, E. Levina, and S. F. Taylor (2019). Network classification with applications to brain connectomics. *The Annals of Applied Statistics 13*(3), 1648–1677.

Richiardi, J., H. Eryilmaz, S. Schwartz, P. Vuilleumier, and D. Van De Ville (2011). Decoding brain states from fMRI connectivity graphs. *Neuroimage 56*(2), 616–626.

Rigon, T., A. Herring, and D. Dunson (2020). A generalized bayes framework for probabilistic clustering. http://arxiv.org/abs/2006.05451.

Robins, G., T. Snijders, P. Wang, M. Handcock, and P. Pattison (2007). Recent developments in exponential random graph (p*) models for social networks. *Social Networks 29*(2), 192–215.

Ročková, V. and E. I. George (2018). The spike-and-slab lasso. *Journal of the American Statistical Association 113*(521), 431–444.

Rodríguez, A., D. B. Dunson, and A. E. Gelfand (2009). Bayesian nonparametric functional data analysis through density estimation. *Biometrika 96*(1), 149–162.

Rodríguez, A., D. B. Dunson, and A. E. Gelfand (2010). Latent stick-breaking processes. *Journal of the American Statistical Association 105*(490), 647–659.

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.

Rousseau, J. and K. Mengersen (2011). Asymptotic behaviour of the posterior distribution in overfitted mixture models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 73*(5), 689–710.

Rue, H. (2001). Fast sampling of gaussian markov random fields. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 63*(2), 325–338.

Rue, H. and L. Held (2005). *Gaussian Markov random fields: theory and applications*. CRC press.

Santos-Santos, M. A., M. L. Mandelli, R. J. Binney, J. Ogar, S. M. Wilson, M. L. Henry, H. I. Hubbard, M. Meese, S. Attygalle, L. Rosenberg, M. Pakvasa, J. Q. Trojanowski, L. T. Grinberg, H. Rosen, A. L. Boxer, B. L. Miller, W. W. Seeley, and M. L. Gorno-Tempini (2016). Features of patients with nonfluent/agrammatic primary progressive aphasia with underlying progressive supranuclear palsy pathology or corticobasal degeneration. *JAMA Neurol 73*(6), 733–742.

Sato, M.-A. (2001). Online model selection based on the variational bayes. *Neural computation 13*(7), 1649–1681.

Scheffler, A., D. Telesca, Q. Li, C. A. Sugar, C. Distefano, S. Jeste, and D. Şentürk (2018, 08). Hybrid principal components analysis for region-referenced longitudinal functional EEG data. *Biostatistics 21*(1), 139–157.

Scheffler, A. W., A. Dickinson, C. DiStefano, S. Jeste, and D. Şentürk (2020). Covariate-adjusted hybrid principal components analysis. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pp. 391–404. Springer.

Schonberg, T., C. R. Fox, J. A. Mumford, E. Congdon, C. Trepel, and R. A. Poldrack (2012). Decreasing ventromedial prefrontal cortex activity during sequential risk-taking: an fmri investigation of the balloon analog risk task. *Frontiers in neuroscience 6*, 80.

Schotman, P. and H. K. Van Dijk (1991). A Bayesian analysis of the unit root in real exchange rates. *Journal of Econometrics 49*(1-2), 195–238.

Schweizer, N. (2012). *Non-asymptotic error bounds for sequential MCMC methods in multimodal settings*. phdthesis, Rheinische Friedrich-Wilhelms-Universität Bonn.

Scott, J. G. and J. O. Berger (2010). Bayes and empirical-bayes multiplicity adjustment in the variable-selection problem. *The Annals of Statistics 38*(5), 2587–2619.

Sethuraman, J. (1994). A constructive definition of Dirichlet priors. *Statistica sinica 4*(2), 639–650.

Shalizi, C. R. and A. C. Thomas (2011). Homophily and contagion are generically confounded in observational social network studies. *Sociological methods & research 40*(2), 211–239.

Shoham, D. A., R. Hammond, H. Rahmandad, Y. Wang, and P. Hovmand (2015). Modeling social norms and social influence in obesity. *Current Epidemiology Reports 2*(1), 71–79.

Sidén, P., A. Eklund, D. Bolin, and M. Villani (2017). Fast Bayesian whole-brain fMRI analysis with spatial 3D priors. *NeuroImage 146*, 211–225.

Snijders, T., C. Steglich, and M. Schweinberger (2007). *Modeling the coevolution of networks and behavior*. https://s3.amazonaws.com/academia.edu.documents.

Snijders, T. A. (2001). The statistical evaluation of social network dynamics. *Sociological Methodology 31*(1), 361–395.

Snyder, C., T. Bengtsson, P. Bickel, and J. Anderson (2008). Obstacles to high-dimensional particle filtering. *Monthly Weather Review 136*(12), 4629–4640.

Song, Q. and F. Liang (2017). Nearly optimal bayesian shrinkage for high dimensional regression. arXiv preprint arXiv:1712.08964.

Sowell, E. R., P. M. Thompson, S. E. Welcome, A. L. Henkenius, A. W. Toga, and B. S. Peterson (2003). Cortical abnormalities in children and adolescents with attention-deficit hyperactivity disorder. *The Lancet 362*(9397), 1699–1707.

Spencer, D., R. Guhaniyogi, and R. Prado (2020). Joint bayesian estimation of voxel activation and inter-regional connectivity in fmri experiments. *Psychometrika 85*, 845–869.

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

Stuss, D., P. Ely, H. Hugenholtz, M. Richard, S. LaRochelle, C. Poirier, and I. Bell (1985). Subtle neuropsychological deficits in patients with good recovery after closed head injury. *Neurosurgery 17*(1), 41–47.

Suarez, A. J. and S. Ghosal (2016). Bayesian clustering of functional data using local features. *Bayesian Analysis 11*(1), 71–98.

Sui, J., T. Adali, Q. Yu, J. Chen, and V. D. Calhoun (2012). A review of multivariate methods for multimodal fusion of brain imaging data. *Journal of neuroscience methods 204*(1), 68–81.

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 (2017). Store: sparse tensor response regression and neuroimaging analysis. *The Journal of Machine Learning Research 18*(1), 4908–4944.

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.

Suzuki, T. (2015). Convergence rate of bayesian tensor estimator and its minimax optimality. In *International Conference on Machine Learning*, pp. 1273–1282.

Szabó, B., A. van der Vaart, J. van Zanten, et al. (2015). Frequentist coverage of adaptive nonparametric bayesian credible sets. *The Annals of Statistics 43*(4), 1391–1428.

Tadesse, M. G., N. Sha, and M. Vannucci (2005). Bayesian variable selection in clustering high-dimensional data. *Journal of the American Statistical Association 100*(470), 602–617.

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

Tenenbaum, J. B. and W. T. Freeman (2000). Separating style and content with bilinear models. *Neural computation 12*(6), 1247–1283.

Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological) 58*(1), 267–288.

Turkstra, L. (2011). *Western Aphasia Battery*. Springer New York.

Valera, E. M., S. V. Faraone, K. E. Murray, and L. J. Seidman (2007). Meta-analysis of structural imaging findings in attention-deficit/hyperactivity disorder. *Biological psychiatry 61*(12), 1361–1369.

Van de Geer, S., P. Bühlmann, Y. Ritov, R. Dezeure, et al. (2014). On asymptotically optimal confidence regions and tests for high-dimensional models. *The Annals of Statistics 42*(3), 1166–1202.

van der Pas, S., J. Scott, A. Chakraborty, and A. Bhattacharya (2019). *horseshoe: Implementation of the Horseshoe Prior*. Leiden University. R package version 0.2.0.

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

Viroli, C. et al. (2011). Model based clustering for three-way data structures. *Bayesian Analysis 6*(4), 573–602.

Wackernagel, H. (2013). *Multivariate geostatistics: an introduction with applications*. Springer Science & Business Media.

Wang, H. (2012). Bayesian graphical lasso models and efficient posterior computation. *Bayesian Analysis 7*(4), 867–886.

Wang, L., D. Durante, R. E. Jung, and D. B. Dunson (2017). Bayesian network–response regression. *Bioinformatics 33*(12), 1859–1866.

Wang, M. and Y. Zeng (2019). Multiway clustering via tensor block models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, Volume 32. Curran Associates, Inc.

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.

Wang, X., D. Dunson, and C. Leng (2016, December). DECOrrelated feature space partitioning for distributed sparse regression. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, Red Hook, NY, USA, pp. 802–810. Curran Associates Inc.

Wang, X., H. Zhu, and A. D. N. Initiative (2017). Generalized scalar-on-image regression models via total variation. *Journal of the American Statistical Association 112*(519), 1156–1168.

Wang, Y., P.-M. Jodoin, F. Porikli, J. Konrad, Y. Benezeth, and P. Ishwar (2014). Cdnet 2014: an expanded change detection benchmark dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pp. 387–394.

Ward, M. D., J. S. Ahlquist, and A. Rozenas (2013). Gravity's rainbow: A dynamic latent space model for the world trade network. *Network Science 1*(1), 95–118.

Ward, M. D. and P. D. Hoff (2007). Persistent patterns of international commerce. *Journal of Peace Research 44*(2), 157–175.

Wasserman, S. and K. Faust (1994). *Social Network Analysis: Methods and Applications*, Volume 8. Cambridge University Press.

Wasserman, S. and P. Pattison (1996). Logit models and logistic regressions for social networks: An introduction to markov graphs. *Psychometrika 61*(3), 401–425.

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.

Wertz, R. T., L. L. LaPointe, and J. C. Rosenbek (1984). *Apraxia of speech in adults the disorder and its management*. Grune and Stratton.

Wheeler, D. C. and C. A. Calder (2007). An assessment of coefficient accuracy in linear regression models with spatially varying coefficients. *Journal of Geographical Systems 9*(2), 145–166.

Wigren, A., L. Murray, and F. Lindsten (2018). Improving the particle filter in high dimensions using conjugate artificial process noise. *IFAC-PapersOnLine 51*(15), 670–675.

Wilson, S. M., T. H. Brandt, M. L. Henry, M. Babiak, J. M. Ogar, C. Salli, L. Wilson, K. Peralta, B. L. Miller, and M. L. Gorno-Tempini (2014). Inflectional morphology in primary progressive aphasia: an elicited production study. *Brain Lang 136*, 58–68.

Wilson, S. M., A. T. DeMarco, M. L. Henry, B. Gesierich, M. Babiak, M. L. Mandelli, B. L. Miller, and M. L. Gorno-Tempini (2014). What role does the anterior temporal lobe play in sentence-level processing? neural correlates of syntactic processing in semantic variant primary progressive aphasia. *J Cogn Neurosci 26*(5), 970–985.

Woolrich, M. W., T. E. Behrens, and S. M. Smith (2004). Constrained linear basis sets for HRF modelling using Variational Bayes. *NeuroImage 21*(4), 1748–1761.

Xing, E. P., W. Fu, L. Song, et al. (2010). A state-space mixed membership blockmodel for dynamic network tomography. *The Annals of Applied Statistics 4*(2), 535–566.

Xu, L., T. D. Johnson, T. E. Nichols, and D. E. Nee (2009). Modeling inter-subject variability in fmri activation location: a Bayesian hierarchical spatial model. *Biometrics 65*(4), 1041–1051.

Xue, W., F. D. Bowman, and J. Kang (2018). A bayesian spatial model to predict disease status using imaging data from various modalities. *Frontiers in neuroscience 12*, 184.

Xue, W., F. D. Bowman, A. V. Pileggi, and A. R. Mayer (2015). A multimodal approach for determining brain networks by jointly modeling functional and structural connectivity. *Frontiers in computational neuroscience 9*, 22.

Yang, Y. and D. B. Dunson (2013). Sequential Markov Chain Monte Carlo. arXiv: 1308.3861.

Young, S. J. and E. R. Scheinerman (2007). Random dot product graph models for social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pp. 138–149. Springer.

Yu, R. and Y. Liu (2016). Learning from multiway data: Simple and efficient tensor regression. In *International Conference on Machine Learning*, pp. 373–381.

Yu, Z., R. Prado, E. B. Quinlan, S. C. Cramer, and H. Ombao (2016). Understanding the impact of stroke on brain motor function: a hierarchical Bayesian approach. *Journal of the American Statistical Association 111*(514), 549–563.

Zellner, A. (1962). An efficient method of estimating seemingly unrelated regressions and tests for aggregation bias. *Journal of the American statistical Association 57*(298), 348–368.

Zhang, C.-H. (2010). Nearly unbiased variable selection under minimax concave penalty. *The Annals of statistics 38*(2), 894–942.

Zhang, C.-H. and S. S. Zhang (2014). Confidence intervals for low dimensional parameters in high dimensional linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76*(1), 217–242.

Zhang, L., M. Guindani, and M. Vannucci (2015). Bayesian models for functional magnetic resonance imaging data analysis. *Wiley Interdisciplinary Reviews: Computational Statistics 7*(1), 21–41.

Zhang, Z., G. Dai, and M. I. Jordan (2010). Matrix-variate Dirichlet process mixture models. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pp. 980–987.

Zhang, Z., M. Descoteaux, and D. B. Dunson (2019). Nonparametric Bayes models of fiber curves connecting brain regions. *Journal of the American Statistical Association 114*(528), 1505–1517.

Zhao, Q., G. Zhou, L. Zhang, and A. Cichocki (2014). Tensor-variate Gaussian processes regression and its application to video surveillance. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 1265–1269. IEEE.

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

Zhou, H. and L. Li (2014). Regularized matrix regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 76*(2), 463–483.

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.

Zhou, J., E. D. Gennatas, J. H. Kramer, B. L. Miller, and W. W. Seeley (2012). Predicting regional neurodegeneration from the healthy brain functional connectome. *Neuron 73*(6), 1216–1227.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American statistical association 101*(476), 1418–1429.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the royal statistical society: series B (statistical methodology) 67*(2), 301–320.