UNIVERSITY OF CALIFORNIA
SANTA CRUZ

ON BAYESIAN METHODS IN NETWORK REGRESSION

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

DOCTOR OF PHILOSOPHY

in

STATISTICAL SCIENCE

by

Sharmistha Guha

December 2019

The Dissertation of Sharmistha Guha
is approved:

Professor Abel Rodriguez, Chair

Professor Athanasios Kottas

Professor Herbert Lee

Quentin Williams
Acting Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

List of Tables

Abstract

On Bayesian Methods in Network Regression

by

Sharmistha Guha

There has been a growing interest during recent years in connectomics, which is the study of interconnections or networks within the human brain. This interest has been spurred by the development of new imaging technologies, which allow researchers to peer non-invasively into the human brain and obtain data on connections. Motivated by these datasets, this dissertation develops a novel class of Bayesian regression models which study the relationships between neuro-scientific phenotypes and brain connectome networks of individuals.

First, we introduce a novel approach that develops a regression framework of the brain network (represented in the form of a symmetric matrix) on a continuous phenotypic response. We propose a novel network shrinkage prior on the network predictor coefficient matrix. The proposed framework is able to identify nodes or functional regions in the brain network and interconnections between different regions, significantly related to the phenotypic response. To the best of our knowledge, our framework is the first principled Bayesian framework that enables identification of network nodes and edges significantly related to the response. The performance of the proposed model is evaluated with respect to a wide range of existing competitors available in the high dimensional frequentist and Bayesian literature using a variety of simulation studies. The proposed model identifies important brain regions and interconnections significantly associated with creativity for a group of subjects.

Next, we extend our model to build network classifiers when a brain connectome network along with a binary response is provided for a group of individuals. Here we develop a broader class of global-local network shrinkage priors which includes the novel prior distribution specified earlier as a special case. We specifically consider two different global-local network shrinkage priors from this class of priors and investigate them using simulation studies. In particular, we assess their performance in terms of network classification and identifying influential network nodes and edges for the purpose of classification. We also demonstrate superior performance of our proposed network classifiers over state-of-the-art high dimensional classification techniques. Another major contribution remains developing theoretical conditions to guarantee asymptotically consistent classification for the proposed framework. In particular, we derive conditions on the number of network nodes, sparsity in the network coefficient matrix as a function of the sample size to achieve asymptotically optimal classification. While theoretical results on high dimensional binary regression with ordinary shrinkage priors have emerged recently, developing theory for our network classifier model involves several additional challenges due to the complex nature of the global local shrinkage prior developed here. The framework is used to classify individuals into high and low IQ groups based on their brain connectomes.

Notably, the work discussed in the last two paragraphs tacitly assumes that all nodes and edges have similar impact on a phenotype for every individual. In our next project, we study a brain connectome data where this assumption is violated. In fact, there is a relatively less developed literature in neuroscience that argues for different groups of individuals having shared relationships between brain networks and phenotypes, though this literature lacks a principled

Bayesian approach that takes into account different relationships of nodes and edges with the response for different groups of individuals and facilitates clustering of individuals. Motivated by this problem and our dataset, we have developed a Bayesian network mixture regression model. Simulation studies and analysis of the brain connectome dataset demonstrate superior performance of the proposed approach over the approach described earlier. Simulation studies are also used to evaluate the performance of the proposed approach by varying the true and fitted number of clusters, size of the network and sample size.

For these projects, computationally efficient Bayesian sampling algorithms are developed to enable computations even for reasonably large networks in presence of moderately large sample size.

To Shruti, the joy of my life; to my parents Subrata and Shorashi, for their unceasing love and care; and to Rajarshi, my friend, philosopher and guide, without whose constant love and unwavering support, all this would not have been possible.

Chapter 1

Introduction

1.1 Terminology and Network Properties

Interconnections among independent (or otherwise) components of a system can yield valuable information and may be of scientific interest in several scenarios. The intercommunication between these components (or actors) along with the structure formed by them is generally known as a *network* or a *graph*. One may find several applications of networks in fields such as the bio-sciences (eg. genetic interactions, protein networks), epidemiology (transmission of infectious diseases), the social sciences (social relationships and interactions), political science (international relations), finance (interactions between multinational corporations, economic interactions between various economies) and engineering (communication networks, networks across the internet) to name a few.

Network data is challenging to analyze, not only because it requires dimensionality reduction procedures to effectively deal with the large number of pairwise relationships, but also because flexible formulations are needed to account for the topological structure of the network. In addition to creating models that can efficiently explain the network structure, it is also of scientific interest to make predictions about missing and/or future relationships between network nodes and edges. An advantage of creating effective statistical models to explain and make predictions regarding networks is that they come with measures of uncertainty around the estimates and predictions.

The simplest form of network is a *binary network* in which the edges simply denote connection or lack of the same amongst any pair of nodes, thus being dichotomous in nature. Examples of this type of network could include ones providing information on whether a pair of actors are friends or not, or whether they are involved in a conflict or not, and so on. A network might also be one in which the edges are *weighted*. The weights may denote counts, e.g., distance or the number of transactions of a specific kind between a pair of nodes. Such a network is commonly known as a *valued* or a *weighted* network.

Networks may also be classified as *directed* or *undirected*. A directed (or asymmetric) relationship between a pair of actors would consist of two values, each value representing the stance of one actor towards the other. On the other hand, an undirected (or symmetric) relationship would consist only of a single value representing the stance of each pair of members. A simple example of an undirected network would be a brain imaging network where the relationship between a pair of regions of interest in the brain is captured by a single value. On the other hand, an example of a directed network could be a social influence network in which there is an influencer whose opinions or actions influence several followers but not the other way round.

Network data can usually be encoded using a so-called *adjacency matrix*. For a network with V nodes, the adjacency matrix is a $V \times V$ matrix, with the cell entries being dichotomous or continuous depending on whether the network is binary or weighted, respectively. The matrix would be symmetric or asymmetric depending upon the nature of the relationships between pairs of nodes, i.e. whether they are undirected or directed. Also, if there are no self-relationships, diagonal elements are not modeled. Notationally, $A = ((a_{k,l}))_{k,l=1}^{V}$ will be used to denote the $V \times V$ adjacency matrix corresponding to a network, where $a_{k,l}$ corresponds to the weighted or unweighted relationship between nodes k and l. Again, a network is often associated with edge specific covariates. Let $X = [x_{k,l}]$ be a covariate array of predictor variables $x_{k,l}$ corresponding to dyad (k,l). Sometimes covariates are available corresponding to every node, referred to as *node specific attributes*. Mathematically, we denote the attribute vector corresponding to the kth node by h_k.

There are various approaches in the literature in order to visualize and characterize networks, several of them being graph-theoretic in nature. Of course, the most appropriate way to visualize a network in a given context depends on the scientific question at hand. A review of network properties and measure summaries can be found in [140]; [104] and [103].

There are certain measures which are often used in the literature to summarize a network. A very important measure in the characterization of a network is the *degree* of its nodes. The degree of a node is the number of edges connected to that node. This is a measure of the extent of "connectedness" of each of the nodes in a network. Another measure is the *vertex centrality* which gauges the relative importance of a node in a network and is usually based on the *geodesic distance* or shortest distance between two nodes [140]; [86]. The connectivity of nodes within a network is represented by the *cohesion* of the system. Connection between nodes of a network based on their corresponding attributes is known as *homophily* or *assortative mixing* and is often encountered in social networks. Acute cases of homophily in which the network exhibits strong *community structure*, or in other words, a situation in which subsets of nodes or actors display cohesive patterns as a result of the underlying relational framework, also constitute an active field of research.

1.2 Statistical models for networks

Some of the pioneering work in the statistical modeling of networks dates back to the late 1950s and early 1960s. Prevailing literature in this field deals mainly with single network observations, with or without accompanying information on nodal attributes. By and large, the relationship between network and nodal attributes has been studied using two separate approaches. One of these approaches focuses on modeling the structure of the network conditional of the nodal attributes. The goal in this case is to understand how social relationships are formed based on attributes of individuals, a process known as "selection". The other approach consists of models of the nodal attributes and their association conditional on the network structure. These models are employed to understand how relationships affect attributes of the individuals in a network, a process referred to as "influence" or "contagion." Additional scenarios include the one in which the network and nodal attributes are jointly modeled. Another scenario of interest is when a response (continuous, binary or categorical) is regressed on a network, leading to a network regression problem, which is extensively studied in this proposal. We proceed to discuss each scenario in more detail below.

1.2.1 Models for Selection

Some of the pioneering work in the statistical modeling of networks dates back to the late 1950s and early 1960s. Prevailing literature in this field deals mainly with single network observations, with or without accompanying information on nodal attributes. More specifically, in most of the existing literature, a single network is subjected to an unsupervised analysis using random graph models [39]; [56], exponential random graph models [49], social space models [72]; [67], stochastic block models [106], bilinear mixed models [67] or eigenmodels [68]. We offer brief descriptions on these classes of models below.

The *random graph model* [39]; [56] is one of the foremost network models in the literature and is constructed in such a way that the edge between any pair of nodes is incorporated into the graph independently and with a fixed probability. In most real-world scenarios, the distribution of the *degree* of a network turns out to be positively skewed, since only a few nodes are expected to be very highly connected. This is a drawback for the random graph models since they imply a lighter tailed distribution of the degree. They are also more inclined to be dense, have small diameter and low clustering, which make them unrealistic for practical purposes.

More realistic situations in network data are accommodated by the *exponentially parameterized random graph models* (ERGM), also known as the p^* models [49]; [141]. ERGMs are expressed in exponential form and usually involve some summary statistics of the network. Specifically, the probability mass function for an ERGM is given by

$$p(A\,|\,X,\theta) = \frac{\exp\left\{\sum_{k=1}^{K}\theta_k S_k(A,X)\right\}}{\kappa(\theta)}$$

where each $S_k(A,X)$ is a network statistic, $\theta = (\theta_1,\ldots,\theta_K)^T$ is a K-dimensional unknown parameter vector and $\kappa(\theta)$ is a normalizing constant. Recall that examples of network statistics include degree, vertex centrality, cohesion and homophily, as described in section 1.1. ERGMs, though having some desirable features, have some shortcomings. They can be computationally challenging and can have the issue of model degeneracy (i.e. putting inordinate importance to a few network configurations). A detailed treatment of ERGMs can be found in [115] and [96].

A broad class of network models can be included under the umbrella of *social space models*. In the realm of *social space models*, the use of *random effects* in the context of probit or logistic regression to model binary networks has also become popular in recent times. Consider a probit model (the logistic model is analogous and has been used by [72] and [67] in which the $a_{k,l}$'s are conditionally independent with probability of interaction

$$\theta_{k,l} \equiv p(a_{k,l} = 1 \,|\, \beta, \gamma_{k,l}, x_{k,l}) = \Phi(x_{k,l}^T \beta + \gamma_{k,l});\;\; k,l = 1,...,V;\, k < l$$

where Φ denotes the cumulative distribution function of a standard normal random variable, β is an unknown vector of fixed effects and $\gamma_{k,l}$ is an unobserved dyad (k,l)-specific random effect unrelated to the predictor variable.

If the matrix of random effects $\Gamma = [\gamma_{k,l}]$ is jointly exchangeable, there exists a symmetric function $\alpha(\cdot,\cdot)$ such that $\gamma_{k,l} = \alpha(u_k, u_l)$ where u_k, $k \in \{1,...,V\}$ [4]. The form of the function $\alpha(\cdot,\cdot)$ is directly associated with the important structural characteristics of the network. There have been a number of alternatives to select the latent factors which give rise to different classes of social space models. For example, **stochastic block models** [106] assume that each node k is associated with an unobserved latent class and there is a probability distribution characterizing the relationship between each pair of nodes. Here the latent effects are specified as $\alpha(u_k, u_l) = m_{u_k,u_l}$, where $u_k, u_l \in \{1,2,3,...,R\}$, R is the number of latent classes, and also $m_{r,s} \in \mathcal{R}$ and $m_{r,s} = m_{s,r}$. **Latent distance models** [72], on the other hand, assume that $\alpha(u_k, u_l) = -|u_k - u_l|$, where $|\cdot|$ denotes the euclidean norm. The underlying assumption here is that the probability of an edge between two nodes increases as the latent characteristics of these nodes come closer in terms of their euclidean distance. **Bilinear models** [67] assume that the probability of an edge between two nodes is a symmetric multiplicative effect. The multiplicative interaction for a dyad (k,l) is expressed in terms of a *bilinear effect*, i.e. the inner product of the unobserved latent vectors u_k and u_l. Hence, the latent effects are specified as $\alpha(u_k, u_l) = u_k^T u_l$, where $u_k^T u_l$ is the bilinear effect. The rationale behind this type of models is that the probability of an edge between two nodes increases as the angle formed by the corresponding latent positions becomes wider, i.e. nodes k and l would be prone to having a tie if the angle between them is acute ($u_k^T u_l > 0$), neutral to a tie if the angle is a right angle ($u_k^T u_l = 0$) and averse to having a tie if the angle between them is obtuse ($u_k^T u_l < 0$). Bilinear models can generalize distance models, but not latent class models, since the eigenvalues of latent class models may be negative [68]. **Eigenmodels** [68] are a generalization of the latent class and latent distance models due to the fact that they can be used to represent the same network features but not the other way round. These models are based on the principles of eigen-analysis and render the relationship between two nodes as the inner-product of node-specific latent vectors, i.e. $\alpha(u_k, u_l) = u_k^T \Lambda u_l$, where Λ is a $R \times R$ diagonal matrix.

1.2.2 Models of Contagion

Models of contagion are usually constructed by regressing a nodal attribute on the attributes of other nodes in the social network (e.g., see [24]; [47]; [124] and references therein), with common methodological approaches including simultaneous autoregressive (SAR) models [94] and threshold models [142]. For instance, node specific responses $\{y_k : k \in \{1, \ldots, V\}\}$ are regressed on the node specific attributes using the simultaneous autoregressive models (SAR) that respect the network structure.

1.2.3 Joint Modeling of Network and Attributes

It is usually a complicated problem to ascertain the direction of a causal relationship between network structure and link or nodal attributes, i.e. whether it pertains to selection or contagion [33]. Hence, a section of the literature focusses on jointly modeling the co-evolution of network and nodal attributes through shared latent variables. In recent years, joint models of network and attributes have been receiving increased attention. [46] have recently proposed an extension of the bilinear model of [67] in a static setting where the nodal attributes and latent factors used to describe *transitivity* (the extent to which the relation between two nodes in a network that are connected by an edge is transitive) in the network are jointly modeled using a multivariate normal distribution. [36], on the other hand, propose joint modeling of a binary/categorical response and a network using latent variable tensor factorization of the joint probability model. [30] have proposed time varying joint models for network and attributes when the attributes are binary or categorical in a dynamic setting, while [105] extend the framework to accommodate continuous nodal attributes. [61] propose a Bayesian approach to inference, testing and prediction for co-evolving networks and nodal attributes by accommodating both discrete and continuous attributes and considering the more general case of time series data. They use a common set of latent factors to explain network transitivity and covariation among attributes and network structure, and provide a fully Bayesian test of association in order to study individual nodal attributes. When the nodal attributes are assumed to follow conditional Gaussian distribution, their model can be interpreted as a dynamic version of the model presented in [46], with a structured and more parsimonious prior on the covariance matrix between the latent traits and the nodal attributes.

1.2.4 Models for Network Regression

Previous models focus on the analysis of a single network. There are situations in which a network is collected for each observational unit. This is especially pertinent to biological and physiological problems wherein, for example, each node corresponds to a certain fixed location in the human brain or a particular genetic unit in a gene network. Furthermore, the data might contain a continuous or categorical outcome corresponding to each individual in the sample, possibly associated with the network. Examples of such datasets include brain connectome applications for multiple individuals which we discuss in detail in Chapters 2, 3 and 4. The nodes in the network correspond to the brain regions of interest (ROI) shared by all individuals in the sample and are registered by mapping every brain to a common brain atlas. Additionally, data on a phenotype is available for every individual. For example, the phenotype can be continuous such as a measure of creativity for each individual called the *Composite Cre-*

ativity Index (CCI). Sometimes the outcome can be binary representing whether a subject has 'high' or 'low' IQ.

In relating the response to the undirected network, a common approach would be to vectorize the network predictor (originally obtained in the form of a symmetric matrix) and treat it as a collection of a large number of edge weights [114]; [27]. Subsequently, the response would be regressed on the high dimensional collection of edge weights. This idea can take advantage of the recent developments in high dimensional regression, consisting of both penalized optimization [133] and Bayesian shrinkage [109],[17],[5] perspectives. Additionally, these models are computationally convenient and are generally accompanied by theoretical guarantee. While the predictive performance of these methods turns out to be satisfactory, their interpretability is limited to individual edge selection, which is scientifically less interesting than identifying nodes impacting the response. Furthermore, they ignore the network structure, i.e. the relevant wiring mechanism in the brain architecture for brain connectome analysis, which may contain a plethora of scientific information.

While there are existing approaches for network classification, most of them fail to incorporate the full network information in the process of classification and rather use a few summary measures from the network, for e.g. see [11] and references therein. [113] have recently proposed a penalized optimization scheme that not only enables classification of networks, but also identifies important nodes and edges. Although their framework is demonstrated for classification purposes, it can be adapted to facilitate regression settings (as described in Chapter 2). One key shortcoming of this approach is that it is unable to provide any measure of predictive uncertainty. The need for valid measures of uncertainty on parameter (predictive) estimates is crucial, especially in settings with low or moderate sample sizes with complex predictor dependence, which naturally motivates our Bayesian approach.

There are recent Bayesian approaches which propose joint modeling of response and predictors, see e.g. [36]. However, these methods are somewhat restrictive for multiple reasons. First, their approach is heavily dependent on the assumption that the network is binary and does not find easy extension when the network is weighted. Secondly, their modeling perspective focuses on the classification of a population of networks into two groups and does not assume easy extension to regression settings. In a separate approach, [137] regress a network response on a scalar predictor, which is a different problem from the one we are interested in.

1.3 Thesis Outline

In Chapter 2 we develop a novel framework to answer some important questions arising from datasets of these types. Primarily, in Chapter 2, our inferential focus lies in developing a high-dimensional regression model of a continuous response on the network predictors that employs all edge weights, but aims at identifying influential nodes and edges to yield scientifically meaningful results. To this end, we construct a novel Bayesian *network shrinkage prior* that incorporates network information in the coefficients corresponding to the network predictors through a social space model [72] with latent variables embedded within a Bayesian shrinkage prior [109], [17], [5]. We index these latent variables by nodes in the network and incorporate a spike-and-slab variable selection prior to choose the relevant node specific latent variables explaining variation in the response. The proposed framework is simple enough to allow computation through a data-augmented Gibbs sampler. We make the practical benefit of the proposed approach in terms of inference and prediction amply clear by comparing it to other existing methods in various simulation studies. Further, we provide detection of influential brain regions and influential interconnections between different regions responsible for creativity of individuals in a principled Bayesian way which was hitherto not present in the literature. The model provides additional inferences which can be found in Chapter 2.

Chapter 3 focuses on a network classification problem where a binary response along with a network is available from every subject. The aim lies in developing a classification of subjects, along with identifying network nodes and edges influential for the purpose of classification. We broadens the formulation of Bayesian network shrinkage prior developed in Chapter 2 and propose a new class of Bayesian network global-local shrinkage prior that includes the network shrinkage prior formulated in Chapter 2 as a special case. Simulation studies show superior performance of the proposed formulation over the existing network classification models. We employ the framework to analyze a dataset that aims at classifying subjects into a 'low' or 'high' IQ group based on her/his brain connectome network. One important contribution of Chapter 3 remains theoretical study of asymptotic properties of the posterior distribution for binary network regression model. In particular, we offer theoretical conditions to ensure asymptotically optimal classification from the binary network regression model proposed in Chapter 3. The proofs of the theoretical results in Chapter 3 can be easily adapted to show the consistency of the posterior distribution for the model proposed in Chapter 2.

Chapter 4 presents a brain connectome dataset with a phenotype and brain connectome network corresponding to multiple subjects. Analysis of the data with the Bayesian network regression model proposed in Chapter 2 seems inadequate. Indeed, there is a literature in neuroscience arguing differential relationships between brain networks and human creativity for different groups of individuals. In particular, they argue that the relationship may be very different from people with high IQ compared to people with low IQ. To address this issue, Chapter 4 proposes a Bayesian network mixture regression model, allowing for different network regression models for different groups of subjects. Finally, Chapter 7 presents appendices with details of model implementations and proofs of theorems.

Chapter 2

Bayesian Regression with Undirected Network Predictors with an Application to Brain Connectome Data

2.1 Introduction

In recent years, network data has become ubiquitous in disciplines as diverse as neuroscience, genetics, finance and economics. Nonetheless, statistical models that involve network data are particularly challenging, not only because they require dimensionality reduction procedures to effectively deal with the large number of pairwise relationships, but also because flexible formulations are needed to account for the topological structure of the network.

The literature has paid heavy attention to models that aim to understand the relationship between node-level covariates and the structure of the network. A number of classic models treat the dyadic observations as the response variable, examples include exponential random graph models [49], social space models [72, 67, 68] and stochastic block models [106]. The goal of these models is often either to predict unobserved links or to investigate *homophily*, i.e., the process of formation of social ties due to matching individual traits. Alternatively, models that investigate *influence* or *contagion* attempt to explain the node-specific covariates as a function of the network structure (e.g., see [24]; [47]; [124] and references therein). Common methodological approaches in this context include simultaneous autoregressive (SAR) models [94] and threshold models [142]. However, ascertaining the direction of a causal relationship between network structure and link or nodal attributes, i.e., whether homophily or contagion are in play, is difficult (e.g., see [33] and [120] and references therein). Hence, there has been a growing interest in joint models for the coevolution of the network structure and nodal attributes [46, 36, 30, 105, 61].

In this chapter we investigate Bayesian models for network regression. Unlike the problems discussed above, in network regression we are interested in the relationship between the structure of the network and one or more global attributes of the experimental unit on which the network data is collected. As a motivating example, we consider the problem of predicting the composite creativity index of individuals on the basis of neuroimaging data measuring the connectivity of different brain regions. The goal of these studies is twofold. First, neuroscientists are interested in identifying regions of the brain that are involved in creative thinking. Secondly, it is important to determine how the strength of connection among these influential regions affects the level of creativity of the individual. To address these challenges we construct a novel Bayesian *network shrinkage prior* that combines ideas from spectral decomposition methods and spike-and-slab priors to generate a model that takes into account the structure of the predictors. The model produces accurate predictions, allows us to identify both nodes and links that have influence on the response, and yields well-calibrated interval estimates for the model parameters.

A common approach to network regression is to use a few summary measures from the network in the context of a flexible regression or classification approach (e.g., see [11] and references therein). Clearly, the success of this approach is highly dependent on selecting the right summaries to include. Furthermore, this kind of approach cannot identify the impact of specific nodes on the response, which is of clear interest in our setting. Alternatively, a number of authors have proceeded to vectorize the network predictor (originally obtained in the form of a symmetric matrix). Subsequently, the continuous response would be regressed on the high dimensional collection of edge weights [114, 27]. This approach can take advantage of the recent developments in high dimensional regression, consisting of both penalized optimization [133] and Bayesian shrinkage [109, 17, 5]. However, this approach treats the links of the network as if they were fully exchangeable, ignoring the fact that coefficients that involve common nodes can be expected to be correlated a priori. Ignoring this correlation is known to lead to poor predictive performance and to potentially impact model selection.

Recently, [113] proposed a penalized optimization scheme that not only enables classification using network predictors, but also identifies important nodes and edges. Although this model seems to perform well for prediction problems, uncertainty quantification is difficult because standard bootstrap methods are not consistent for Lasso-type methods [89, 21].

Modifications of the bootstrap that produce well-calibrated confidence intervals in the context of standard Lasso regression have been proposed [22], but it is not clear whether they extend to the kind of group Lasso penalties discussed in [113]. Recent developments on tensor regression [147, 62] are also relevant to our work. However, these approaches tend to focus mainly on prediction and identification of important edges, but are not designed to detect important nodes impacting the response.

The rest of the chapter evolves as follows. Section 3.2 proposes the novel network shrinkage prior and discusses posterior computation for the proposed model. Empirical investigations with various simulation studies are presented in Section 2.3, while Section 2.4 analyzes the brain connectome dataset. We provide results on *region of interest* (ROI) and *edge* selection and find them to be scientifically consistent with previous studies. Finally, Section 2.5 concludes the chapter with an eye towards future work.

2.2 Model Formulation

2.2.1 Definitions and Notations

Let y_i and $A_i \in \mathbb{R}^{V \times V}$ represent the observed scalar response and the corresponding weighted undirected network for the i-th sample, $i = 1, \dots, n$, respectively. Depending on the problem y_i is continuous or binary. For example, $y_i \in \mathbb{R}$ is continuous in Chapters 2 and 4, and $y_i \in \{0, 1\}$ is binary in Chapter 3. All graphs share the same labels on their nodes. In all our applications, A_i encodes the strength of the network connections between different regions of the brain for the i-th individual. Mathematically, this amounts to A_i being a $V \times V$ matrix, with the (k,l)-th entry of A_i denoted by $a_{i,k,l} \in \mathbb{R}$. We focus on networks that contain no self relationship, i.e., $a_{i,k,k} \equiv 0$, and are undirected ($a_{i,k,l} = a_{i,l,k}$). The brain connectome application considered here naturally justifies these assumptions. Although we present our models specific to these settings, it will be evident that the proposed model can be easily extended to directed networks with self-relations. Throughout all chapters, we denote the Frobenius inner product between two $V \times V$ matrices A and B by $\langle A, B \rangle_F = Trace(B'A)$. Frobenius inner product is the natural inner product on the space of matrices and is a generalization of the dot product from vector to matrix spaces. Frobenius norm of a matrix A is defined as $||A||_F = \sqrt{\langle A, A \rangle_F}$. Additionally, for any vector $a = (a_1, ..., a_p)'$, define the L_1, L_2 and L_∞ norms by $||a||_1 = \sum_{l=1}^{p} |a_l|$, $||a||_2 = \sqrt{\sum_{l=1}^{p} a_l^2}$ and $||a||_\infty = \max_l |a_l|$ respectively. $|| \cdot ||_0$ denotes the L_0-norm, i.e. the number of non-zero entries for vectors. The $|| \cdot ||_1$, $|| \cdot ||_2$ and $|| \cdot ||_\infty$ norms of a matrix are defined analogously. All vectors and matrices are denoted by lowercase bold letters and uppercase bold letters respectively.

2.2.2 Bayesian Network Regression Model

We propose the high dimensional regression model of the response y_i for the i-th individual on the undirected network predictor $A_i = ((a_{i,k,l}))_{k,l=1}^{V}$ as

$$y_i = \mu + \langle A_i, B \rangle_F + \varepsilon_i, \; \varepsilon_i \overset{iid}{\sim} N(0, \tau^2), \tag{2.1}$$

where B is the symmetric network coefficient matrix of dimension $V \times V$ whose (k,l)-th element is given by $\gamma_{k,l}/2$ and $\langle A_i, B \rangle_F = Trace(B'A_i)$ denotes the Frobenius inner product between A_i and B. The Frobenius inner product is the natural inner product in the space of matrices and is a generalization of the dot product from vector to matrix spaces. Note that, similar to the network predictor, the network coefficient matrix B is assumed to be symmetric with zero diagonal entries. The parameter τ^2 is the variance of the observational error.

Since self relationship is absent and both A_i and B are symmetric, $\langle A_i, B\rangle_F = \sum_{1\leq k<l\leq V} a_{i,k,l}\gamma_{k,l}$, and (2.1) can be rewritten as

$$y_i = \mu + \sum_{1\leq k<l\leq V} a_{i,k,l}\gamma_{k,l} + \varepsilon_i, \qquad \varepsilon_i \sim N(0,\tau^2). \tag{2.2}$$

Equation (2.2) connects the network regression model with the linear regression framework with $a_{i,k,l}$'s as predictors and $\gamma_{k,l}$'s as the corresponding coefficients. However, while in ordinary linear regression the predictor coefficients are indexed by the natural numbers $\mathcal{N}$, Model (2.2) indexes the predictor coefficients by their positions in the matrix B. This is done in order to keep tabs not only on the edge itself but also on the nodes connecting the edges.

2.2.3 Developing the Network Shrinkage Prior

Vector Shrinkage Prior

High dimensional regression with vector predictors has recently been of interest in Bayesian statistics. Continuous shrinkage priors, which strongly shrink coefficients corresponding to unimportant variables to zero while minimizing the shrinkage of coefficients corresponding to influential variables, have become particularly popular. Many of these priors can be expressed as a scale mixture of normal distributions, commonly referred to as *global-local scale mixtures* [110], that enable fast computation employing simple conjugate Gibbs sampling. More precisely, in the context of model (2.2), a global-local scale mixture prior would take the form

$$\gamma_{k,l} \sim N(0, s_{k,l}\tau^2), \qquad s_{k,l} \sim g_1, \qquad \tau^2 \sim g_2, \qquad 1 \leq k < l \leq V.$$

Note that $s_{1,2}, \ldots, s_{V-1,V}$ are local scale parameters controlling the shrinkage of the coefficients, while τ^2 is the global scale parameter. Different choices of g_1 and g_2 lead to different classes of Bayesian shrinkage priors. For example, the Bayesian Lasso [109] prior takes g_1 as exponential and g_2 as the Jeffreys prior, the Horseshoe prior [17] takes both g_1 and g_2 as half-Cauchy distributions, and the Generalized Double Pareto Shrinkage prior [5] takes g_1 as exponential and g_2 as the Gamma distribution.

The direct application of this global-local prior in the context of (2.2) is unappealing. In practice, we expect the matrix of coefficients B (which itself can be regarded as describing a weighted network) to exhibit transitivity effects, i.e., we expect that if the interactions between regions i and j and between regions i and k both influence the response, the interaction between regions j and k will also be influential [93]. Ordinary global-local shrinkage priors do not necessarily conform to such an important restriction.

Network Shrinkage Prior

We propose a shrinkage prior on the coefficients $\gamma_{k,l}$ and refer to it as the *Bayesian Network Shrinkage prior* (BNSP). The prior borrows ideas from low-order spectral representations of matrices, and aims to capture transitivity effects in the matrix of regression coefficients. Let $u_1, \ldots, u_V \in \mathbb{R}^R$ be a collection of R-dimensional latent variables, one for each node, such that u_k corresponds to node k. We draw each $\gamma_{k,l}$ conditionally independent from a density that can be represented as a location and scale mixture of normals. More precisely,

$$\gamma_{k,l}|s_{k,l},u_k,u_l,\tau^2 \sim N(u_k'\Lambda u_l,\tau^2 s_{k,l}), \qquad s_{k,l} \sim Exp(\theta/2), \qquad \theta \sim Gamma(\zeta,\iota), \tag{2.3}$$

where $s_{k,l}$ is the scale parameter corresponding to each $\gamma_{k,l}$, and $\Lambda = \text{diag}(\lambda_1,\dots,\lambda_R)$ is an $R\times R$ diagonal matrix with $\lambda_r \in \{0,1\}$. Conditional on the latent variables u_k, u_l and Λ, if $s_{k,l} = 0$ then (4.5) implies a reduced rank-decomposition $\Gamma = 2B = U'\Lambda U$, where U is an $R\times V$ matrix whose k-th column corresponds to u_k and $\Gamma = ((\gamma_{k,l}))_{k,l=1}^{V}$. Drawing intuition from [67], we can interpret the latent vectors $u_1,\dots,u_V$ as the positions of the nodes in a latent "social" space, with the strength of the edge effect being controlled by the angular distance between the vectors. In this interpretation, $\sum_{r=1}^{R}\lambda_r = R_{eff} \leq R$, represents the *effective dimensionality* of the latent space. The effect of the interaction between the k-th and l-th nodes has a positive, negative or neutral impact on the response depending on whether the node specific latent variables u_k and u_l are in the same direction, opposite direction or orthogonal to each other respectively. In other words, whether the angle between u_k and u_l is acute, obtuse or right, i.e., $u_k'\Lambda u_l > 0$, $u_k'\Lambda u_l < 0$ or $u_k'\Lambda u_l = 0$ respectively. This kind of bilinear structure is commonly used to model social and biological networks because of its ability to capture the kind of transitive effects we discussed before [67, 66].

In order to learn which components of u_k are informative for (4.5), we assign a hierarchical prior

$$\lambda_r \sim Ber(\pi_r), \qquad \pi_r \sim Beta(1,r^{\eta}), \qquad \eta > 1.$$

The choice of hyper-parameters of the beta distribution is crucial. In particular, note that $E[\lambda_r] = 1/(1+r^{\eta}) \to 0$ as $r \to \infty$ and that $\sum_{r=1}^{R} var(\lambda_r) = \sum_{r=1}^{R} \frac{r^{\eta}}{(1+r^{\eta})^2(2+r^{\eta})} < \infty$ as $R \to \infty$.

The first property provides (weak) identifiability of the different latent dimensions, while the second ensures that $\lim_{R\to\infty} var(u_k) < \infty$ as long as the prior for the u_k's has a finite second moment. In fact, we can think of our model as a level-R truncation of an infinite dimensional model, similar in spirit to the stick-breaking construction of the Indian Buffet process [132]. Therefore, as long as R is chosen to be "large enough", the inferences will be roughly invariant to this choice. In our illustrations, we perform sensitivity analyses to determine an optimal value of R that maintains computational efficiency, and at the same time ensures the robustness of the results.

In order to determine which nodes are most influential in explaining the response, we assign a *spike-and-slab* mixture prior [76] to the latent factor u_k,

$$u_k \sim \begin{cases} N(0,M), & \text{if } \xi_k = 1, \\ \delta_0, & \text{if } \xi_k = 0, \end{cases} \qquad \xi_k \sim Ber(\Delta), \tag{2.4}$$

where δ_0 is the Dirac-delta function at 0 and M is a covariance matrix of order $R \times R$. The parameter Δ corresponds to the probability of the nonzero mixture component. Note that if the k-th node of the network predictor is not influential in predicting the response then, a-posteriori, ξ_k should provide high probability to 0. Thus, based on the posterior probability of ξ_k, it will be possible to identify unimportant nodes, which we loosely refer to as "uninfluential nodes", in the network regression.

The rest of the hierarchy is accomplished by assigning prior distributions on Δ and M as follows:

$$M \sim IW(\nu, \mathbf{I}), \qquad \Delta \sim Beta(a_\Delta, b_\Delta),$$

where $IW(\nu,\mathbf{I})$ denotes an Inverse-Wishart distribution with identity scale matrix $\mathbf{I}$ and degrees of freedom ν. Finally, we choose a non-informative prior on (μ,τ^2) such that $p(\mu,\tau^2) \propto \frac{1}{\tau^2}$. Appendix A shows the propriety of the posterior distribution under this prior.

The previous discussion assumes that we have conditioned on the latent positions $u_1,\ldots,u_V$ and the local scale parameters $(s_{k,l})$. Now that we have described the full hierarchical structure of the model, it is instructive to briefly discuss the structure of the marginal prior distribution obtained after integrating these latent variables. In this regard, note first that integrating over the $s_{k,l}$'s alone leads to double exponential priors that are reminiscent of the Lasso. On the other hand, while no closed form expression exists for the marginal prior after integrating $u_1,\ldots,u_V$, it is easy to see that, marginally, the edge coefficients have mean zero and are not independent. Hence, from this point of view, the latent positions $u_1,\ldots,u_V$ simply provide a mechanism to sparsely model the prior dependence among coefficients.

2.2.4 Posterior Computation

Although summaries of the posterior distribution cannot be computed in closed form, full conditional distributions for all the parameters are available and correspond, in most cases, to standard families. Thus, posterior computation can proceed through a Markov chain Monte Carlo algorithm. We note, however, that a naive implementation of such algorithm to update Γ would have complexity $O(q^3)$, where $q = V(V-1)/2$. The resulting algorithm would therefore be computationally too expensive for situations such as our real data application, where $V = 68$ and $q = 2278$. To address this issue, we follow [8], who propose the use of the Woodbury matrix identity to instead compute the inverse of an $n \times n$ matrix. Since in the type of applications with which this chapter is concerned n is typically much smaller than q, this approach leads to substantial computational savings that make real-life applications feasible. Details of all the Markov chain Monte Carlo algorithm are presented in Appendix B.

While inferences on the latent positions $u_1,\dots,u_V$ is not our main focus, being able to visualize these positions can be helpful in terms of interpreting the model results. However, note that vectors $u_1,\dots,u_V$ are not identifiable because the model is invariant to rotations of the latent space. Hence, before we can use the posterior samples generated by our algorithm to conduct inferences on these latent positions we must first rotate them to a common orientation. This is done using a "Procrustean" transformation [125, 72, 67]. For each posterior sample $U^{(\ell)}$ we find the rotation $\tilde{U}^{(\ell)}$ that has the smallest sum of squared deviations from an arbitrary fixed reference matrix U_0. This rotation is given by $\tilde{U}^{(\ell)} = U_0\left(U^{(\ell)}\right)'\left\{U^{(\ell)}U_0'U_0\left(U^{(\ell)}\right)'\right\}^{-1/2}U^{(\ell)}$. In our analysis, we use the first iterate after burn-in, $U^{(1)}$, as the reference matrix U_0.

In order to identify whether the k-th node is important in terms of predicting the response, we rely on the post burn-in L samples $\xi_k^{(1)},\dots,\xi_k^{(L)}$ of ξ_k. Node k is said to be influential if $\frac{1}{L}\sum_{l=1}^{L}\xi_k^{(l)} > 0.5$. To identify influential edges we utilize a modification of the algorithm proposed in [92] that allows us to estimate the false discovery rate of the procedure as a function of the number of discoveries. Details are provided in Appendix C. Finally, an estimate of $P(R_{eff} = r\,|\,Data)$ is given by $\frac{1}{L}\sum_{l=1}^{L} I(\sum_{m=1}^{R}\lambda_m^{(l)} = r)$, where $I(A)$ for an event A is 1 if the event A happens and 0 otherwise, and $\lambda_m^{(1)},\dots,\lambda_m^{(L)}$ are the L post burn-in MCMC samples of λ_m.

2.3 Simulation Studies

This section comprehensively contrasts both the inferential and predictive performances of our proposed approach with a number of competitors in various simulation settings. As competitors, we consider both penalized likelihood methods as well as Bayesian shrinkage priors for high-dimensional regression.

Our first set of competitors use generic variable selection and shrinkage methods that treat edges between nodes as “bags of predictors” and rely on high dimensional regression, thereby ignoring the relational nature of the predictor. More specifically, we use Lasso [133], which is a popular penalized optimization scheme, and the Bayesian Lasso [109] and Horseshoe priors [17], which are popular Bayesian shrinkage regression methods. The Horseshoe in particular is considered to be a state-of-the-art Bayesian shrinkage prior and is known to perform well, both in sparse and not-so-sparse regression settings. We use the `glmnet` package in `R` [50] to implement Lasso regression, and the `monomvn` package in `R` [59] to implement the Bayesian Lasso (BLasso for short) and the Horseshoe (BHS for short).

A thorough comparison with these methods will indicate the relative advantage of exploiting the structure of the network predictor.

Additionally, we compare our method to a frequentist approach that develops network regression in the presence of a network predictor and scalar response [113]. To be precise, we adapt [113] to a *continuous response* context and propose to estimate the network regression coefficient matrix B by solving

$$\hat{B} = \arg \min_{B \in \mathbb{R}, B=B', diag(B)=0} \left\{ \frac{1}{n} \sum_{i=1}^{n} (y_i - \mu - \langle A_i, B \rangle_F)^2 + \right.$$
$$\left. \frac{\varphi}{2} ||B||_F^2 + \varsigma \left(\sum_{k=1}^{V} ||B_{(k)}||_2 + \rho ||B||_1 \right) \right\}, \tag{2.5}$$

where $||B||_F = \sqrt{\langle B, B \rangle_F}$ denotes the Frobenius norm, $||B||_1$ is the sum of the absolute values of all the elements of matrix B, $|| \cdot ||_2$ is the l_2 norm of a vector, $B_{(k)}$ is the k-th row of B and φ, ρ, ς are tuning parameters. The best possible choice of the tuning parameter triplet $(\varphi, \rho, \varsigma)$ is made using cross validation over a grid of possible values. [113] argue that the penalty in (2.5) incorporates the network information of the predictor, thereby yielding superior inference to any ordinary penalized optimization scheme. Hence comparison with (2.5) will highlight the advantages of a carefully structured Bayesian network shrinkage prior over the penalized optimization scheme incorporating network information. In the absence of open source code, we implemented the algorithm in [113] ourselves. All Bayesian competitors are allowed to draw $50,000$ MCMC samples, out of which the first $30,000$ are discarded as burn-ins. All posterior inference is carried out based on the rest $20,000$ MCMC samples after suitably thinning the post burn-in chain. Convergence is assessed by comparing different simulated sequences of representative parameters started at different initial values [52].

2.3.1 Predictor and Response Data Generation

In all simulation studies, the response y_i is generated according to the network regression model

$$y_i = \mu_0 + \langle A_i, B_0 \rangle_F + \varepsilon_i, \qquad \varepsilon_i \sim N(0, \tau_0^2), \tag{2.6}$$

with τ_0^2 as the true noise variance. In all of our simulations, we use $V = 20$ nodes and $n = 70$ samples.

Simulation 1

In this group of simulations, the (k,l)-th entry of B_0 is given by $\frac{w_k' w_l}{2}$, where the vectors $w_1, \ldots, w_V$, each of dimension R_{gen}, are generated from a mixture

$$w_k \sim \pi_w N_{R_{gen}}(w_{mean}, w_{sd}^2) + (1-\pi_w)\delta_0, \qquad k \in \{1, \ldots, V\}, \tag{2.7}$$

where δ_0 is the Dirac-delta function and π_w is the probability of any w_k being nonzero. $(1-\pi_w)$ is the probability of a node not being influential, it is referred to as the *node sparsity parameter*. This data generation mechanism is quite similar (although not identical) to our hierarchical prior. Hence, the goal of this first simulation is to evaluate the ability of the model to recover the true data-generation mechanism and, in particular, its ability to identify the true dimension of the latent space, as well as the sensitivity of the results to the choice of the maximum latent dimension R.

For a comprehensive picture of *Simulation 1*, we consider 11 different cases as summarized in Table 3.1. In each of these cases, the network predictor coefficient and the response are generated by changing the sparsity π_w and the true dimension R_{gen} of the latent variables w_k's. The table also presents the maximum dimension R used to fit the model of the latent variables u_k for the network regression model (2.2). Note that we include various cases of model mis-specification in which $R > R_{gen}$. For all simulations, w_{mean} and w_{sd}^2 are set as $0.5 \times \mathbf{1}_{Rgen}$ and $I_{R_{gen} \times R_{gen}}$, respectively, and the variance τ_0^2 is fixed at 1. In Cases 1-9, the entries of the network predictor A_i for the i-th sample are simulated from a standard normal distribution. In Cases 10 and 11 the network predictor A_i for the i-th sample follows a stochastic blockmodel. In Case 10, we assume that each brain network has 3 local clusters with high within-cluster and low between-cluster connectivity. More specifically, the matrices A_i's consist of 3 symmetric block diagonal matrices of dimensions 6×6, 7×7 and 7×7 respectively. Elements in these matrices have been drawn from $N(j, j^2)$ where $j \in \{1, 2, 3\}$, for the j-th block diagonal. The off-diagonal blocks are highly sparse with very few randomly chosen non-sparse elements denoting connections between nodes in different clusters randomly chosen from $N(0, 1)$. In Case 11, the adjacency matrices A_i's also consists of 3 block diagonal matrices, in this case of dimensions 5×5, 8×8 and 7×7. As before, the elements in these matrices have been drawn from $N(j, j^2)$ 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 $N(4, 1)$, $N(5, 1)$ and $N(6, 1)$.

Simulation 2

In this case, the matrix of coefficients B_0 is constructed by first generating V binary indicators $\xi_1^0, \ldots, \xi_V^0$ independently from a $Ber(\pi_{2,w})$, one for each node in the network. If both $\xi_k^0 = 1$ and $\xi_l^0 = 1$, the edge coefficient connecting the k-th and the l-th nodes ($k < l$) is simulated from $N(0.8, 1)$. Otherwise, we set the (k, l)-th edge coefficient to be 0. Similar to *Simulation 1*, we refer to $1 - \pi_{2,w}$ as the *node sparsity parameter*. While this simulation scenario has some similarities to our proposed model, the mean effect for active nodes is constant. Therefore, the goal of this simulation is to evaluate the performance of the model in situations where there are weak network effects in the matrix of coefficients. The network predictor A_i for the i-th sample is simulated by drawing $a_{i,k,l}$ independently from a $N(0,1)$ distribution for $k < l$ and setting $a_{i,k,l} = a_{i,l,k}$ and $a_{i,k,k} = 0$ for all $k,l \in \{1,\ldots,V\}$. Finally, the variance τ_0^2 is fixed at 1 as in *Simulation 1*. Table 2.2 presents the two cases we consider for *Simulation 2*, which are obtained by varying the node sparsity parameter.

Simulation 3

In this case, we draw V indicator variables $\xi_1^0,\ldots,\xi_V^0$ from a $Ber(\pi_{2,w})$ corresponding to the V nodes of the network. If both $\xi_k^0 = 1$ and $\xi_l^0 = 1$, then the edge coefficient connecting the k-th and the l-th nodes ($k < l$) is simulated from a mixture distribution given by

$$\pi_{3,w} \sim N_{R_{gen}}(0.8,1) + (1-\pi_{3,w})\delta_0, \qquad\qquad k,l \in \{1,\ldots,V\}. \tag{2.8}$$

Otherwise, if $\xi_k^0 = 0$ for any k, we set (k,l)-th edge coefficient to be 0 for all l. Contrary to *Simulation 2*, *Simulation 3* allows the possibility of an edge between the k-th and the l-th nodes having no impact on the response even when both ξ_k^0 and ξ_l^0 are nonzero. In the context of *Simulation 3*, $(1-\pi_{2,w})$ and $(1-\pi_{3,w})$ are referred to as the *node sparsity* and the *edge sparsity* parameters, respectively. Hence, the goal of this simulation is to evaluate the impact of edge sparsity and its interaction with node sparsity on model performance. Network predictors are randomly generated using the same mechanism as in *Simulation 2* and the true variance τ_0^2 is again fixed at 1 for all cases. Table 2.3 presents the four cases we consider in this evaluation, which are generated by changing the *node sparsity* and *edge sparsity*.

Cases	R_{gen}	R	Sparsity
Case - 1	2	2	0.5
Case - 2	2	3	0.6
Case - 3	2	5	0.3
Case - 4	2	4	0.4
Case - 5	2	5	0.5
Case - 6	4	5	0.4
Case - 7	3	4	0.5
Case - 8	2	4	0.7
Case - 9	3	5	0.7
Case - 10	3	5	0.5
Case - 11	2	5	0.6

Table 2.1: Table presents different cases for *Simulation 1*. The true dimension R_{gen} is the dimension of vector object w_k using which data has been generated. The maximum dimension R is the dimension of vector object u_k using which the model has been fit. *Sparsity* refers to the fraction of generated $w_k = 0$, i.e., $(1 - \pi_w)$.

Cases	R	Sparsity
Case - 1	5	0.7
Case - 2	5	0.2

Table 2.2: Table presents different cases for *Simulation 2*. The maximum dimension R is the dimension of vector object u_k using which the model has been fit. *Simulation 2* only has one sparsity parameter $\pi_{2,w}$.

2.3.2 Results

In all simulation results shown in this section, our BNSP model is fitted with the choices of the hyper-parameters given by $\nu = 10$, $a_\Delta = 1$, $b_\Delta = 1$, $\zeta = 1$ and $\iota = 1$. Our extensive simulation studies reveal that both inference and prediction are robust to various choices of the hyper-

Cases	R	Node Sparsity	Edge Sparsity
Case - 1	5	0.7	0.5
Case - 2	5	0.2	0.5
Case - 3	5	0.7	0.3
Case - 4	5	0.2	0.7

Table 2.3: Table presents different cases for *Simulation 3*. The maximum dimension R is the dimension of vector object u_k using which the model has been fit. While *Simulation 2* only has a sparsity parameter, *Simulation 3* has a node sparsity ($\pi_{2,w}$) and an edge sparsity ($\pi_{3,w}$) parameter respectively.

parameters.

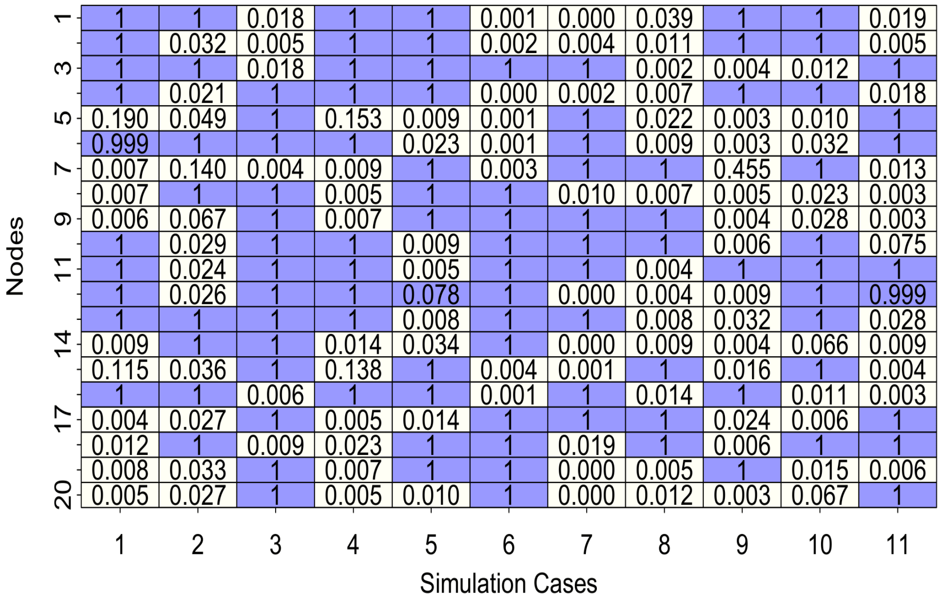
Identification of Influential Nodes

Figures 2.1 and 2.2 show the posterior probability of the k-th node being detected as influential, i.e., $P(\xi_k = 1 \,|\, Data)$, for each node and each case within *Simulation 1*, *Simulation 2* and *Simulation 3*. In the case of *Simulation 1*, the model is able to accurately identify nodes influencing the response for any reasonable cutoff threshold. Indeed, the receiver operating characteristic (ROC) curves associated with all these simulations have areas under the curve (AUC) very close to 1. For *Simulation 2*, the model performs very well in Case 1, which corresponds to relatively high node sparsity. However, when the node sparsity is relatively low (Case 2), using our default threshold of 0.5 leads to all nodes being identified as influential. While this is a somewhat disappointing result, we note that the model does tend to assign lower posterior probabilities to truly non-influential nodes. Hence, the associated AUC for Case 2 is nonetheless quite high (0.98). A similar pattern can be observed in *Simulation 3*, with the model performing very well when the node sparsity is high, and somewhat poorly when the node sparsity is low. Furthermore, it is interesting to observe that the level of edge sparsity has very little effect on the results when the node sparsity is high (Cases 1 and 3), but does impact the results when node sparsity is low (Cases 2 and 4). In particular, when node sparsity is low but edge sparsity is high, the model yields a very high number of false negatives for our default 0.5 detection threshold, while the reverse seems to be true when both node and edge sparsity are low. Digging a bit deeper, when both node and edge sparsity are low, the model assigns lower posterior probabilities to the non-influential nodes, resulting in a relatively high AUC (0.88), which is consistent with our *Simulation 2* results. On the other hand, when we have low node sparsity but high edge sparsity (the most unfavorable conditions for our model), the model struggles to even get the ranking of the nodes correctly, resulting in a relatively poor AUC (0.66). Among the competitor models, the only one that allows for the identification of influential nodes is the method of [113]. When this approach is applied to these simulations, it selects all nodes as significant in every case.

Parameter estimation

Tables 2.4, 2.5, 2.6 present the mean squared error (MSE) of all the competitors in *Simulations 1*, *2* and *3* respectively. Given that both the fitted network regression coefficient B and the true coefficient B_0 are symmetric, the MSE is calculated as $\frac{2}{V(V-1)}\sum_{k<l}(\hat{\gamma}_{k,l} - \gamma_{k,l,0})^2$, where $\hat{\gamma}_{k,l}$ is the point estimate of $\gamma_{k,l}$. For Bayesian models (including our proposed model), $\hat{\gamma}_{k,l}$ is taken to be the posterior mean of $\gamma_{k,l}$.

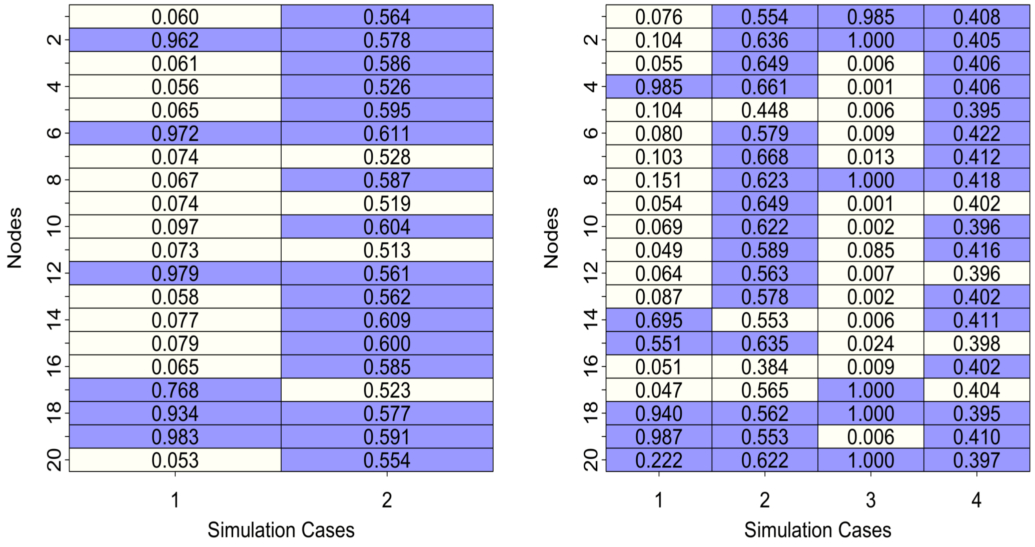
Table 2.4 shows that BNSP outperforms all its competitors in all cases of *Simulation*



Nodes	1	2	3	4	5	6	7	8	9	10	11
1	1	1	0.018	1	1	0.001	0.000	0.039	1	1	0.019
	1	0.032	0.005	1	1	0.002	0.004	0.011	1	1	0.005
3	1	1	0.018	1	1	1	1	0.002	0.004	0.012	1
	1	0.021	1	1	1	0.000	0.002	0.007	1	1	0.018
5	0.190	0.049	1	0.153	0.009	0.001	1	0.022	0.003	0.010	1
	0.999	1	1	1	0.023	0.001	1	0.009	0.003	0.032	1
7	0.007	0.140	0.004	0.009	1	0.003	1	1	0.455	1	0.013
	0.007	1	1	0.005	1	1	0.010	0.007	0.005	0.023	0.003
9	0.006	0.067	1	0.007	1	1	1	1	0.004	0.028	0.003
	1	0.029	1	1	0.009	1	1	1	0.006	1	0.075
11	1	0.024	1	1	0.005	1	1	0.004	1	1	1
	1	0.026	1	1	0.078	1	0.000	0.004	0.009	1	0.999
	1	1	1	1	0.008	1	1	0.008	0.032	1	0.028
14	0.009	1	1	0.014	0.034	1	0.000	0.009	0.004	0.066	0.009
	0.115	0.036	1	0.138	1	0.004	0.001	1	0.016	1	0.004
	1	1	0.006	1	1	0.001	1	0.014	1	0.011	0.003
17	0.004	0.027	1	0.005	0.014	1	1	1	0.024	0.006	1
	0.012	1	0.009	0.023	1	1	0.019	1	0.006	1	1
	0.008	0.033	1	0.007	1	1	0.000	0.005	1	0.015	0.006
20	0.005	0.027	1	0.005	0.010	1	0.000	0.012	0.003	0.067	1

Figure 2.1: Posterior probability that a node is influential, $P(\xi_k = 1 \mid Data)$, for each node and each of the 11 cases associated with *Simulation 1*. Dark cells correspond to the truly influential nodes.

1. In Cases 1-7, where the sparsity parameter is low to moderate, we perform overwhelmingly better than all the competitors. When the sparsity parameter in *Simulation 1* is high (Cases 8-9), our simulation scheme sets a very large proportion of $\gamma_{k,l,0}$'s to zero. As a result, BNSP only slightly outperforms Horseshoe and BLasso. BNSP also shows superior performance when the network predictor has modular structure (Cases 10-11). While BNSP is expected to perform much better than BLasso, Horseshoe and Lasso due to incorporation of network information, it is important to note that the carefully chosen global-local shrinkage prior with a well formulated hierarchical mean structure seems also to outperform [113], which is explicitly designed to account for the network structure.



Nodes	1	2
	0.060	0.564
2	0.962	0.578
	0.061	0.586
4	0.056	0.526
	0.065	0.595
6	0.972	0.611
	0.074	0.528
8	0.067	0.587
	0.074	0.519
10	0.097	0.604
	0.073	0.513
12	0.979	0.561
	0.058	0.562
14	0.077	0.609
	0.079	0.600
16	0.065	0.585
	0.768	0.523
18	0.934	0.577
	0.983	0.591
20	0.053	0.554

(a) Simulation 2

Nodes	1	2	3	4
	0.076	0.554	0.985	0.408
2	0.104	0.636	1.000	0.405
	0.055	0.649	0.006	0.406
4	0.985	0.661	0.001	0.406
	0.104	0.448	0.006	0.395
6	0.080	0.579	0.009	0.422
	0.103	0.668	0.013	0.412
8	0.151	0.623	1.000	0.418
	0.054	0.649	0.001	0.402
10	0.069	0.622	0.002	0.396
	0.049	0.589	0.085	0.416
12	0.064	0.563	0.007	0.396
	0.087	0.578	0.002	0.402
14	0.695	0.553	0.006	0.411
	0.551	0.635	0.024	0.398
16	0.051	0.384	0.009	0.402
	0.047	0.565	1.000	0.404
18	0.940	0.562	1.000	0.395
	0.987	0.553	0.006	0.410
20	0.222	0.622	1.000	0.397

(b) Simulation 3

Figure 2.2: Posterior probability that a node is influential, $P(\xi_k = 1 \,|\, Data)$, for each node and all cases associated with *Simulation 2* and *Simulation 3*. Dark cells correspond to the truly influential nodes.

For *Simulations 2* and *3*, Tables 2.5 and 2.6 demonstrate that, when node or edge sparsity are high, BNSP performs very similarly to Horseshoe. This might be due to the fact that a high degree of sparsity in the edge coefficients in the truth favors ordinary high dimensional regression. As node sparsity decreases, so that more edge coefficients are nonzero in the truth and the network structure in the predictors dominates, BNSP tends to show increasing advantage in terms of estimating the network coefficient B.

	MSE				
Cases	**BNSP**	Lasso	Relión(2017)	BLasso	Horseshoe
Case - 1	**0.008**	0.438	0.524	0.472	0.395
Case - 2	**0.007**	0.660	0.929	0.863	0.012
Case - 3	**0.006**	1.295	1.117	1.060	1.070
Case - 4	**0.008**	0.455	0.552	0.465	0.393
Case - 5	**0.006**	0.371	0.493	0.699	0.299
Case - 6	**0.008**	1.986	1.892	2.138	2.043
Case - 7	**0.009**	1.344	1.629	1.638	1.381
Case - 8	**0.004**	0.010	0.069	0.008	0.004
Case - 9	**0.004**	0.029	0.071	0.019	0.007
Case - 10	**0.091**	2.231	2.207	0.751	0.706
Case - 11	**0.003**	0.025	0.047	0.018	0.012

Table 2.4: Performance of BNSP vis-a-vis competitors for cases in *Simulation 1*. Parametric inference in terms of point estimation of edge coefficients has been captured through the Mean Squared Error (MSE). The minimum MSE among competitors for any case is made bold.

	MSE				
Cases	**BNSP**	Lasso	Relión(2017)	BLasso	Horseshoe
Case - 1	0.015	0.012	0.036	0.008	**0.006**
Case - 2	**0.629**	0.843	0.859	0.836	0.948

Table 2.5: Performance of BNSP vis-a-vis competitors for cases in *Simulation 2*. Parametric inference in terms of point estimation of edge coefficients has been captured through the Mean Squared Error (MSE). The minimum MSE among competitors for any case is made bold.

Identifying influential edges

Tables 2.7 and 2.8 show the true positive rates (TPR) and false positive rates (FPR) associated with the detection of important edges for *Simulation 1* and *Simulation 3* using BNSP,

	MSE				
Cases	**BNSP**	Lasso	Relión(2017)	BLasso	Horseshoe
Case - 1	0.005	0.006	0.017	0.004	**0.002**
Case - 2	**0.457**	0.636	0.617	0.669	0.629
Case - 3	0.008	**0.004**	0.036	0.005	**0.004**
Case - 4	**0.131**	0.178	0.145	0.182	0.145

Table 2.6: Performance of BNSP vis-a-vis competitors for cases in *Simulation 3*. Parametric inference in terms of point estimation of edge coefficients has been captured through the Mean Squared Error (MSE). The minimum MSE among competitors for any case is made bold.

Lasso and [113]. The results for our method are based on controlling the FDR at 0.05 using the algorithm described in Appendix C. In *Simulation 1*, BNSP outperforms Lasso and [113],

	BNSP		Lasso		Relión (2017)	
Cases	TPR	FPR	TPR	FPR	TPR	FPR
Case - 1	0.69	0	0.60	0.29	1	1
Case - 2	1	0.02	0.86	0.25	1	1
Case - 3	0.96	0	0.14	0.05	1	1
Case - 4	1	0.08	0.53	0.23	1	1
Case - 5	0.80	0.08	0.47	0.27	1	1
Case - 6	0.92	0	0.59	0.29	1	1
Case - 7	0.97	0.04	0.60	0.27	1	1
Case - 8	0.86	0.01	0.73	0.22	1	1
Case - 9	0.70	0.02	0.87	0.29	1	1
Case - 10	0.84	0	0.58	0.18	1	1
Case - 11	0.85	0.04	0.61	0.17	1	1

Table 2.7: True Positive Rates (TPR) and False Positive Rates (FPR) for edges for cases in *Simulation 1*.

although when node sparsity becomes high, Lasso becomes competitive with BNSP. Lasso is also competitive with BNSP in *Simulation 3*, although in this case all models tend to perform poorly when node sparsity is low but edge sparsity is relatively high. [113] appears to have a very poor performance, as it identifies all edges as important in all the simulation scenarios, resulting in high FPRs.

	BNSP		Lasso		Relión(2017)	
Cases	TPR	FPR	TPR	FPR	TPR	FPR
Case - 1	0.71	0	0.86	0.20	1	1
Case - 2	0.35	0.12	0.36	0.21	1	1
Case - 3	1	0.02	0.91	0.15	1	1
Case - 4	0.91	0.86	0.23	0.07	1	1

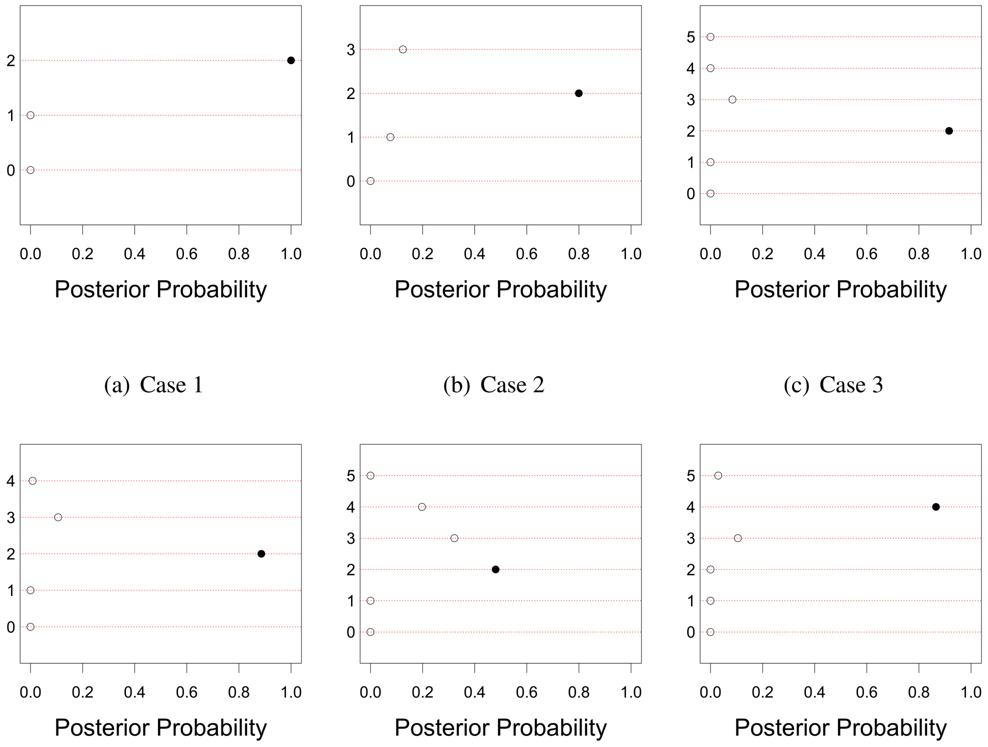
Table 2.8: True Positive Rates (TPR) and False Positive Rates (FPR) for edges for cases in *Simulation 3*.

Inference on the effective dimensionality

Figures 2.3 and 2.4 present posterior probabilities of effective dimensionality in all 11 cases in *Simulation 1*, which is the only setting in which the true dimension of the latent space is known. In all 11 cases the posterior mode corresponds to the true dimension of the latent space.

Predictive Inference

We compare the out-of-sample predictive ability of the different models based on the point prediction and characterization of predictive uncertainties using test samples of size



(a) Case 1

(b) Case 2

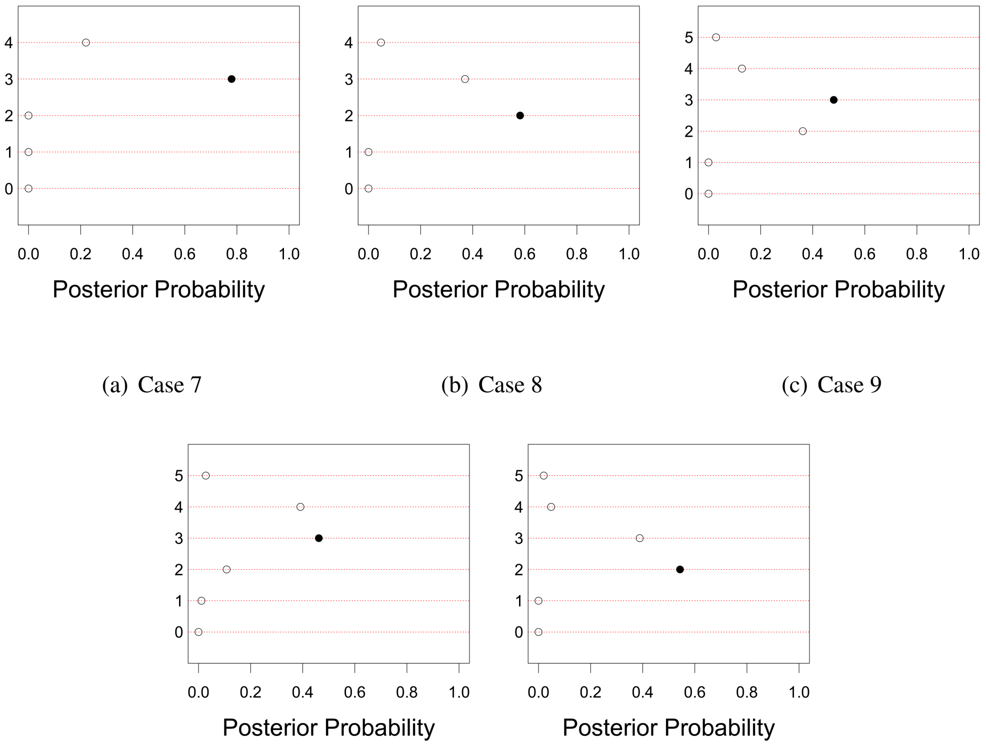
(c) Case 3

(d) Case 4

(e) Case 5

(f) Case 6

Figure 2.3: Posterior probability distributions of the effective dimensionality in cases $1-6$ in *Simulation 1*. Filled bullets indicate the true value of effective dimensionality.



(a) Case 7

(b) Case 8

(c) Case 9

(d) Case 10

(e) Case 11

Figure 2.4: Posterior probability distributions of the effective dimensionality in cases $7 - 11$ in *Simulation 1*. Filled bullets indicate the true value of effective dimensionality.

$n_{pred} = 30$. To assess point prediction, we employ the mean squared prediction error (MSPE). As measures of predictive uncertainty, we provide coverage and length of 95% predictive intervals. For frequentist competitors, 95% predictive intervals are obtained by using predictive point estimates plus and minus 1.96 times standard errors.

Tables 2.9, 2.10, 2.11 and 2.12 show results for *Simulation 1*, *Simulation 2* and *Simulation 3*. For *Simulation 1*, BNSP clearly outperforms other competitors in terms of point prediction. Horseshoe becomes competitive in cases with a higher degree of sparsity (Cases 2, 8 and 9). Lasso and BLasso are competitive only in Case 8, while our approach seems to dominate the method of [113] in all cases. In terms of prediction uncertainty, BNSP tends to generate by far the shortest intervals, but also to exhibit a slight under-coverage, particularly in Cases 5 and 11. As in the case of point prediction, Horseshoe seems to yield results that are very similar to those of our model in Cases 2, 8 and 9.

In the case of *Simulations 2* and *3*, BNSP seems to outperform all other methods in situations where the node sparsity is low. Note that this is the opposite of what we found when investigating the performance of the model to identify influential nodes. Similar observations can be made with respect to the coverage and length of the intervals. BNSP seems to have the shortest intervals and about nominal coverage in Case 2 of *Simulation 1* and in Cases 2 and 4 of *Simulation 3*, making it the obvious top performer. For the remaining cases in *Simulations 2* and *3*, Horseshoe seems to be at least competitive with our method.

	MSPE				
Cases	**BNSP**	Lasso	Relión(2017)	BLasso	Horseshoe
Case - 1	**0.012**	0.324	0.537	0.405	0.421
Case - 2	**0.008**	0.707	0.574	0.638	0.013
Case - 3	**0.007**	0.442	0.498	0.487	0.409
Case - 4	**0.012**	0.494	0.571	0.426	0.317
Case - 5	**0.014**	0.412	0.517	0.759	0.238
Case - 6	**0.005**	0.447	0.539	0.821	0.745
Case - 7	**0.007**	0.533	0.605	0.572	0.563
Case - 8	**0.039**	0.075	0.365	0.060	0.046
Case - 9	**0.044**	0.236	0.486	0.151	0.067
Case - 10	**0.029**	0.830	0.816	0.381	0.385
Case - 11	**0.062**	1.000	0.446	0.230	0.153

Table 2.9: MSPE under the BNSP vis-a-vis competitors for cases in *Simulation 1*. Lowest MSPE for any case is made bold.

Sensitivity to the choice of R

In order to examine the behavior of the model with increasing R, we rerun our model for each simulation scenario with $R = 10, 15$ and 20 (in addition to our original choice of R). For the sake of brevity, we only provide results for the data corresponding to Case 9 in *Simulation 1* (see Table 2.14). The behavior of all metrics is quite stable. The only summary that seems to be slightly affected are the posterior means of R_{eff} and the length of the 95% credible intervals, which increase by about 16% and 11% respectively when we go from $R = 5$ to $R = 20$.

	Coverage of 95% PI				
Case - 1	0.867	0.967	0.567	0.967	0.967
Case - 2	0.933	0.967	0.867	1.000	1.000
Case - 3	0.933	0.900	0.767	1.000	1.000
Case - 4	0.900	0.900	0.567	0.967	0.967
Case - 5	0.800	1.000	0.700	0.933	0.933
Case - 6	1.000	0.967	0.667	0.900	0.967
Case - 7	0.933	0.967	0.633	1.000	0.967
Case - 8	0.933	1.000	0.900	1.000	0.967
Case - 9	0.967	1.000	0.733	0.933	0.933
Case - 10	1.000	0.933	0.900	1.000	1.000
Case - 11	0.833	0.333	0.867	1.000	0.900
	Length of 95% PI				
Case - 1	5.093	41.528	16.789	39.656	32.868
Case - 2	5.040	49.254	27.983	58.449	9.366
Case - 3	5.900	38.259	30.126	67.251	61.534
Case - 4	5.321	37.814	21.848	39.728	33.529
Case - 5	4.461	41.251	22.115	43.027	30.132
Case - 6	11.053	67.922	36.434	75.322	76.089
Case - 7	5.214	70.655	31.746	83.132	68.103
Case - 8	4.753	23.964	12.122	8.578	5.846
Case - 9	4.780	14.397	8.227	8.783	5.868
Case - 10	21.571	75.309	61.221	55.603	69.886
Case - 11	3.874	13.216	10.419	11.485	6.618

Table 2.10: Coverage and length of 95% predictive intervals (PIs) under the BNSP vis-a-vis competitors for cases in *Simulation 1*.

Scalability and Computation Time

Computation times for competing methods are provided in Table 2.15. It is to be noted that computation times for frequentist methods and BNSP are not directly comparable as

	MSPE				
Cases	**BNSP**	Lasso	Relión(2017)	BLasso	Horseshoe
Case - 1	0.213	0.144	0.335	0.131	**0.122**
Case - 2	**0.426**	0.532	0.621	0.568	0.626
	Coverage of 95% PI				
Case - 1	0.900	1.000	0.900	0.933	0.900
Case - 2	0.933	0.800	0.600	0.967	0.933
	Length of 95% PI				
Case - 1	8.323	15.940	9.544	7.957	6.079
Case - 2	43.834	34.413	24.117	45.959	43.219

Table 2.11: MSPE, coverage and length of 95% predictive intervals (PIs) under the BNSP vis-a-vis competitors for cases in *Simulation 2*. Lowest MSPE for any case is made bold.

BNSP is based on $50,000$ MCMC iterations while the former methods yield results just after a few of iterations. For the Bayesian method BNSP, the table records run time (in seconds) per equivalent effective posterior sample, to account for the fact that posterior samples are correlated. In absence of any open source code, we have implemented [113] by ourselves with the run time provided in the table. Perhaps a more efficient implementation of [113] could reduce its run time. As expected, the computation time of BNSP grows approximately linearly with V and n^3.

	MSPE				
Cases	**BNSP**	Lasso	Relión(2017)	BLasso	Horseshoe
Case - 1	0.108	0.183	0.452	0.138	**0.101**
Case - 2	**0.677**	0.959	0.817	0.869	0.888
Case - 3	0.066	0.049	0.354	0.050	**0.047**
Case - 4	**0.604**	0.877	0.732	0.781	0.720
	Coverage of 95% PI				
Case - 1	1.000	1.000	0.900	0.933	0.967
Case - 2	0.967	0.700	0.533	1.000	1.000
Case - 3	0.900	1.000	0.900	1.000	0.833
Case - 4	0.967	0.400	0.533	0.967	0.967
	Length of 95% PI				
Case - 1	6.371	13.080	7.877	6.508	5.268
Case - 2	41.492	26.028	18.387	51.459	48.694
Case - 3	5.069	22.774	11.760	5.980	4.005
Case - 4	18.704	7.397	8.547	22.049	20.227

Table 2.12: MSPE, coverage and length of 95% predictive intervals (PIs) under the BNSP vis-a-vis competitors for cases in *Simulation 3*. Lowest MSPE for any case is made bold.

2.4 Application to Human Brain Network Data

Human creativity has been at the crux of the evolution of the human civilization, and has been the topic of research in several disciplines, including neuroscience. Though creativity can be defined in numerous ways, one could envision a creative idea as one that is unusual as well as effective in a given social context [44]. Neuroscientists generally concur that a coalescence of several cognitive processes determines the creative process, which often involves

Cases	R_{gen}	R	Sparsity	**BNSP**	Lasso	Relión(2017)	BLasso	Horseshoe
Case - 1	2	2	0.5	0.009	0.438	0.524	0.472	0.395
Case - 2	2	3	0.6	0.007	0.660	0.929	0.863	0.012
Case - 3	2	5	0.3	0.006	1.295	1.117	1.060	1.070
Case - 4	2	5	0.4	0.006	0.371	0.493	0.699	0.298
Case - 5	3	5	0.5	0.009	1.344	1.629	1.638	1.381
Case - 6	4	5	0.4	0.006	3.054	2.601	2.680	3.284
Case - 7	2	4	0.5	0.009	0.438	0.524	0.472	0.395

Table 2.13: Performance of Bayesian Network Regression vis-a-vis competitors. Predictive point estimation has been captured through the Mean Squared Prediction Error (MSPE).

R	MSE	MSPE	Coverage	Length of 95% PI	Posterior Mean of R_{eff}
5	0.0044	0.044	0.967	4.780	2.83
10	0.0038	0.0437	0.967	4.996	2.95
15	0.0039	0.0438	0.967	5.362	3.23
20	0.0041	0.0433	0.967	5.341	3.31

Table 2.14: Model behavior in terms of model performance metrics with changing values of R for data corresponding to *Simulation 1*, Case 9. We report MSE, MSPE, length and coverage of 95% predictive intervals and the posterior mean of effective dimensionality R_{eff}.

a *divergence of ideas* to conceivable solutions for a given problem. To measure the creativity of an individual, [80] propose the CCI, which is formulated by linking measures of divergent thinking and creative achievement to cortical thickness of young (23.7 ± 4.2 years), healthy subjects. Three independent judges grade the creative products of a subject from which the "composite creativity index" (CCI) is derived.

Along with CCI measurements, brain network information for $n = 79$ subjects is gathered using diffusion weighted magnetic resonance imaging (DWI). DWI is an imaging tech-

V	n	BNSP	Lasso	Relión(2017)
20	70	0.1392	0.3606	2.3954
20	100	0.1594	0.6693	3.1306
20	150	0.2069	0.4900	3.3002
40	70	0.6435	0.5150	30.0046
40	100	0.8296	0.4829	39.9697
40	150	1.1467	0.8013	54.5337
60	70	2.7874	1.0954	150.9617
60	100	3.7153	0.7423	200.4439
60	150	5.3052	0.8603	285.5792
80	70	8.1378	1.7925	435.5506
80	100	11.6997	1.3206	645.1986
80	150	17.2309	2.0388	995.7408
100	70	20.1989	0.8699	1165.969
100	100	26.5559	1.3059	1467.85
100	150	31.4653	1.5472	2031.46

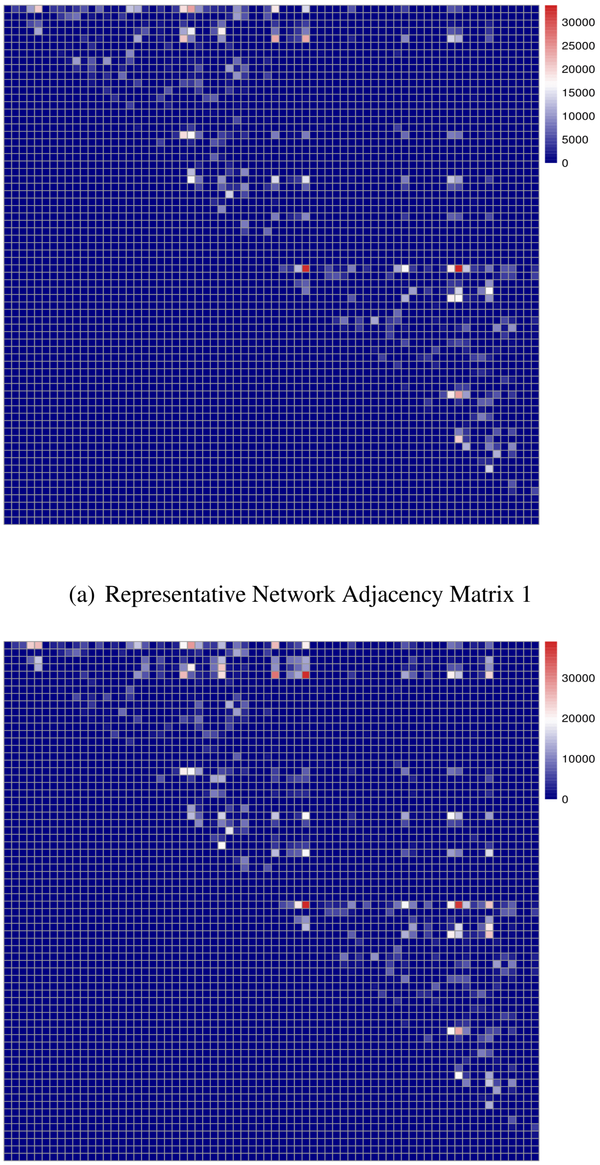
Table 2.15: Computation time of competing methods for different values of sample size (n) and number of nodes (V). For the Bayesian method BNSP, the table records run time (in seconds) per equivalent effective posterior sample for BNSP, to account for the fact that posterior samples are correlated. The last two columns record total run time for frequentist methods.

nique that enables measurement of the restricted diffusion of water in tissue in order to produce neural tract images. The brain imaging data we use has been pre-processed using the NDMG pre-processing pipeline [82, 81, 83]. In the context of DWI, the human brain is divided according to the Desikan atlas [32] that identifies 34 cortical regions of interest (ROIs) in each of the left and right hemispheres of the human brain, implying 68 cortical ROIs in all. A 'brain network' for each subject is represented by a symmetric adjacency matrix whose rows and columns correspond to different ROIs and entries correspond to estimates of the number of ‘fibers’ connecting pairs of brain regions. A “lobe” in a human brain is composed of a number of ROIs. According to Desikan atlas, brain consists of 12 lobes, 6 on right and left hemisphere each. Figure 2.5 shows maps of the brain network for two representative individuals in the sample.

In this Section we are interested in predicting the CCI of a subject from his/her brain network, and to identify brain regions (nodes in the brain network) that are involved with creativity, as well as influential connections between different brain regions. Before carrying out our analysis, each cell of the adjacency matrix is standardized by subtracting the mean and dividing by the standard deviation with respect to all $n = 79$ samples. CCI is also standardized in a similar fashion. The MCMC chain for our model is run for $50,000$ iterations, with the first $40,000$ iterations discarded as burn-in. Convergence is assessed by comparing different simulated sequences of representative parameters started at different initial values [52]. We monitor the auto-correlation plots and effective sample sizes. Prior distributions for all the parameters are chosen as in the simulation studies.

2.4.1 Findings from BNSP

For the purpose of this data analysis, BNSP was fitted with $R = 5$. Later, we show that the results are robust to moderate increases in the value of R. A posteriori, the mean of the effective dimension R_{eff} was 3. Figure 2.6 shows the posterior means of the latent positions $u_1,\dots,u_V$ for the two highest-variance components of the latent space. The clump of nodes located at the origin all correspond to ROIs that our method deems to be non-influential on the response (see discussion below).



(a) Representative Network Adjacency Matrix 1

(b) Representative Network Adjacency Matrix 2

Figure 2.5: Maps of the brain network (weighted adjacency matrices) for two representative individuals in the sample. Since the (k,l)-th off-diagonal entry in any adjacency matrix corresponds to the number of *fibers* connecting the k-th and the l-th ROIs, the adjacency matrices are symmetric. Hence the figure only shows the upper triangular portion.

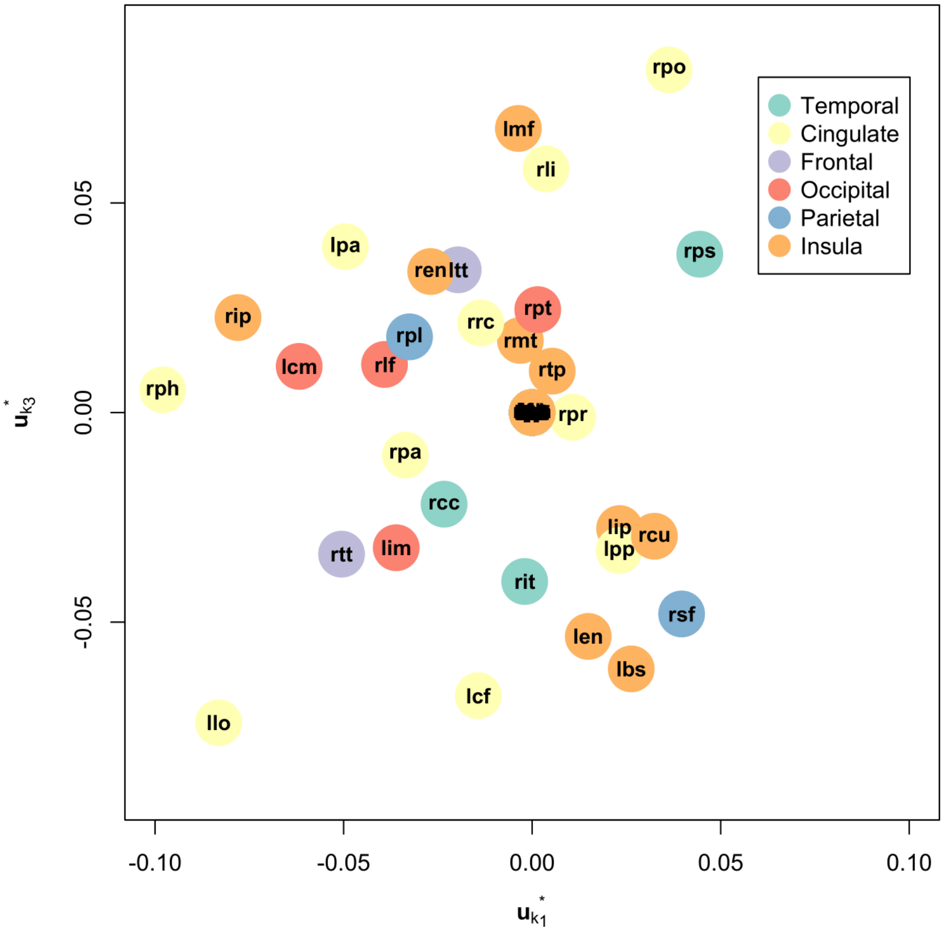


Figure 2.6: Posterior means of the latent positions $u_1,\ldots,u_V$ for the two highest-variance components of the latent space.

Recall that the k-th node is identified as *influential* if $P(\xi_k = 1 | Data)$ exceeds 0.5. In this dataset, this criteria identifies 41 ROIs out of 68 as influential. Of the influential ROIs, 19 belong to the left hemisphere and 22 belong to the right hemisphere (see Table 2.17). This coincides with results that have been previously presented in the literature. A large number of the 41 influential nodes detected by our method are part of the *frontal* (16) and *temporal* (7) cortices in both hemispheres. The frontal cortex has been scientifically associated with divergent thinking and problem solving ability, in addition to motor function, spontaneity, memory, language, initiation, judgement, impulse control and social behavior [131]. Some of the other functions directly related to the frontal cortex seem to be behavioral spontaneity, interpreting environmental feedback and risk taking [112, 99, 87]. Similarly, [43] report *de novo* artistic expression to be associated with the frontal and temporal regions. Our method also finds a strong relationship between creativity and the *right parahippocampal gyrus* and *right inferior parietal lobule*, regions found to be involved with creativity by a few earlier scientific studies, see e.g., [23].

Our results also show substantial overlap with those of [80], in which a regression model is used to understand the relationship between CCI and ROI-specific measures to account for the relationship between creativity and different brain regions. In particular, both approaches identify the *middle frontal gyrus*, the *left cingulate cortex*, the *left orbitofrontal* region, the *left lingual* region, the *left fusiform*, the *right cuneus*, the *right superior parietal lobule*, the *superior parietal* lobules and the *right singulate* regions as influencing CCI. However, although there is significant intersection between the findings of [80] and our method, there are a few regions that we detect as influential and they do not, and vice versa. For example, our model detects the *right precuneus* and the *supramarginal* regions in both the hemispheres to be significantly related to CCI, while [80] do not. On the other hand, they identify the *right angular* region to be significant while we do not. Applying the method of [113] to our dataset leads to the identification of 65 out of 68 ROIs as influential. The three regions that are found to be uninfluential are the *frontalpole*, *temporalpole* and the *transversetemporal* regions in the right hemisphere.

Along with influential ROIs, we are interested in identifying the statistically significant edges or connections between the 68 ROIs. Figure 2.7 plots the 523 interconnections that appear to be influential (out of a total of 2,016), controlling for a 0.05 FPR.

Our interest turns now to the predictive ability of the Bayesian network regression model. Table 2.16 reports the mean squared prediction error (MSPE), length and coverage of 95% predictive intervals for a ten-fold cross-validation exercise. As reference, we also present MSPE, length and coverage values for Lasso, BLasso and [113].

BNSP clearly outperforms all other methods in terms of point prediction. In terms of prediction intervals, all methods perform similarly. However, note that, while the coverage of BNSP is slightly under our target, the coverage all of the other methods is slightly above target.

Finally, we assess the sensitivity of the model to the choice of R. Table 2.18 shows nearly identical results by choosing $R = 5, 6, 7$ and 10, suggesting that our original choice of R is sufficiently large for this application.

	BNSP	Lasso	Relión(2017)	BLasso	Horseshoe
MSPE	**0.77**	0.98	0.98	1.84	1.78
Coverage of 95% PI	0.92	0.97	0.97	0.97	0.93
Length of 95% PI	3.73	3.88	3.89	3.40	4.99

Table 2.16: Predictive performance of competitors in terms of mean squared prediction error (MSPE), coverage and length of 95% predictive intervals, obtained through 10-Fold Cross Validation in the context of real data. Note that since the response has been standardized, an MSPE value greater than or around 1 will denote an inconsequential analysis.

2.5 Conclusion

This chapter proposes a novel Bayesian framework to address a regression problem with a continuous response and network-valued predictors. Our contribution lies in carefully constructing a novel class of network shrinkage priors that account for the correlation in the regression coefficients that is expected from the relational nature of the predictor. Empirical results from simulation studies show that our method is superior to popular alternatives in situations where the level of node sparsity is at least moderate, and mostly competitive in other circumstances. In our analysis of the Composite Creativity Index, the results generated by our model largely agree with those previously reported in the literature.

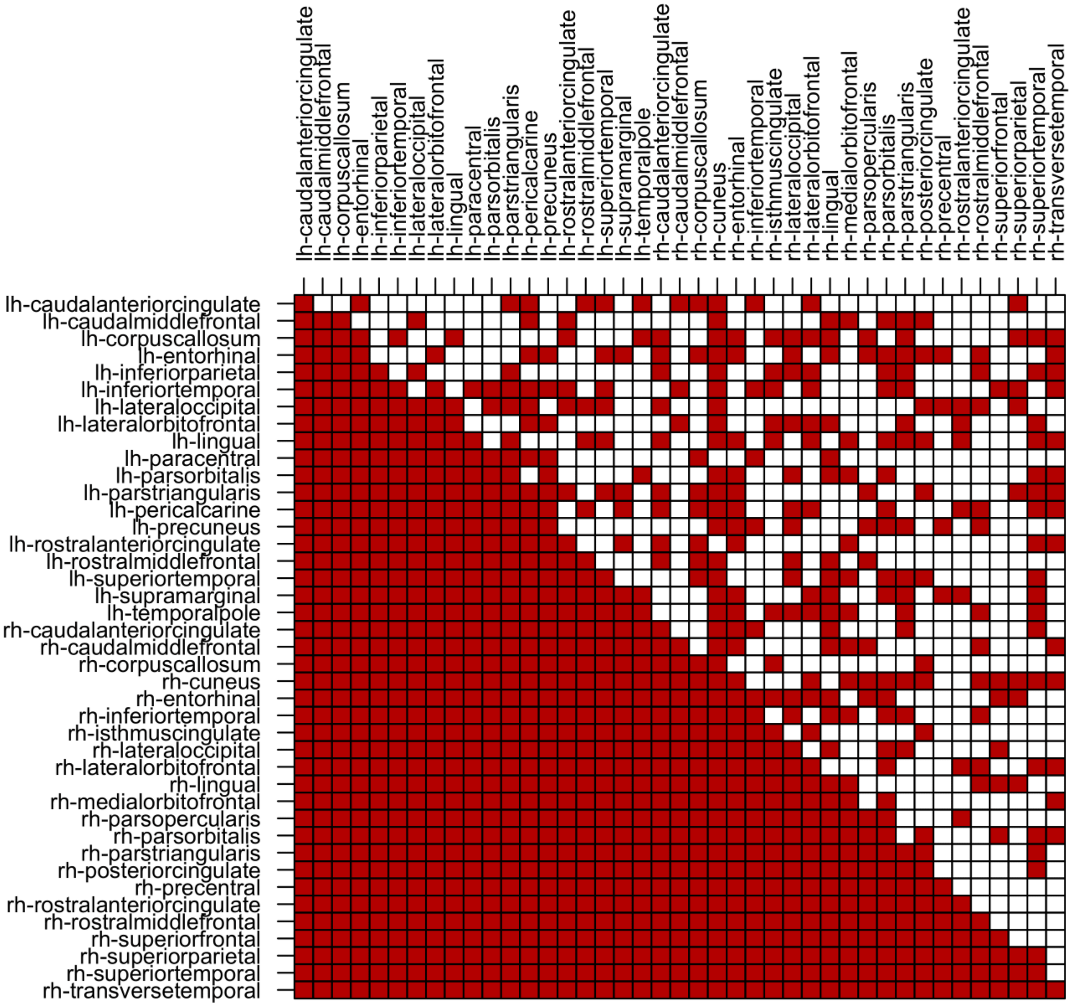


Figure 2.7: Significant inter-connections detected among influential brain regions of interest (ROIs) in the Desikan atlas. White cells show significant nodal associations among ROIs. Prefix ‘lh-’ and ‘rh-’ in the ROI names denote their positions in the left and right hemispheres of the brain respectively.

Left Hemisphere Lobes					
Temporal	**Cingulate**	**Frontal**	**Occipital**	**Parietal**	**Insula**
fusiform	rostral-anteriorcingulate	caudal-middlefrontal	cuneus	postcentral	
inferiortemporal	caudal-anteriorcingulate	lateral-orbitofrontal	lingual	supramarginal	
transversetemporal	isthmus-cingulate	pars-opercularis	pericalcarine		
		pars-triangularis			
		rostral-middlefrontal			
		superior-frontal gyrus			
		frontalpole			
		medial-orbitofrontal			
Right Hemisphere Lobes					
Temporal	**Cingulate**	**Frontal**	**Occipital**	**Parietal**	**Insula**
middle-temporal	caudal-anteriorcingulate	caudal-middlefrontal	cuneus	precuneus	insula
superior-temporal	isthmus-cingulate	lateral-orbitofrontal	lateral-occipital	superior-parietal	
entorhinal		medial-orbitofrontal	pericalcarine	supramarginal-gyrus	
fusiform		pars-orbitalis	lingual		
		precentral			
		rostral-middlefrontal			
		superior-frontal			
		pars-triangularis			

Table 2.17: Brain regions (ROIs) detected as influential for the composite creativity index by BNSP.

	BNSP ($R = 5$)	BNSP ($R = 6$)	BNSP ($R = 7$)	BNSP ($R = 10$)
MSPE	0.77	0.87	0.83	0.85
Coverage of 95% PI	0.92	0.92	0.92	0.91
Length of 95% PI	3.73	3.78	3.81	3.84
Posterior Mean of R_{eff}	2.41	2.69	2.46	2.57

Table 2.18: Predictive performance of BNSP with $R = 5, 6, 7, 10$ to assess the sensitivity of predictive inference with the choice of R.

Chapter 3

High Dimensional Bayesian Network Classification with Network Global-Local Shrinkage Priors

3.1 Introduction

Chapter 2 discusses the network regression problem with a continuous response and an undirected network predictor. However, there are pertinent biological and physiological studies where a network along with a binary response is obtained for each subject. The goal of these studies is usually to classify the networks according to the binary response and predict the associated binary response from a network. We refer to this problem as the network or graph classification problem. Additionally, Chapter 2 focuses on a specific network shrinkage prior, whereas this chapter generalizes the inference to a class of network global-local shrinkage priors, which includes the prior specification in Chapter 2 as a special case.

Earlier literature on network or graph classification has been substantially motivated by the problem of classification of chemical compounds [129], [65], where a graph represents a compound's molecular structure. In such analyses, certain discriminative patterns in a graph are identified and used as features for training a standard classification method [31], [42]. Another type of method is based on graph kernels [135], which defines a similarity measure between two networks. Both of these approaches are computationally feasible only for small networks, do not account for uncertainty, and do not facilitate influential network node identification. When the number of network nodes is moderately large, a common approach to network classification is to use a few summary measures (average degree, clustering coefficient, or average path length) from the network and then apply statistical procedures in the context of standard classification methods (see, for e.g., [11] and references therein). These procedures have been recently employed in exploring the relationship between the brain network and neuropsychiatric diseases, such as Parkinson's [107] and Alzheimer's [29], but the analyses are sensitive to the chosen network topological measures, with substantially different results obtained for different types of summary statistics. Indeed, global summary statistics collapse all local network information, which can affect the accuracy of classification. Furthermore, identification of the impact of specific nodes on the response, which is of clear interest in our setting, is not feasible. As with network regression problems, an alternate approach proceeds to vectorize the network predictor and treat edge weights together as a long vector followed by developing a high dimensional regression model with this long vector of edge weights as predictors [114]; [27];

[146]. This approach can take advantage of the recent developments in high dimensional binary regression, consisting of both penalized optimization [133] and Bayesian shrinkage [109]; [17]; [5] perspectives. However, as mentioned in Chapter 2, this treats the links of the network as exchangeable, ignoring the fact that coefficients involving common nodes can be expected to be correlated a priori. In a related work, [136] propose to look for a minimal set of nodes which best explains the difference between two groups of networks. This requires solving a combinatorial problem. Again, [35] propose a high dimensional Bayesian tensor factorization model for a population of networks that allows to test for local edge differences between two groups of subjects. Both of these approaches tend to focus mainly on classification and are not designed to detect important nodes and edges impacting the response.

Our goal in this chapter is to develop a high-dimensional Bayesian network classifier that additionally infers on influential nodes and edges impacting classification. To achieve this goal, we formulate a high dimensional logistic network regression model with the binary response regressed on the network predictor corresponding to each subject. The network predictor coefficient is assigned a prior from the class of *Bayesian network global-local shrinkage priors* discussed in this chapter. The proposed prior imparts low-rank and near sparse structures a priori on the network predictor coefficient. The low-rank structure of the coefficient is designed to address the transitivity effect on the network predictor coefficient and captures the effect of network edge coefficients on classification due to the interaction between nodes. On the other hand, the near sparse structure accounts for the residual effect due to edges.

One important contribution of this chapter is a careful study of the asymptotic properties of the proposed binary network classification (BNC) framework. In particular, we focus on consistency properties for the posterior distribution of the BNC framework using a specific network global-local shrinkage prior, namely the *Bayesian Network Lasso prior*. Theory of posterior contraction for high dimensional regression models has gained traction lately, though the literature is less developed in shrinkage priors compared to point-mass priors. For example, [19] and [7] have established posterior concentration and variable selection properties for certain point-mass priors in the normal-means models. The latter chapter also establishes asymptotically nominal coverage of Bayesian credible sets. Results on posterior concentration and variable selection in high dimensional linear models are also established by [18] and [98] for certain point-mass priors. In contrast, literature on posterior contraction properties for high dimensional Bayesian shrinkage priors is relatively limited. To this end, [6] were the first to show posterior consistency in the ordinary linear regression model with shrinkage priors for low-dimensional settings under the assumption that the number of covariates *does not* exceed the number of observations. Using direct calculations, [134] show that the posterior based on the ordinary horseshoe prior concentrates at the optimal rate for normal-mean problems. Recently, [128] considers a general class of continuous shrinkage priors and obtains posterior contraction rates in ordinary high dimensional linear regression models. In the same vein, [143] offers analysis of posterior concentration for logistic regression models with shrinkage priors on coefficients. While [143] are the first to delineate a theoretical approach for ordinary high dimensional binary classification models with shrinkage priors, the study of posterior contraction properties for more structured binary network classification problems in the Bayesian paradigm has not appeared in the literature. In fact, developing the theory for Bayesian network classification with the Bayesian Network Lasso prior proposed in this chapter is faced with two major challenges. First, the novel Bayesian Network Lasso prior imparts a more complex prior structure (incorporating a low-rank structure in the prior mean of edge coefficients, as described in Chapter 2) than that in [143], introducing additional theoretical challenges. Second, we aim at proving a challenging but practically desirable result of asymptotically optimal classification when the number of edges in the network predictor grows at a super-linear rate as a function of the sample size. Both of these present obstacles which we overcome in this work. The theoretical results provide insights on how the number of nodes in the network predictor, or the sparsity in the true network predictor coefficients should vary with sample size n to achieve asymptotically optimal classification. We must mention that developing a similar theory for the Bayesian Network Horseshoe prior proposed in this chapter faces more challenges due to complex prior structure in parameters. We plan to tackle that problem as part of future work.

Section 3.2 develops the model and the prior distributions. Section 3.3 discusses theoretical developments justifying the asymptotically desirable prediction from the proposed model. Section 3.4 details posterior computation. Results from various simulation experiments and a brain connectome data analysis have been presented in Sections 3.5 and 3.6 respectively. Finally, Section 3.7 concludes the chapter with a brief discussion of the proposed methodology.

3.2 Model Formulation

In the context of network classification, we propose the high dimensional logistic regression model of the binary response $y_i \in \{0,1\}$ on the undirected network predictor A_i as

$$y_i \sim Ber\left[\frac{\exp(\psi_i)}{1+\exp(\psi_i)}\right], \ \psi_i = \mu + \langle A_i, \Gamma \rangle_F, \tag{3.1}$$

where Γ is a $V \times V$ symmetric network coefficient matrix whose (k,l)th element is given by $\gamma_{k,l}/2$, with $\gamma_{k,k} = 0$, for all $k = 1,...,V$.

Model (3.1) can be expressed in the form of a generalized linear model. To be more specific, $\langle A_i, \Gamma \rangle_F = \sum\limits_{1 \le k < l \le V} a_{i,k,l}\gamma_{k,l}$, so that $\psi_i = \mu + \sum\limits_{1 \le k < l \le V} a_{i,k,l}\gamma_{k,l}$ and the probability mass function of y_i can be written as

$$p(y_i) = \frac{\exp(\psi_i)^{y_i}}{1 + \exp(\psi_i)} \tag{3.2}$$

Note that, if $x_i = (a_{i,1,2},...,a_{i,(V-1),V})' \in \mathbb{R}^{V(V-1)/2}$ is the collection of all upper triangular elements of A_i, and $\gamma = (\gamma_{1,2},...,\gamma_{(V-1),V})' \in \mathbb{R}^{V(V-1)/2}$ is the vector of corresponding upper triangular elements of 2Γ, then (3.1) can be written as

$$y_i \sim Ber\left(f_\gamma(x_i)\right),\ f_\gamma(x_i) = \frac{\exp(\mu + x_i'\gamma)}{(1 + \exp(\mu + x_i'\gamma))}. \tag{3.3}$$

Although the binary network regression model is proposed for the logit link, it assumes natural extension for any other link function. The next section describes a class of network global-local shrinkage priors on network coefficients.

3.2.1 Bayesian network global-local shrinkage prior on the network predictor coefficient

In this chapter, we propose the network global-local shrinkage prior given by,

$$\gamma_{k,l}|s_{k,l},\sigma^2 \sim N(u_k'\Lambda u_l, \sigma^2 s_{k,l}^2),\ \sigma \sim H_1(\cdot),\ s_{k,l} \sim H_2(\cdot). \tag{3.4}$$

Note that this framework a priori centers $\gamma_{k,l}$ at a low-rank decomposition and controls the spread of the prior distribution of $\gamma_{k,l}$ using a global-local shrinkage prior. The formulation includes a wide variety of network shrinkage priors by choosing different functions $H_1(\cdot)$ and $H_2(\cdot)$. For example, Chapter 2 has investigated a particular class of such prior distributions, obtained by choosing $H_1(\sigma) = \delta_1(\sigma)$, where $\delta_1(\sigma)$ is the Dirac-delta function that is defined as $\delta_1(\sigma) = 1$ if $\sigma = 1$, and 0 otherwise; and $H_2(s_{k,l}^2)$ as an exponential density, referred to as the *Network Lasso prior*. To show the generality of (3.4), we additionally investigate performance of (3.4) in binary regression with $s_{k,l} \sim C^+(0,1)$ and $\sigma \sim C^+(0,1)$. The resulting prior is referred to as the *Network Horseshoe prior*. The rest of the hierarchy on λ_r's, u_k's follows as in Chapter 2.

3.3 Posterior Contraction of the Binary Network Classification Model

This section establishes convergence results for (3.1) with $\gamma_{k,l}$'s following the Bayesian Network Lasso shrinkage prior. From the hierarchical specification given in (3.4), the Bayesian Network Lasso shrinkage prior is given by $\gamma_{k,l}|s_{k,l} \sim N(u_k'\Lambda u_l, s_{k,l}^2)$, $s_{k,l}^2 \sim Exp(\theta_n/2)$. For the theoretical study, a common practice is to fix θ_n as a function of n [5]. Our theoretical investigations will also fix θ_n (the exact expression is given in Condition (F) in the next subsection) with the fixed values specified later.

Here we consider an asymptotic setting in which the number of nodes in the network predictor, V_n, grows with the sample size n. This paradigm attempts to capture the fact that the number of elements in A_i, given by V_n^2 can be substantially larger than sample size. Since model (3.1) is equivalent to model (3.3), the size of the coefficient γ in (3.3) is also a function of n, given by $q_n = \frac{V_n(V_n-1)}{2}$. This creates theoretical challenges, related to (but distinct from) those faced in showing posterior consistency for high dimensional continuous [5] and binary regressions [143].

Let $y_n = (y_1, ..., y_n)'$. Using the superscript (0) to indicate true parameters, the true data generating model is given by

$$y_i \sim Bernoulli\left[\frac{\exp(\psi_i^{(0)})}{1+\exp(\psi_i^{(0)})}\right], \ \psi_i^{(0)} = \langle A_i, \Gamma^{(0)}\rangle_F. \tag{3.5}$$

where $\Gamma^{(0)}$ is the true network coefficient. Let $\gamma^{(0)}$ be the vectorized upper triangular part of $\Gamma^{(0)}$. We assume, $\gamma_{k,l}^{(0)} = u_k^{(0)'}\Lambda u_l^{(0)} + \gamma_{2,k,l}^{(0)}$, where $u_k^{(0)}$ is a R_0 dimensional vector, $k = 1, ..., V$. $\gamma_2^{(0)}$ is the vector of all $\gamma_{2,k,l}^{(0)}$, $k < l$, and we denote the number of nonzero elements of $\gamma_2^{(0)}$ by $s_{2,n}^0$, i.e. $||\gamma_2^{(0)}||_0 = s_{2,n}^0$.

For any $\varepsilon > 0$, define $\mathcal{A}_n = \left\{\gamma : \frac{1}{n}\sum_{i=1}^{n} |f_\gamma(x_i) - f_{\gamma^{(0)}}(x_i)| \le \varepsilon\right\}$ as a neighborhood around the true density. Further suppose $\pi_n(\cdot)$ and $\Pi_n(\cdot)$ are the prior and posterior densities of γ with n observations, so that

$$\Pi_n(\mathcal{A}_n^c) = \frac{\int_{\mathcal{A}_n^c} p_\gamma(y_n)\pi_n(\gamma)}{\int p_\gamma(y_n)\pi_n(\gamma)},$$

where $p_\gamma(y_n)$ denotes the likelihood of the $n$dimensional response vector y_n.

3.3.1 Main Results

To show the posterior contraction results, we follow [143] and [5], with substantial modifications required due to the nature of our proposed network lasso prior distribution. In proving the results, we make a couple of simplifications. It is assumed that the dimension R of u_k is fixed and is the same as R_0, the dimension of $u_k^{(0)}$. Consequently, *effective dimensionality* is not required to be estimated, and hence $\Lambda = I$ is a non-random matrix. Additionally, we assume M to be non-random and $M = I$. We emphasize that both these assumptions are *not* essential for the posterior contraction rate result to be true, and are only introduced for simplifying calculations.

For two sequences $\{C_{1,n}\}_{n\geq 1}$ and $\{C_{2,n}\}_{n\geq 1}$, $C_{1,n} = o(C_{2,n})$ if $C_{1,n}/C_{2,n} \to 0$, as $n \to \infty$. To begin with, we state the following assumptions under which posterior contraction will be shown.

(A) $\sup_{r=1,..,R;k=1,..,V_n} |u_{k,r}^{(0)}| < \infty$;

(B) $V_n = o(\frac{n}{\log(n)})$;

(C) $||A_i||_\infty$ is bounded for all $i = 1,..,$, w.l.o.g assume $||A_i||_\infty \leq 1$.

(D) $s_{2,n}^0 \log(q_n) = o(n)$

(E) $||\gamma_2^{(0)}||_\infty < \infty$;

(F) $\theta_n = \frac{C}{q_n n^{\rho/2}\log(n)}$ for some $C > 0$ and some $\rho \in (1,2)$.n

Remark: Conditions (A), (C) and (E) are technical conditions ensuring that each of the entries in the true network coefficient and the network predictor are bounded. Condition (B) puts an upper bound on the growth of the number of network nodes with sample size to achieve asymptotically optimal classification. Similarly, (D) puts a restriction on the number of nonzero elements of $\gamma_2^{(0)}$ with respect to n.

The following theorem shows contraction of the posterior asymptotically under mild sufficient conditions on $V_n, s_{2,n}^0$. The proof of the theorem is provided in Appendix F.

Theorem 3.3.1 *Under assumptions (A)-(F) for the Bayesian Network Lasso prior on* γ*,* $\Pi_n(\mathcal{A}_n) \to 0$ *in* $P_{\gamma^{(0)}}$ *as* $n \to \infty$*, for any* $\varepsilon > 0$*.*

3.4 Posterior Computation

We have implemented both the Bayesian Network Lasso and Network Horseshoe shrinkage priors on γ. Using the result in [111], the data augmented representation of the distribution of y_i given in (3.2) follows as below

$$p(y_i|\omega_i) = 2^{-b}\exp(k_i\psi_i)\exp(-\omega_i\psi_i^2/2), \ \omega_i \sim PG(1,0), \tag{3.6}$$

where $k_i = y_i - 1/2$. Let $x_i = (a_{i,1,2}, a_{i,1,3}, ..., a_{i,1,V}, a_{i,2,3}, a_{i,2,4}, ..., a_{i,2,V}, ..., a_{i,V-1,V})'$ be of dimension $q \times 1$, where $q = \frac{V(V-1)}{2}$. Assume $X = (x_1 : \cdots : x_n)'$ is an $n \times q$ matrix. Then the conditional likelihood of $y = (y_1, ..., y_n)'$ given $\omega = (\omega_1, ..., \omega_n)'$ and γ is given by

$$\begin{aligned} p(y\,|\,X,\gamma,\omega) &\propto \prod_{i=1}^{n} p(y_i\,|\,x_i,\gamma,\omega_i,...) \\ &\propto \prod_{i=1}^{n} \exp\left\{(y_i - 0.5)(\mu + x_i'\gamma) - \omega_i(\mu + x_i'\gamma)^2/2\right\} \\ &\propto \prod_{i=1}^{n} \exp\left\{-\frac{\omega_i}{2}\left[\frac{(y_i - 0.5)}{\omega_i} - (\mu + x_i'\gamma)\right]^2\right\} \end{aligned}$$

In matrix notation, the likelihood may be written as

$$p(y\,|\,X,\gamma,\omega...) \propto N(t\,|\,\mu 1 + X\gamma, \Omega^{-1})$$

where $t = ((y_1 - 0.5)/\omega_1, ..., (y_n - 0.5)/\omega_n)' = (k_1/\omega_1, ..., k_n/\omega_n)'$ and $\Omega = diag(\omega_1, ..., \omega_n)$. While the full posterior distributions for the parameters are not in closed forms, they mostly belong to the standard families. Hence drawing posterior samples using MCMC can be readily implemented. Appendix D and Appendix E describe full conditional distributions of parameters for Bayesian Network Lasso and Network Horseshoe priors on γ, respectively.

Let $\Omega^{(1)},...,\Omega^{(L)}$, $\Gamma^{(1)},...,\Gamma^{(L)}$ and $\mu^{(1)},...,\mu^{(L)}$ be the L post burn-in MCMC samples for Ω, Γ and μ respectively after suitable thinning. To classify a newly observed network M_* as a member of one of the two groups, we compute $S^{(l)} = \frac{\exp(\mu^{(1)}+\langle M_*,\Gamma^{(l)}\rangle)}{1+\exp(\mu^{(1)}+\langle M_*,\Gamma^{(l)}\rangle)}$ for $l = 1,...,L$. M_* is classified as a member of group 'low' or 'high' if $\frac{1}{L}\sum_{l=1}^{L} S^{(l)}$ is less than or greater than 0.5, respectively. To judge sensitivity to the choice of the cut-off, the simulation section presents Area under Curve (AUC) of ROC curves with True Positive Rates (TPR) and False Positive Rates (FPR) of classification corresponding to a range of cut-off values.

Node k is recognized to be influential in the classification process if $\frac{1}{L}\sum_{l=1}^{L}\xi_k^{(l)} > 0.5$, where $\xi_k^{(1)},...,\xi_k^{(L)}$ are the L post burn-in MCMC samples of ξ_k. Again, one of the goals of the proposed framework is to identify influential network edges impacting the response. We employ the algorithm described in Appendix C to identify influential edges. The algorithm takes care of multiplicity correction by controlling the false discovery rate (FDR) at 5% level. Finally, we present an estimate of $P(R_{eff} = r\,|\,Data)$ computed by $\frac{1}{L}\sum_{l=1}^{L} I(\sum_{m=1}^{R}\lambda_m^{(l)} = r)$, where $I(A)$ for an event A is 1 if the event A happens and 0 otherwise, and $\lambda_m^{(1)},...,\lambda_m^{(L)}$ are the L post burn-in MCMC samples of λ_m.

3.5 Simulation Studies

This section evaluates the inferential and classification ability of our proposed Bayesian network classification (BNC) framework, along with a number of competitors, using synthetic networks generated under various simulation settings. Our proposed network classification approach with the Bayesian Network Lasso prior and the Bayesian Network Horseshoe prior are referred to as the Bayesian Network Lasso classifier (BNLC) and Bayesian Network Horseshoe classifier (BNHC), respectively. In each simulation, we assess the ability of the BNLC and BNHC approaches to correctly identify influential nodes and edges, to accurately estimate predictive edge coefficients and to classify a network with precise characterization of uncertainties. Classification performance of both methods are assessed using the area under the Receiving Operating Characteristics (ROC) curve (AUC).

To study all competitors under various data generation schemes, we simulate the response from (3.1) given by

$$y_i \sim Ber\left(\frac{\exp(\mu_0 + \langle A_i, \Gamma_0 \rangle_F)}{1+\exp(\mu_0 + \langle A_i, \Gamma_0 \rangle_F)}\right), \tag{3.7}$$

where Γ_0 is a symmetric matrix with zero diagonal entries. The intercept μ_0 is fixed at 2 in all simulation scenarios. We consider two different schemes of generating the network A_i, referred to as *Simulation 1* and *Simulation 2*, respectively.

Simulation 1. In *Simulation 1*, the network edges (i.e., the elements of the matrix A_i) are simulated from $\mathrm{N}(0,1)$. Thus, *Simulation 1* assumes that the network predictor follows an Erdos-Renyi graph.

Simulation 2. In *Simulation 2*, the network predictor A_i corresponding to the ith sample is generated from a stochastic blockmodel. Here nodes in a simulated network are organized into communities so that nodes in the same community tend to have stronger connections than nodes belonging to different communities. This simulation scenario simulates networks which closely mimic brain connectome networks [11]. To simulate networks with such community structures, we assign each node a community label, $f_k \in \{1,2,...,3\}$, $k = 1,...,V$. The node assignments are the same for all networks in the population. Given the community labels, the (k,k')th element of A is simulated from $N(m_{f_k,f_{k'}}, \sigma_0^2)$, where $m_{k,l} = 0.5$ when $k = l$. When $k \neq l$, i.e., the concerned edges connect nodes belonging to different clusters, we sample a fixed number of edge locations randomly and simulate the values from $N(0,1)$, assigning the values at the remaining locations to be 0. We set $\sigma_0^2 = 1$ and the three clusters with 8, 9 and 8 nodes respectively, in the three communities. We note that the network predictors are simulated from a stochastic blockmodel in *Simulation 2* which also ensures transitivity in the network predictor.

Simulating the network predictor coefficient Γ_0. In both Simulations 1 and 2, the network predictor coefficient Γ_0 is constructed as the sum of two matrices $\Gamma_{0,1}$ and $\Gamma_{0,2}$. We provide the details of constructing the two matrices as below.

In both Simulations 1 and 2, we draw V latent variables $u_{k,0}$, each of dimension R_g, from a mixture distribution given by

$$u_{k,0} \sim \pi N_{R_g}(u_{m,g}, u_{s,g}^2) + (1-\pi)\delta_0; \, k \in \{1,...,V\}, \tag{3.8}$$

where δ_0 is the Dirac-delta function and π is the probability of any $u_{k,0}$ being nonzero. Define a symmetric matrix $\Gamma_{0,1}$ whose (k,l)th element is given by $\frac{u'_{k,0}u_{l,0}}{2}$, $k < l$ and $= 0$ if $k = l$. Note that if $u_{k,0}$ is zero, then the kth node has no contribution to the mean function in (4.8), i.e., the kth node becomes non-influential in predicting the response. Since $(1-\pi)$ is the probability of a node being inactive, it is referred to as the *node sparsity* parameter in the context of the data generation mechanism under *Simulations 1* and *2*. All elements of $u_{m,g}$ are taken to be 0.5 and $u_{s,g}$ is taken to be 1.

We also construct another symmetric sparse matrix $\Gamma_{0,2}$ to add additional edge effects corresponding to edges connecting a few randomly selected nodes. Let π_2 be the proportion of nonzero elements of $\Gamma_{0,2}$, set randomly at either 0.05 or 0.1. We randomly choose π_2 proportion of locations from the set of all (k,l). The nonzero entries are drawn using one of the three following strategies:

Strategy 1: Nonzero entries are simulated from $\mathrm{N}(1,0.1)$.

Strategy 2: Nonzero entries are simulated from $\mathrm{N}(0.5,0.1)$.

Strategy 3: All nonzero entries are fixed at 0.5.

The quantity $(1-\pi_2)$ is referred to as the *residual edge sparsity*.

Note that the specification of true edge coefficients largely preserves the transitivity property in Γ_0. To see this, note that $\Gamma_{0,2}$ is highly sparse, so that $\gamma_{0,1,k,l} = \gamma_{0,k,l}$ for most pairs (k,l), $k<l$. For those pairs, $\gamma_{0,k,l} \neq 0$ and $\gamma_{0,l,l'} \neq 0$ imply that $u_{k,0} \neq 0$, $u_{l,0} \neq 0$ and $u_{l',0} \neq 0$. Thus it follows that $\gamma_{0,k,l'} = \frac{u'_{k,0}u_{l',0}}{2} \neq 0$.

For a comprehensive picture of *Simulation 1* and *Simulation 2*, we consider 4 different cases each in both simulations as summarized in Table 3.1 and 3.2 respectively. In each of these cases, the network predictor coefficient and the response are generated by changing the node sparsity $(1-\pi)$, the residual edge sparsity $(1-\pi_2)$ and the true dimension R_g of the latent variables $u_{k,0}$'s. The table also presents the maximum fitted dimension R of the latent variables u_k for the logistic regression model (3.2). Note that the various cases also allow model misspecification with unequal choices of R and R_g.

Cases	R_g	R	Node Sparsity $(1-\pi)$	Residual Edge Sparsity $(1-\pi_2)$	Strategy
Case - 1	2	2	0.5	0.95	Strategy 1
Case - 2	3	5	0.6	0.95	Strategy 1
Case - 3	2	5	0.5	0.90	Strategy 2
Case - 4	2	5	0.4	0.90	Strategy 3

Table 3.1: Table presents different cases for ***Simulation 1***. The true dimension R_g is the dimension of vector object $u_{k,0}$ using which data has been generated. The maximum dimension R is the dimension of vector object u_k using which the model has been fitted. Node sparsity and residual edge sparsity are described in the text.

Cases	R_g	R	Node Sparsity $(1-\pi)$	Residual Edge Sparsity $(1-\pi_2)$	Strategy
Case - 1	2	2	0.5	0.95	Strategy 1
Case - 2	2	4	0.5	0.95	Strategy 1
Case - 3	2	3	0.7	0.95	Strategy 1
Case - 4	2	5	0.4	0.90	Strategy 3

Table 3.2: Table presents different cases for ***Simulation 2***. The true dimension R_g is the dimension of vector object $u_{k,0}$ using which data has been generated. The maximum dimension R is the dimension of vector object u_k using which the model has been fitted. Node sparsity and residual edge sparsity are described in the text.

As competitors, we use generic variable selection and shrinkage methods that treat edges between nodes together as a long predictor vector to run high dimensional regression, thereby ignoring the relational nature of the predictor. More specifically, we use Lasso [133], which is a popular penalized optimization scheme, and the Bayesian Lasso (BLasso for short)[109] and Bayesian Horseshoe (BHS for short) priors [17], which are popular Bayesian shrinkage regression methods, all three under the logistic regression framework. We use the `glmnet` package in R [50] to implement the frequentist Lasso, while we write our own codes for BLasso and BHS. A comparison with these methods will indicate any relative advantage of exploiting the structure of the network predictor. Additionally, we compare our methods to a frequentist approach that develops network classification in the presence of a network predictor and a binary response [113]. We refer to this approach as *Relión.*

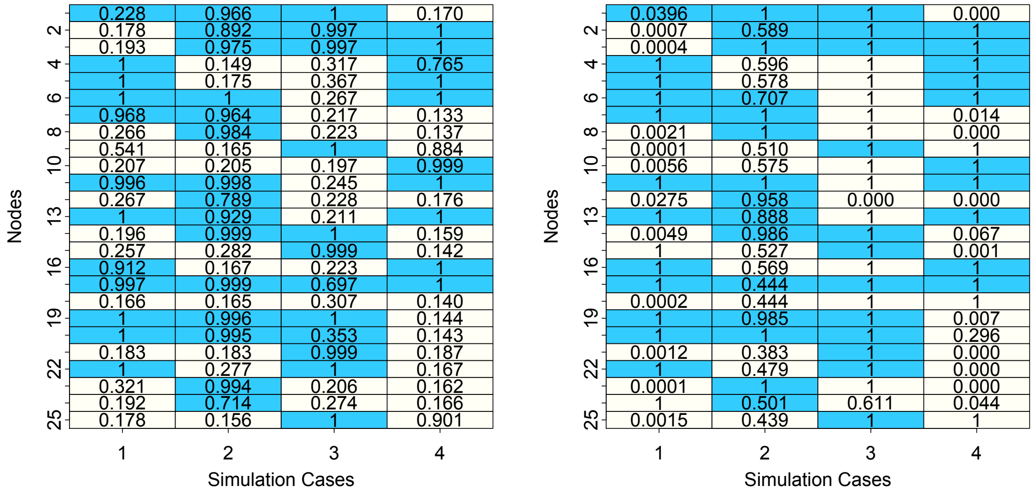
All Bayesian competitors are allowed to draw $50,000$ MCMC samples, out of which the first $30,000$ are discarded as burn-ins. Convergence is assessed by comparing different simulated sequences of representative parameters starting at different initial values [53]. All posterior inference is carried out based on the rest $20,000$ MCMC samples after suitably thinning the post burn-in chain. We monitor the auto-correlation plots and effective sample sizes of the iterates, and they are found to be satisfactorily uncorrelated. In all of our simulations, we set $V = 25$ nodes and $n = 250$ samples.

We present analysis for $\nu = 20$, $a_\Delta = b_\Delta = 1$. For BNLC, there are two additional hyper-parameters ι and ζ, both of which are set to 1. Note that the choice of $a_\Delta = b_\Delta = 1$ ensures that the prior on models is such that we have a uniform distribution on the number of active nodes, and conditional on the size of the model, a uniform distribution on all possible models of that size. The choice of $\nu = 20$ ensures that the prior distribution of M is concentrated around a scaled identity matrix. Since model is invariant to rotations of the latent positions, so we want the prior on u_k's to also be invariant under rotation. That requires that we center M around a matrix that is proportional to the identity. Our choice of ι and ζ set the prior mean of $s_{k,l}$ at 0.5 which is the suggested prior mean for the local parameters proposed in [109]. Sensitivity to the choice of hyper-parameters is discussed later, both for simulation studies and for the real data analysis.

3.5.1 Identification of Influential Nodes

Figures 3.1 and 3.2 show the posterior probability of the k-th node being detected as influential, i.e., $P(\xi_k = 1|Data)$, by BNLC and BNHC for each node and each case within *Simulations 1* and *2*, respectively. Some interesting observations emerge from the results. We find that both methods work well with lower node sparsity and higher residual edge sparsity. Decreasing the residual edge sparsity and increasing the node sparsity have adverse effects on the performance. In general, BNLC shows relatively better performance than BNHC in cases with higher node sparsity and/or lower residual edge sparsity. We provide a brief discussion below to support these observations.

For BNHC, case 2 exhibits a few false positives, and the separation of posterior probabilities for truly active and truly inactive nodes is much more stark in case 1 than in case 2. BNLC does a better job of node identification than BNHC in case 2. Residual edge effect does have an impact on the probabilities, which is evident by comparing cases 1 and 3. For BNHC, case 3 (Simulation 1) displays poor performance with a higher number of both false positives and false negatives. Performance of BNLC appears to be better than BNHC in case 3. Fixing the residual edge sparsity and increasing the node sparsity has a negative impact on node identification, as seen by comparing performances in cases 3 and 4 (Simulation 1). For Simulation 2, both competitors perform quite well in cases 1 a nd 2. Again, case 3 (Simulation 2) represents a higher node sparsity, so that both BNHC and BNLC do not perform well in this case. Similar to Simulation 1, BNHC shows inferior performance to BNLC in case 3. While BNHC offers a few false positives and false negatives in case 4 (Simulation 2), the performance appears to be much better than in case 3. Notice that case 3 has both higher node sparsity and residual edge sparsity than case 4. While they have opposing effects, it appears that higher node sparsity demonstrates more of an adverse effect here compared to a small perturbation in the residual edge sparsity. Recall that [113] is the only other competitor which is designed to detect influential nodes. It detects all nodes to be influential in all simulation cases.



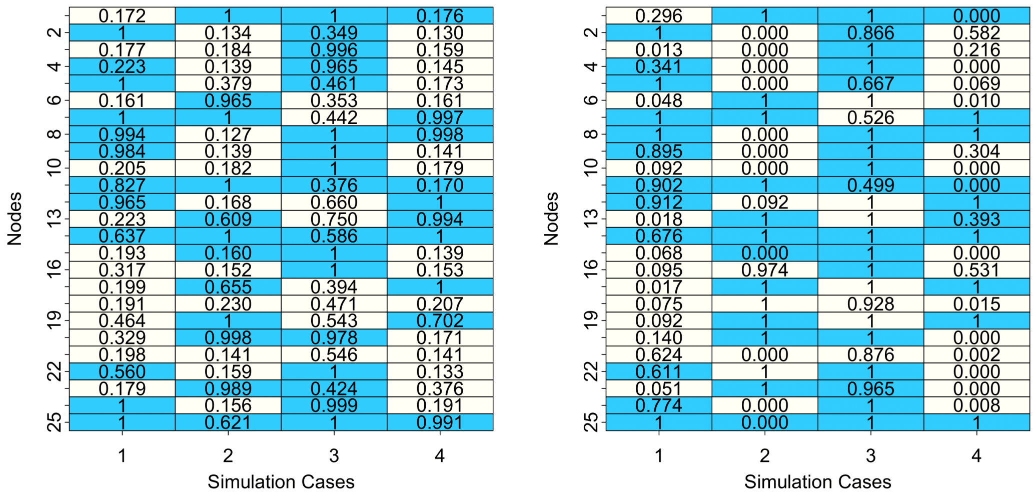
Nodes	1	2	3	4
	0.228	0.966	1	0.170
2	0.178	0.892	0.997	1
	0.193	0.975	0.997	1
4	1	0.149	0.317	0.765
	1	0.175	0.367	1
6	1	1	0.267	1
	0.968	0.964	0.217	0.133
8	0.266	0.984	0.223	0.137
	0.541	0.165	1	0.884
10	0.207	0.205	0.197	0.999
	0.996	0.998	0.245	1
	0.267	0.789	0.228	0.176
13	1	0.929	0.211	1
	0.196	0.999	1	0.159
	0.257	0.282	0.999	0.142
16	0.912	0.167	0.223	1
	0.997	0.999	0.697	1
	0.166	0.165	0.307	0.140
19	1	0.996	1	0.144
	1	0.995	0.353	0.143
	0.183	0.183	0.999	0.187
22	1	0.277	1	0.167
	0.321	0.994	0.206	0.162
	0.192	0.714	0.274	0.166
25	0.178	0.156	1	0.901

(a) BNLC

Nodes	1	2	3	4
	0.0396	1	1	0.000
2	0.0007	0.589	1	1
	0.0004	1	1	1
4	1	0.596	1	1
	1	0.578	1	1
6	1	0.707	1	1
	1	1	1	0.014
8	0.0021	1	1	0.000
	0.0001	0.510	1	1
10	0.0056	0.575	1	1
	1	1	1	1
	0.0275	0.958	0.000	0.000
13	1	0.888	1	1
	0.0049	0.986	1	0.067
	1	0.527	1	0.001
16	1	0.569	1	1
	1	0.444	1	1
	0.0002	0.444	1	1
19	1	0.985	1	0.007
	1	1	1	0.296
	0.0012	0.383	1	0.000
22	1	0.479	1	0.000
	0.0001	1	1	0.000
	1	0.501	0.611	0.044
25	0.0015	0.439	1	1

(b) BNHC

Figure 3.1: Simulation 1: clear background denotes *uninfluential* and dark background denotes *influential* nodes in the truth for BNLC and BNHC models. Note that there are 25 rows (corresponding to 25 nodes) and 4 columns corresponding to 4 different cases in Simulation 1. The model-detected posterior probability of being influential has been super-imposed onto the corresponding node.



(a) BNLC

(b) BNHC

Figure 3.2: Simulation 2: clear background denotes *uninfluential* and dark background denotes *influential* nodes in the truth for BNLC and BNHC models. Note that there are 25 rows (corresponding to 25 nodes) and 4 columns corresponding to 4 different cases in Simulation 2. The model-detected posterior probability of being influential has been super-imposed onto the corresponding node.

3.5.2 Identification of Influential Edges

We apply the algorithm with a mixture of skewed t-distributions described in Appendix C to detect influential edges from the post burn-in MCMC samples of the edge coefficients using a threshold of $t = 0.05$. The proposed approach controls FDR below a threshold of 0.05 to account for multiplicity correction. Tables 3.3 and 3.4 provide the true positive rates (TPR) and false positive rates (FPR) in detecting important edges for Simulations 1 and 2 for the competitors, respectively. It is observed that when node sparsity is moderate and residual edge sparsity is high (cases 1 and 2), both BNLC and BNHC offer moderate performance in terms of identifying true positives, and include very few false positives. In these cases, BNHC generally exhibits a little higher FPR than BNLC. In the case of high node sparsity (e.g., case 3, Simulation 2) both these methods unfortunately show much lower true positive rates. Again, lower edge sparsity (case 3, Simulation 1) has almost no effect on FPR of BNLC, but decreases TPR substantially. For BNHC, both TPR and FPR increase when residual edge sparsity is reduced. Nevertheless, both of them perform significantly better than Lasso in almost all cases. The competitor in [113] appears to have suboptimal performance, as it identifies all edges as important in all the simulation scenarios, resulting in high FPRs.

	BNLC		BNHC		Lasso		Relión (2017)	
Cases	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
Case - 1	0.65	0.01	0.72	0.12	0.50	0.22	1	1
Case - 2	0.64	0.00	0.63	0.02	0.40	0.14	1	1
Case - 3	0.45	0.00	0.86	0.40	0.42	0.22	1	1
Case - 4	0.72	0.09	0.70	0.12	0.54	0.16	1	1

Table 3.3: True Positive Rates (TPR) and False Positive Rates (FPR) for edges for cases in *Simulation 1*.

	BNLC		BNHC		Lasso		Relión(2017)	
Cases	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
Case - 1	0.63	0.00	0.84	0.08	0.44	0.20	1	1
Case - 2	0.56	0.00	0.63	0.12	0.53	0.22	1	1
Case - 3	0.46	0.02	0.59	0.08	0.31	0.16	1	1
Case - 4	0.68	0.03	0.75	0.06	0.34	0.12	1	1

Table 3.4: True Positive Rates (TPR) and False Positive Rates (FPR) for edges for cases in *Simulation 2*.

The results in Tables 3.3 and 3.4 indicate higher number of edges identified as influential by BNHC than BNLC in all simulations. Digging a bit deeper, we report the ratio of the number of edges in the intersection of both methods to the number of total edges identified by each method independently in Table 3.5. In all simulation cases, almost all edges identified as influential by BNLC are also identified as influential by BNHC. In cases 2 and 4 (Simulation 1), the fractions corresponding to BNLC and BNHC are very similar, indicating similar edge identification by both of them. However, this fraction appears to be lower in BNHC for cases 1 and 3 (Simulation 1). This again shows that the edges identified by BNLC are also identified by BNHC, with BNHC identifying more edges. The discrepancy turns out to be more in case 3 (Simulation 1) where BNHC has identified many more edges. Simulation 2 shows a similar trend. We further track the top 10, 20 and 30 edges identified from BNLC and record how many of these edges belong to the top 10, 20 and 30 edges identified from BNHC. Table 3.5 shows a high level of intersection among the top edges identified by these two methods.

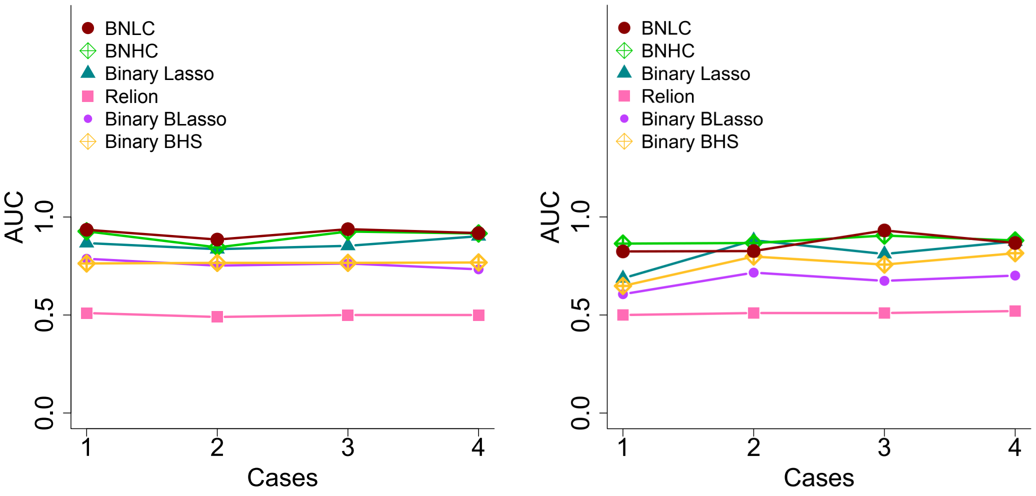
A number of interesting observations emerge from the analysis. First of all, as mentioned earlier, the edges identified by BNLC are generally also identified by BNHC. BNHC tends to identify more edges, leading to higher TPR and FPR. Broadly, in presence of higher node sparsity, the discrepancy is greater, with BNHC having much higher TPR and FPR. Interestingly, the absolute values of the edge coefficients follow very similar rankings for BNHC and BNLC, which leads to high intersections among the top edges selected by these methods. Perhaps the difference in shrinkage mechanism imposed by BNHC and BNLC is responsible for their difference in tail behavior, leading to differences in edge selection.

	Simulation 1					Simulation 2				
Cases	$\frac{N_{BL,BH}}{N_{BL}}$	$\frac{N_{BL,BH}}{N_{BH}}$	Top			$\frac{N_{BL,BH}}{N_{BL}}$	$\frac{N_{BL,BH}}{N_{BH}}$	Top		
			10	20	30			10	20	30
1	0.94	0.61	9	19	27	1.00	0.58	7	17	26
2	0.85	0.83	8	14	21	1.00	0.46	8	17	26
3	1.00	0.25	9	13	24	0.97	0.70	9	18	28
4	0.91	0.87	8	18	27	0.91	0.75	8	17	27

Table 3.5: $N_{BL,BH}$ represents the number of edges identified by both BNLC and BNHC. Similarly, N_{BL} and N_{BH} represent the number of edges identified by BNLC and BNHC, respectively. Top 10 represents the number of edges common among the top ten edges identified by BNLC and BNHC. Top 20 and Top30 are defined analogously.

3.5.3 Estimation of Edge Coefficients and Classification Accuracy

The mean squared errors (MSE) associated with the point estimation of edge coefficients for different competitors are presented in Tables 3.6 and 3.7, corresponding to Simulations 1 and 2, respectively. For the Bayesian competitors, point estimates are computed using the posterior means of the edge coefficients. In all cases, BNLC and BNHC consistently outperform all other competitors, with the binary Bayesian Lasso exhibiting the next best performance. In all simulation cases, BNLC comprehensively outperforms BNHC in terms of estimating edge coefficients. Consistent with earlier observations, both competitors tend to be less accurate when node sparsity increases. Figure 3.3 records AUC for all competitors in Simulations 1 and 2. In almost all cases, AUC for BNHC and BNLC turn out to be higher than other competitors. On the other hand, [113] appears to have close to random classification of samples with AUC around 0.5.



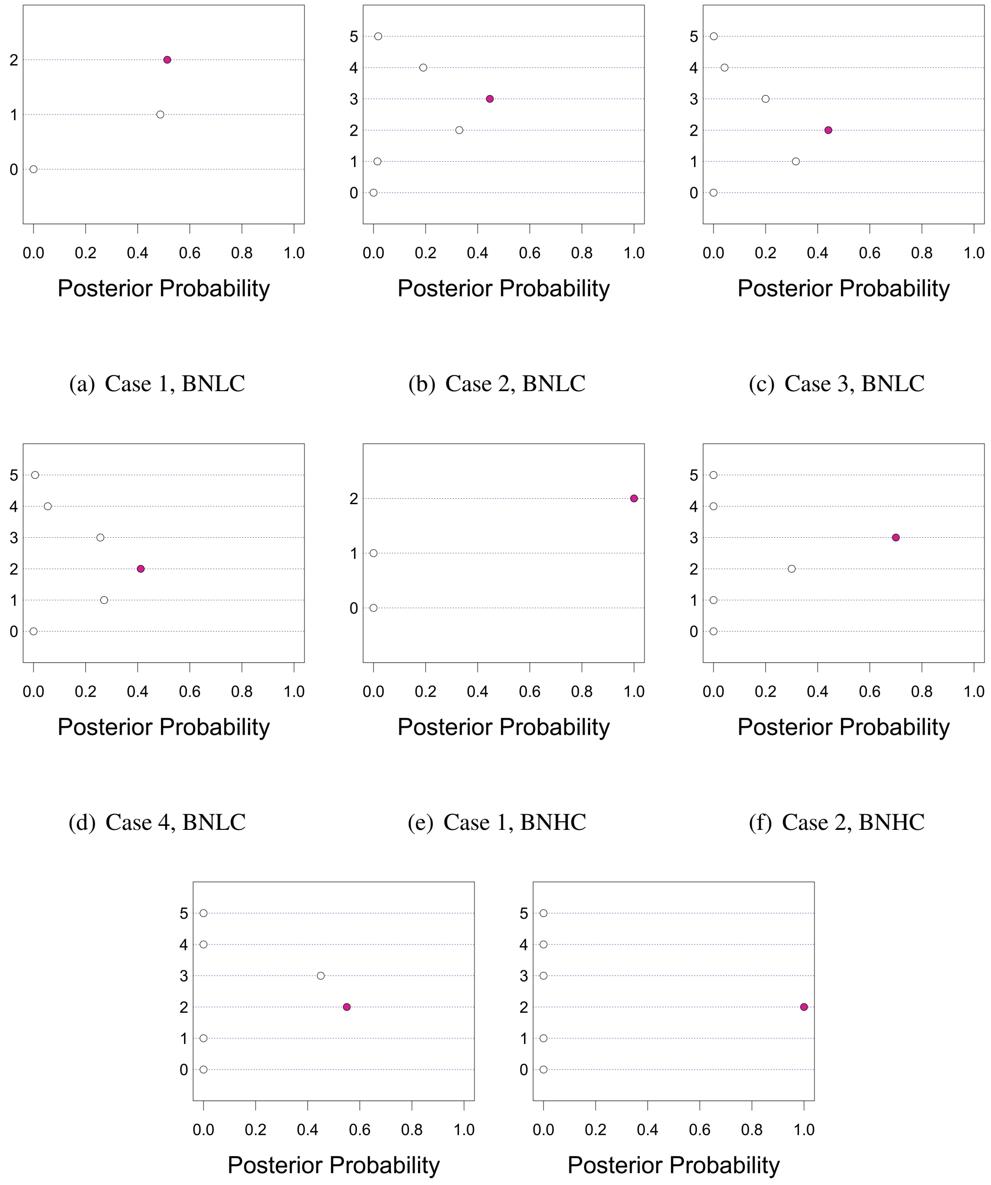
(a) Simulation 1

(b) Simulation 2

Figure 3.3: Figure shows classification performance in the form of Area under Curve (AUC) of ROC for all cases in Simulations 1 and 2.

	MSE					
Cases	BNLC	BNHC	Lasso	Relión(2017)	Binary	Binary
					BL	Horseshoe
Case - 1	**0.164**	0.683	1.197	1.387	0.980	1.160
Case - 2	**2.349**	3.568	3.943	4.368	3.502	3.993
Case - 3	**0.106**	0.467	0.906	1.056	0.695	0.856
Case - 4	**0.166**	0.200	0.485	0.617	0.329	0.415

Table 3.6: Performance of BNLC and BNHC vis-a-vis competitors for cases in Simulation 1. Parametric inference in terms of point estimation of edge coefficients has been captured through the Mean Squared Error (MSE). The minimum MSE among competitors for any case is made bold.

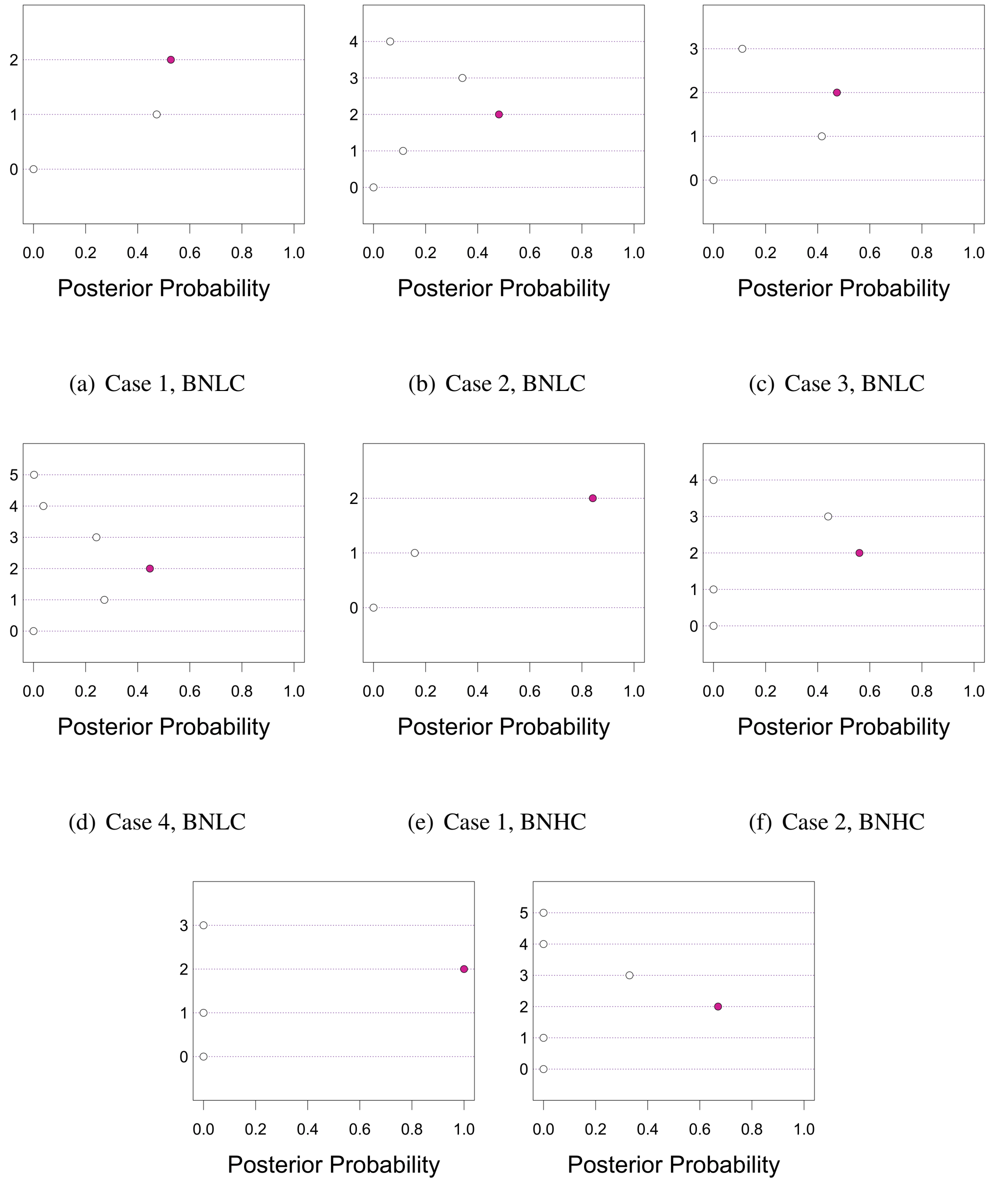


(a) Case 1, BNLC (b) Case 2, BNLC (c) Case 3, BNLC

(d) Case 4, BNLC (e) Case 1, BNHC (f) Case 2, BNHC

(g) Case 3, BNHC (h) Case 4, BNHC

Figure 3.4: Plots showing posterior probability distribution of effective dimensionality for BNLC and BNHC models in all 4 cases in Simulation 1. Filled bullets indicate the true value of effective dimensionality.



(a) Case 1, BNLC

(b) Case 2, BNLC

(c) Case 3, BNLC

(d) Case 4, BNLC

(e) Case 1, BNHC

(f) Case 2, BNHC

(g) Case 3, BNHC

(h) Case 4, BNHC

Figure 3.5: Plots showing posterior probability distribution of effective dimensionality for BNLC and BNHC models in all 4 cases in Simulation 2. Filled bullets indicate the true value of effective dimensionality.

3.5.4 Estimation of Effective Dimensionality

Figures 3.4 and 3.5 present posterior probabilities of effective dimensionality of the latent positions $u_1,\ldots,u_V$ for BNLC and BNHC in Simulations 1 and 2, respectively. Note that the true dimension of the latent space is known and recorded for all simulations in Tables 3.1 and 3.2. In all 8 cases, the posterior mode corresponds to the true dimension of the latent space for both BNLC and BNHC. Compared to BNLC, the posterior distribution of R_{eff} in BNHC concentrates more sharply around R_g in all cases.

	MSE					
Cases	BNLC	BNHC	Lasso	Relión(2017)	Binary BL	Binary Horseshoe
Case - 1	**0.279**	0.418	0.807	0.939	0.712	0.739
Case - 2	**0.180**	0.388	0.514	0.665	0.423	0.548
Case - 3	**0.134**	0.549	0.906	1.097	0.748	0.883
Case - 4	**0.066**	0.106	0.167	0.221	0.137	0.141

Table 3.7: Performance of BNLC and BNHC vis-a-vis competitors for cases in Simulation 2. Parametric inference in terms of point estimation of edge coefficients has been captured through the Mean Squared Error (MSE). The minimum MSE among competitors for any case is made bold.

3.5.5 Sensitivity to the choice of Hyperparameters

To assess how sensitive the inferences from BNLC and BNHC are, we analyze BNLC and BNHC with different combinations of hyperparameters. Specifically for BNLC, we use the five different combinations given by, (i) $a_\Delta = 1, b_\Delta = 9$; (ii) $\nu = 20, \delta = 5$ (iii) $\nu = 50, \delta = 5$

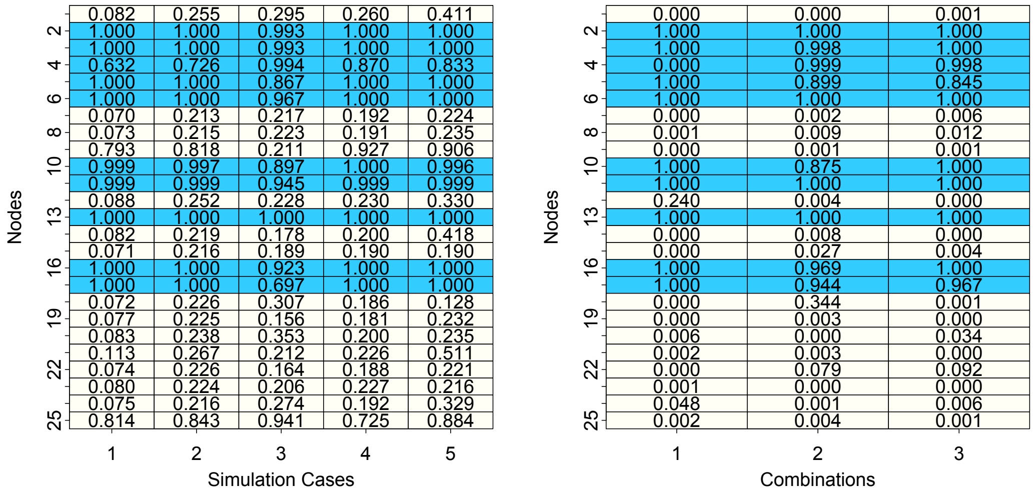
(iv) $\nu = 20, \delta = 0.2$ (v) $\nu = 50, \delta = 0.2$. Combination (i) ensures small prior mean for ξ_k's, while combinations (ii)-(v) allow a range of prior means for θ and M. On the other hand, the three different combinations we employ for BNHC are, (i)' $a = 1$, $b = 9$ (ii)' $\nu = 10$ (iii)' $\nu = 50$. With these hyperparameter combinations for BNLC and BNHC, we analyze the data simulated in case 4, Simulation 1 (case chosen randomly), report performances on influential node and edge identification and the MSE values for estimating the network coefficient matrix. All these inferences with different choices of hyperparameters are compared among themselves and compared with the inferences reported earlier on case 4, Simulation 1.

Table 3.8 records the MSE values for estimating the network coefficient under all these combinations. The MSE values for BNLC range between 0.10 and 0.30 (please see table 3.6). MSE values for BNHC are found to range between 0.19 and 0.28 with different choices of hyperparameters, as shown in able 3.6. Figure 3.6 shows the posterior probabilities of a node being identified as influential under all these hyperparameter combinations. It shows probabilities being only little affected by the change of hyper-parameters. In fact, under hyper-parameter combinations (i),(ii) and (iv), BNLC identifies the same set of nodes as influential which have been identified as influential by the original BNLC prior. Under combination (iii), BNLC does not identify node 9 as influential which has been identified as influential by the original BNLC prior. Under combination (iv) BNLC identifies one additional node (node 21) as influential over the set of nodes identified by the original prior. Under hyperparameter combination (i)', BNHC identifies the same set of nodes with the original BNHC prior except nodes $4, 9, 18, 25$ which are identified as influential by the original prior, but not by the combination (i)'. Combinations (ii)' and (iii)' also identify the same set of nodes with the original BNHC prior except for nodes

9, 18, 25. Finally, Table 3.9 offers TPR and FPR values corresponding to the identification of influential edges for BNLC and BNHC under various combinations of hyper-parameters. The TPR for BNHC under combination (iii)' turns out to be a little higher than the rest, but overall numbers do not show a lot of variation. We emphasize that the results turn out to be better than our competitors under all combinations.

	BNLC					BNHC		
Combinations	(i)	(ii)	(iii)	(iv)	(v)	(i)'	(ii)'	(iii)'
MSE	0.14	0.30	0.22	0.10	0.22	0.19	0.28	0.28

Table 3.8: Mean Squared Error (MSE) of estimating the network coefficient in BNLC and BNHC for different combinations of hyper-parameters.



Nodes	1	2	3	4	5
	0.082	0.255	0.295	0.260	0.411
2	1.000	1.000	0.993	1.000	1.000
	1.000	1.000	0.993	1.000	1.000
4	0.632	0.726	0.994	0.870	0.833
	1.000	1.000	0.867	1.000	1.000
6	1.000	1.000	0.967	1.000	1.000
	0.070	0.213	0.217	0.192	0.224
8	0.073	0.215	0.223	0.191	0.235
	0.793	0.818	0.211	0.927	0.906
10	0.999	0.997	0.897	1.000	0.996
	0.999	0.999	0.945	0.999	0.999
	0.088	0.252	0.228	0.230	0.330
13	1.000	1.000	1.000	1.000	1.000
	0.082	0.219	0.178	0.200	0.418
	0.071	0.216	0.189	0.190	0.190
16	1.000	1.000	0.923	1.000	1.000
	1.000	1.000	0.697	1.000	1.000
	0.072	0.226	0.307	0.186	0.128
19	0.077	0.225	0.156	0.181	0.232
	0.083	0.238	0.353	0.200	0.235
	0.113	0.267	0.212	0.226	0.511
22	0.074	0.226	0.164	0.188	0.221
	0.080	0.224	0.206	0.227	0.216
	0.075	0.216	0.274	0.192	0.329
25	0.814	0.843	0.941	0.725	0.884

(a) BNLC Sensitivity

Nodes	1	2	3
	0.000	0.000	0.001
2	1.000	1.000	1.000
	1.000	0.998	1.000
4	0.000	0.999	0.998
	1.000	0.899	0.845
6	1.000	1.000	1.000
	0.000	0.002	0.006
8	0.001	0.009	0.012
	0.000	0.001	0.001
10	1.000	0.875	1.000
	1.000	1.000	1.000
	0.240	0.004	0.000
13	1.000	1.000	1.000
	0.000	0.008	0.000
	0.000	0.027	0.004
16	1.000	0.969	1.000
	1.000	0.944	0.967
	0.000	0.344	0.001
19	0.000	0.003	0.000
	0.006	0.000	0.034
	0.002	0.003	0.000
22	0.000	0.079	0.092
	0.001	0.000	0.000
	0.048	0.001	0.006
25	0.002	0.004	0.001

(b) BNHC Sensitivity

Figure 3.6: Figure shows $P(\xi_k = 1|Data)$ for BNLC and BNHC under different hyper-parameter combinations in the simulated data for case 4 (Simulation 1).

	BNLC					BNHC		
Combinations	(i)	(ii)	(iii)	(iv)	(v)	(i)’	(ii)’	(iii)’
TPR	0.80	0.76	0.82	0.83	0.78	0.64	0.88	0.82
FPR	0.16	0.21	0.17	0.21	0.18	0.19	0.24	0.18

Table 3.9: True Positive Rates (TPR) and False Positive Rates (FPR) of identifying influential edges in BNLC and BNHC for different combinations of hyper-parameters.

3.6 Brain Connectome Application

In this section, we present the inferential and classification ability of BNLC and BNHC in the context of a weighted diffusion tension imaging (DTI) dataset. Our dataset contains information on the *full scale intelligence quotient* (FSIQ) for multiple individuals. Full scale intelligence quotient (FSIQ) is a measure of an individual’s complete cognitive capacity. It is derived from administration of selected sub-tests from the Wechsler Intelligence Scales (WIS), designed to provide a measure of an individual’s overall level of general cognitive and intellectual functioning, and is a summary score derived from an individual’s performance on a variety of tasks that measure acquired knowledge, verbal reasoning, attention to verbal materials, fluid reasoning, spatial processing, attentiveness to details, and visual-motor integration [15]. A substantial body of literature has suggested that there is an IQ threshold (usually described as an IQ of approximately 120 points) that may be characterized as superior reasoning ability [10, 16]. Following this literature, we have converted the FSIQ scores into a binary response variable y, which takes value 0 if FSIQ is less or equal to 120, and takes value 1 if FSIQ is greater than 120. Thus, we classify the subjects in our study as belonging to the *low IQ* group if $y = 0$, and the *high IQ* group if $y = 1$.

Along with FSIQ measurements, brain connectome information for $n = 114$ subjects is gathered using weighted diffusion tensor imaging (DTI). DTI is a brain imaging technique that enables measurement of the restricted diffusion of water in tissue in order to produce neural tract images. The brain imaging data we use has been pre-processed using the NDMG preprocessing pipeline [82]; [81]; [83]. In the context of DTI, the human brain is divided according to the Desikan atlas [32], which identifies 34 cortical regions of interest (ROIs) both in the left and right hemispheres of the human brain, implying 68 cortical ROIs in all. Similar to Chapter 2, this results in a brain network of a 68×68 matrix for each individual. Our scientific goals in this setting include identification of brain regions or network nodes significantly related to FSIQ and classification of a subject into the low IQ or high IQ group based on his/her brain connectome information.

Identical prior distributions for all the parameters as in the simulation studies have been used. BNLC and BNHC are both fitted with $R = 4$, which is found to be sufficient for this study. Further, Chapter 2 shows robust inference as long as the chosen R is bigger than the effective dimensionality of the latent variables. Similar to Chapter 2, we also do a sensitivity study to check the impact of R on predictive inference. The choice of hyperparameters for BNLC and BNHC are made similar to the simulation studies. A brief explanation for such choices of hyper parameters is provided in the simulation section. The MCMC chain is run for $50,000$ iterations, with the first $30,000$ iterations discarded as burn-in. Convergence is assessed by comparing different simulated sequences of representative parameters started at different initial values [52]. All inference is based on the remaining $20,000$ post burn-in iterates appropriately thinned.

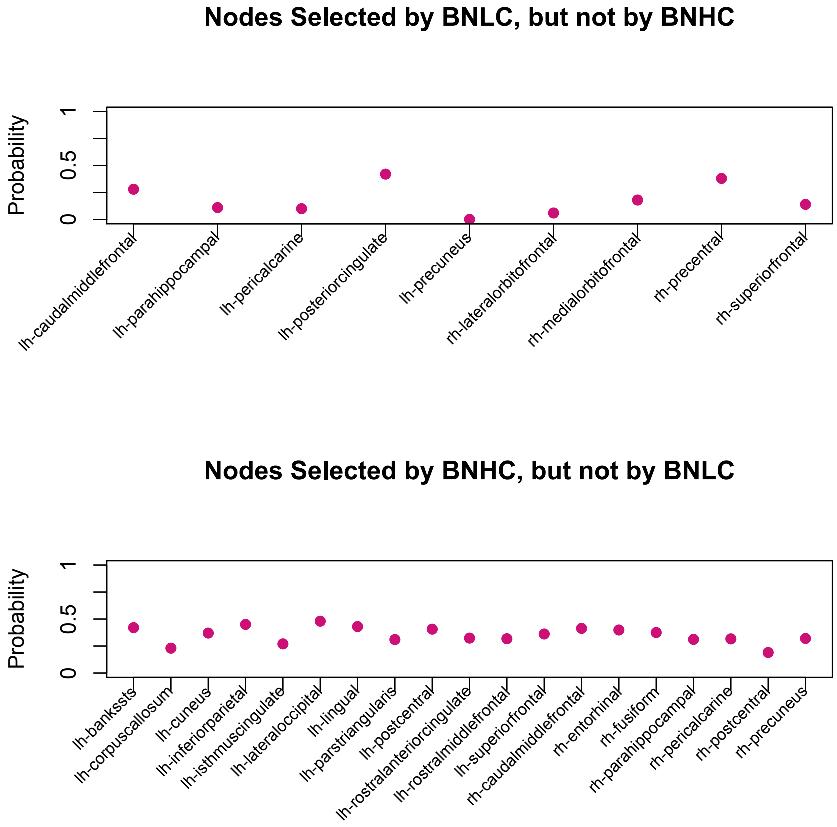
3.6.1 Findings from the Brain Connectome Application

As in simulation studies, we put our emphasis on identifying influential brain regions of interest (ROIs) associated with FSIQ. The BNLC model estimates posterior probabilities over 0.5 (hence detecting as *influential*) for 38 ROIs, out of which 20 regions are in the left hemisphere and 18 regions are in the right hemisphere. Among the regions detected in both the hemispheres, a large number belong to the *frontal*, *temporal* and *cingulate* lobes. Using the same principle, the BNHC model identifies 48 nodes to be influential. Out of the 48 influential nodes, 26 are detected in the left hemisphere and the rest in the right hemisphere. The ROIs are mainly detected in the *temporal, frontal, parietal* and *cingulate* lobes in both hemispheres. Figure 3.8 plots the estimated posterior probability of an ROI being detected as influential by the BNLC and BNHC models. Notably, there are 29 ROIs identified by both BNLC and BNHC, given in Table 3.10.

A large number of the 29 influential nodes detected by both BNLC and BNHC are part of the *frontal* lobes in both the hemispheres. Numerous studies have linked the frontal region to an individual's intelligence and cognitive functions [145, 131, 112, 99, 87]. Our method also finds a significant association between FSIQ and the left *inferior parietal lobule*, the left *precuneus* and the *supramarginal gyri* in both the hemispheres, in the *parietal* lobe, regions also found to be significantly related to FSIQ by [145].

We additionally look into ROIs which are detected by only of the two methods (lets say, BNLC), and report the posterior probabilities of these ROIs being active under the other method (i.e., BNHC). Figure 3.7 shows the posterior probabilities of nodes being active under the 'other' method as discussed above. It is observed that the nodes selected by BNHC but not by BNLC have probabilities not very far from 0.5 under BNLC, which says that BNLC is not enough confident to exclude these nodes from the set of influential nodes. However, most of the nodes selected by BNLC but not by BNHC show smaller probabilities of being influential under BNHC. Perhaps, BNLC is more conservative in including nodes in the set of influential nodes, which is responsible for the discrepancy between the number of identified nodes by BNHC and BNLC.

As described earlier, we identify influential edges connecting pairs of influential nodes using the algorithm described in Appendix C. Figure 3.9 presents the influential edges (among all edges connecting pairs of influential nodes) identified by the BNLC and BNHC models. Note that BNLC and BNHC identify 142 and 291 edges as being influential out of $\binom{38}{2}$ and $\binom{48}{2}$ possibilities, respectively. Since a different number of nodes are detected as influential by BNHC and BNLC, to make a fair comparison, we consider the 29 nodes detected as influential by both methods, and use our algorithm to find the number of influential edges among these $\binom{29}{2}$ possibilities for both BNLC and BNHC. The numbers turn out to be 96 and 184, respectively. We note that there are a few nodes which are identified as influential by either BNHC or BNLC, but none of the edges connecting these nodes are found to be influential. As an example, although the *frontal pole* and the *temporal pole* in the left hemisphere are identified as influential nodes by BNLC, none of the edges connecting these two nodes turn out to be influential. This phenomenon may be due to the use of the FDR in the edge selection procedure, which finds edges that are most likely to be active while controlling for false discoveries. Hence, not



Nodes Selected by BNLC, but not by BNHC

Nodes Selected by BNHC, but not by BNLC

Figure 3.7: Figure shows the posterior probabilities of nodes selected as *influential* by one method, but not by another, of being active.

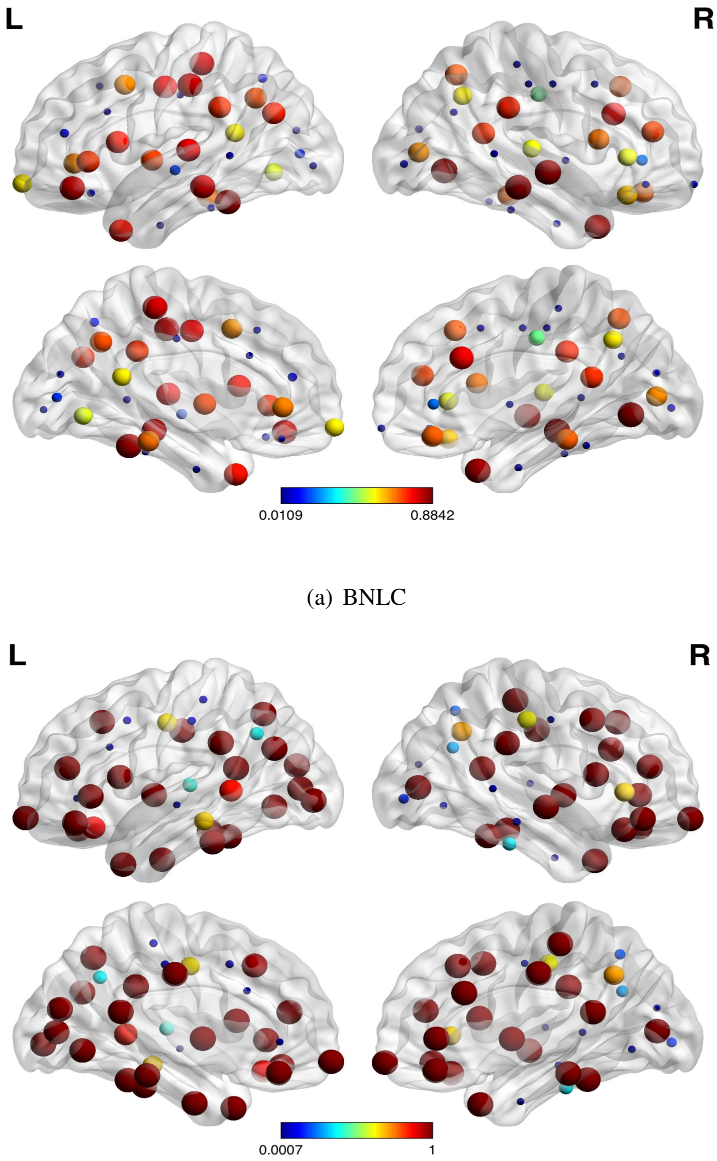
identifying an edge does not necessarily mean that the edge is not active, it just means that there are others that satisfy the criteria better.

Similar to simulation studies, we dig deeper to analyze the discrepancy in the number of influential edges identified by BNLC and BNHC. Specifically, we rank the $\binom{29}{2} = 406$ edges connecting the nodes found to be influential by both BNLC and BNHC, according to the absolute values of their posterior means. Table 3.11 shows between 23-74% intersections.

Hemisphere	Lobe	Node
Left	Temporal	fusiform, middle temporal gyrus, parahippocampal, temporal pole, transverse temporal
	Cingulate	isthmus cingulate cortex
	Frontal	pars opercularis, pars orbitalis, pars triangularis, frontal pole
	Occipital	lingual
	Parietal	inferior parietal lobule, precuneus, supramarginal gyrus
	Insula	insula
Right	Temporal	parahippocampal, superior temporal gyrus, temporal pole
	Cingulate	caudal anterior cingulate, isthmus cingulate cortex
	Frontal	lateral orbitofrontal, medial orbitofrontal, pars opercularis, pars orbitalis, rostral middle frontal gyrus, superior frontal gyrus
	Occipital	pericalcarine
	Parietal	supramarginal gyrus
	Insula	insula

Table 3.10: Nodes identified as influential by both BNLC and BNHC.

To examine the predictive ability of the Bayesian network classification model, we report the area under curve (AUC) of the ROC curve for BNLC and BNHC, along with all competing methods. The AUCs are computed using a 10-fold cross validation approach. The AUC estimates presented in Table 3.12 indicate better performance of both BNLC and BNHC, with BNLC slightly outperforming. Frequentist Binary Lasso turns out to be the next best performer, while BLasso and BHS perform very similar to a random classifier. Finally, the effective dimensionality of the model is investigated for both BNLC and BNHC, and they turn out to be 2.17 and 2, respectively.



(a) BNLC

(b) BNHC

Figure 3.8: Lateral and medial views of the brain (left and right hemispheres) showing all 68 regions of interest (ROIs). The size and color of the ROIs vary according to the value of the posterior probabilities of them being actively related to the binary response for both BNLC and BNHC models.

Top 100	Top 200	Top 300
23	99	222

Table 3.11: Top 100 represents the number of edges common among the top 100 edges identified by BNLC and BNHC. Top 200 and Top 300 are defined analogously.

Method	BNLC	BNHC	Lasso	Relión(2017)	Binary BL	Binary BHS
AUC	0.617	0.598	0.532	0.466	0.461	0.484

Table 3.12: Predictive performance of Bayesian Network Classification (BNC) vis-a-vis competitors in terms of Area Under Curve (AUC) of the ROC. AUC has been calculated in each case using 10-fold cross validation.

3.6.2 Sensitivity to the choice of hyperparameters

We have already discussed how the hyperparameters are chosen for the simulation studies and data analysis. To assess how sensitive the inferences from BNLC and BNHC are, we analyze BNLC and BNHC with different combinations of hyperparameters. Specifically for BNLC, we use the five different combinations (i)-(v) given in Section 3.5.5, and three different combinations (i)'-(iii)' for BNHC also mentioned in Section 3.5.5. We report performances on the number of influential nodes identified. We also find the number of influential edges connecting influential nodes.

Table 3.13 records the number of nodes identified as influential and the number of intersections of influential nodes between different combinations and the original analysis. Recall that the original analysis of BNLC identifies 38 influential nodes. Since this is a high

	BNLC					BNHC		
Combinations	(i)	(ii)	(iii)	(iv)	(v)	(i)’	(ii)’	(iii)’
# Nodes detected	35	39	34	40	37	45	49	44
# Intersections with original analysis	34	36	34	37	37	42	45	43

Table 3.13: Number of nodes identified as influential for all combinations are presented. The table also presents the number of intersections of influential nodes between different combinations and the original analysis.

	BNLC					BNHC		
Combinations	(i)	(ii)	(iii)	(iv)	(v)	(i)’	(ii)’	(iii)’
# Edges detected	122	113	125	118	107	272	265	262
# Intersections with original analysis	117	112	119	111	101	263	264	257

Table 3.14: Number of edges identified as influential for all combinations are presented. The table also presents the number of intersections of influential nodes between different combinations and the original analysis.

dimensional regression paradigm with number of parameters far exceeding the sample size, one expects the prior hyper-parameters to have some effect on the inference. Indeed, there is some variation in the number of identified nodes, though they largely agree with each other under different hyperparameter settings. In fact, we find a large number of intersections among the identified nodes in the original analysis with the nodes identified under different hyperparameter combinations. A similar story emerges from BNHC. We also find 31 nodes identified by all hyperparameter combinations in BNLC. Similarly, 40 nodes are identified by all hyperparameter combinations of BNHC. We calculate the number of influential edges among these $\binom{31}{2}$

edges and $\binom{40}{2}$ edges in BNLC and BNHC respectively, for all hyperparameter combinations. Table 3.14 presents the number of edges detected as influential, as well as the number of intersecting edges with the original analysis. Again, due to the high dimensionality of the problem, the variation in the number of identified edges with different choices of hyperparameters is expected, though the variation turns out not to be very significant.

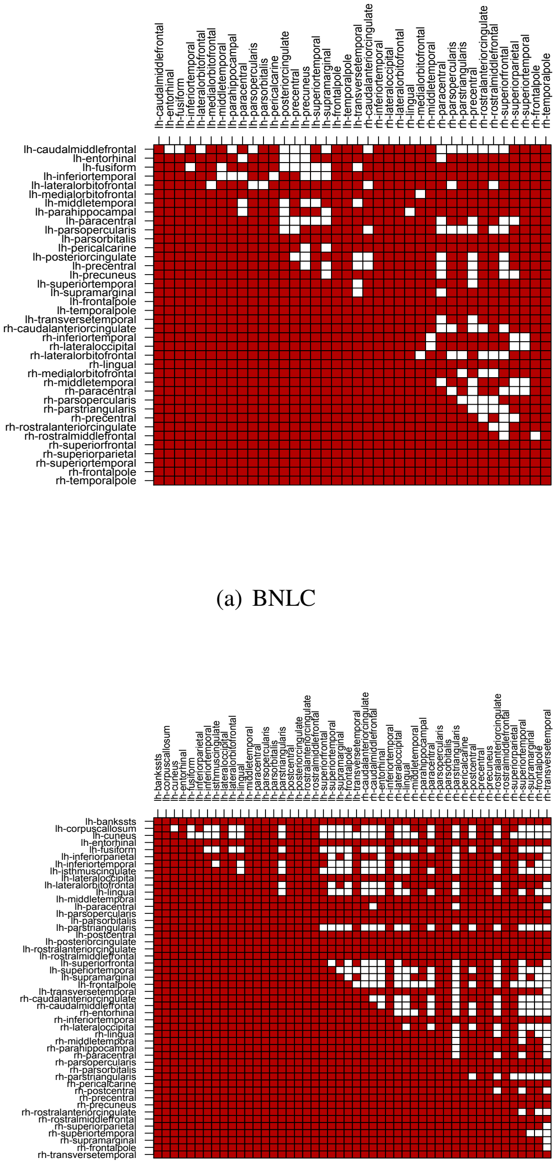
Finally, to check sensitivity to the choice of R on the performance of BNLC and BNHC, we run the data analysis for BNHC and BNLC with $R = 8$ and $R = 10$, and report the posterior mean of the effective dimensionality, along with AUC. Table 3.15 reports the posterior mean of effective dimensionality, which shows very moderate increase with increasing R. However, increasing R seems to have almost no effect on AUC.

	BNLC			BNHC		
	$R=4$	$R=8$	$R=10$	$R=4$	$R=8$	$R=10$
Posterior mean Eff. Dim.	2.17	2.78	2.96	2.00	2.74	3.04
AUC	0.61	0.63	0.59	0.59	0.60	0.59

Table 3.15: AUC and posterior mean of effective dimensionality for BNLC and BNHC under different choices of R.

3.7 Summary

We develop a binary Bayesian network regression model that enables classifying multiple networks with “labeled nodes” into two groups, identifies influential network nodes and predicts the class in which a newly observed network belongs. Our contribution lies in carefully constructing a class of network global-local shrinkage priors on the network predictor coefficient while recognizing the latent network structure in the predictor variable. In particular, we investigate two specific network shrinkage priors from this general class, leading to two network classifiers BNLC and BNHC. Our extensive simulation study shows competitive performance between BNLC and BNHC in terms of inference and classification with no clear winner, and both of them are found to outperform other competitors. Another major contribution of the proposed framework remains theoretically understanding the Bayesian network classifier model with the Network Lasso shrinkage prior. Specifically, we develop theory guaranteeing accurate classification as the sample size tends to infinity. The theoretical developments allow the number of possible interconnections in the network predictor to grow at a faster rate than the sample size. We analyze a brain connectome dataset with brain connectivity networks between different regions of interest for multiple individuals, and information on whether an individual is in a *low* or a *high* IQ category. BNC shows satisfactory out of sample classification and identifies important brain regions actively influencing the FSIQ of an individual.



(a) BNLC

(b) BNHC

Figure 3.9: Plot showing whether an edge connecting two influential nodes is influential or not. Note that the map is a $M \times M$ symmetric matrix, where M denotes the number of influential nodes, and each cell denotes an edge connecting the corresponding pair of nodes. The axis labels are the abbreviated names of the influential ROIs in the left (starting with 'lh -') and the right (starting with 'rh -') hemispheres of the brain. Full names of the ROIs can be obtained from the widely available Desikan brain atlas. A white cell represents an influential edge, while red cell represents a non-influential edge.

Chapter 4

High Dimensional Bayesian Network Mixture Regression

4.1 Introduction

Chapters 2 and 3 introduce a Bayesian framework for regression with a continuous or binary scalar response and a network predictor, and the models proposed therein directly characterize the effect of influential network nodes and edges in explaining the response y_i. However, these models tacitly assume an identical relationship between the scalar response and the network predictor for each subject, and that the same set of nodes and edges influence the regression function in a similar manner for every individual. While this assumption may hold true for some applications, it may appear to be restrictive in explaining the relationship between y_i and A_i for a variety of neuro-scientific data.

The literature provides evidence of differences in the relationship between brain connectivity networks with phenotypic traits for different groups of individuals [26]. However, flexible statistical methods for analyzing such differences have lagged behind the increasingly routine collection of such data in neuroscience. Rather than addressing the general problem of developing a flexible relationship between a response y_i and the corresponding network predictor A_i, for $i = 1, \ldots, n$, that accounts for changes in different groups of individuals, the literature has largely focused on a more specific problem where response y_i is categorized into a binary response with two groups. The literature then proceeds to identify differences between these groups and fit different models relating y_i and A_i in different groups [37].

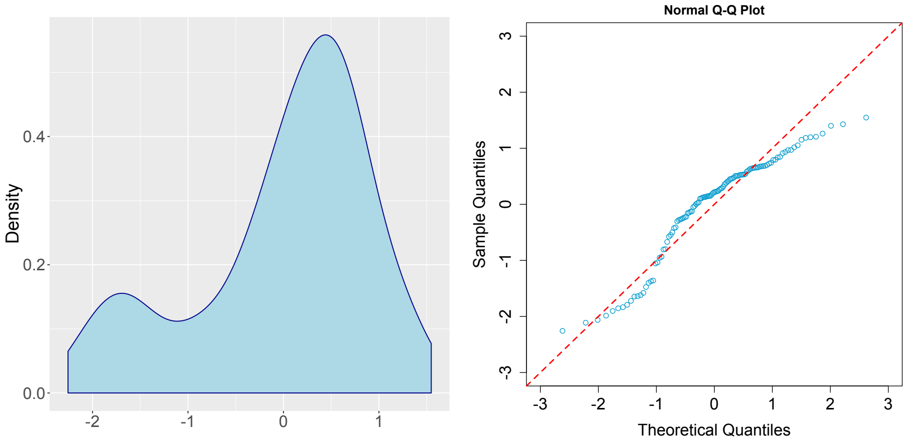
While this literature is effective in identifying differences between brain networks in two groups of individuals, it does not address a number of inferential questions of our concern. First, these methods pre-identify the two groups having potentially different relationships between the response and the network predictor before doing the analysis. Second, none of these methods focus on identifying different sets of network nodes influencing the response for different groups of individuals. This chapter will develop a flexible, nonparametric mixture modeling framework for a continuous response y_i and a network predictor A_i. The proposed framework will allow model based clustering of subjects into groups having similar relationships between y_i and A_i. In each group, the same set of network nodes will significantly influence the response. To begin, we present a dataset in the next section that motivates our framework.

4.1.1 OCEAN Brain Connectome Dataset

The dataset that we use consists of information on the *Big Five* personality traits, namely *Openness, Conscientiousness, Extraversion*, *Agreeableness* and *Neuroticism*, for every subject. The big five personality traits, also known as the *OCEAN* model, is a taxonomy for personality traits. Beneath each of the included global factors, lie a number of correlated and more specific primary factors. For example, extraversion is said to include such related qualities as gregariousness, assertiveness, excitement seeking, warmth, activity, and positive emotions. The relationships of these personality traits with major life indicators such as subjective well-being [123], career success [79], relationship attachments and outcomes [122] have been examined and recognized by neuroscientists. A personality trait for each subject has been assigned a numerical score between 0 and 100, 0 indicating mild level and 100 signifying severe level for a specific personality trait. The five personality traits are correlated, and we focus on the first principal component as our continuous response y_i for the ith individual, which captures 45% of the variability of these traits. The first principle component of the five traits in OCEAN data has been constructed and used in other studies, see [28]. In the first principle component, agreeableness contributes overwhelmingly, while extraversion, openness and conscientiousness have similar weights. The weights corresponding to neuroticism is close to zero.

Along with personality traits, we observe data on the brain connectome matrix for each individual. In this case, the brain connectome matrix for each individual is of dimension 12×12, with the (k,l)th entry signifying the total number of neuron connections between the kth and the lth brain lobes. The dataset contains information on y_i and A_i for $n = 113$ individuals.

To begin with, we fit a frequentist Lasso regression of y_i on vectorized A_i and analyze the residuals. The density plot of the residuals in Figure 4.1 shows signs of multi-modality in the distribution of the residuals, perhaps due to the difference in relationships between y_i and A_i for different groups of subjects. The BNSP model introduced in Chapter 2 is unsuitable



(a) Error density (b) QQ plot

Figure 4.1: Error density and QQ-plot of residuals after fitting Lasso on 113 subjects of OCEAN dataset.

for this data since it is based on the assumption that y_i and A_i share the same relationship for all $i = 1,..,n$. We have demonstrated this in sections 4.4.3 and 4.5.2. Additionally, the QQ plot of the standardized residuals in Figure 4.1 reveals non-normal behavior of the residuals, so that the normality of the error distribution of the BNSP model is not justified for this data. We propose to employ a Bayesian mixture model in this chapter. Bayesian mixture models are able to cluster subjects into different groups having different regression relationships between personality traits and brain connectomes. Thus, our model would offer inference on influential nodes and edges in different clusters, allowing for the scientific understanding of the relationship between personality traits and the brain connectome with characterization of uncertainty in different groups/clusters of subjects. As a byproduct, the proposed mixture model relaxes the normality assumption on the errors, deemed appropriate for the dataset of interest.

4.2 Model and Prior Specification

To develop a sufficiently flexible relationship between y_i and A_i, we propose to use a mixture model to characterize the distribution of $y_i\,|A_i$ flexibly. The conditional distribution of $y_i\,|A_i,\tau^2$, denoted by $f(y_i|A_i,\tau^2)$ is defined as

$$f(y_i|A_i,\tau^2) = \int N(y_i|\mu + \langle A_i, B\rangle, \tau^2) dG(B,\mu), \tag{4.1}$$

which can be seen as a mixture of the network regression model proposed in Chapter 2 with the mixing distribution given by $G(\cdot)$. G is a random probability measure given by

$$G = \sum_{d=1}^{H} \omega_d \delta_{(B_d^*,\mu_d^*)},\; (B_d^*,\mu_d^*) \sim G_0, \tag{4.2}$$

where G_0 is the base measure and $\delta_{(B_d^*,\mu_d^*)}$ corresponds to the Dirac-delta function at (B_d^*,μ_d^*). Equation (4.2) contains a broad class of species sampling priors, including the Dirichlet process prior through the popular stick breaking construction [119]. In this work, we jointly model cluster inclusion probabilities with the following stick breaking construction

$$\omega_1 = v_1^*,\, \omega_2 = v_2^*(1-v_1^*), .., \omega_{H-1} = v_{H-1}^* \prod_{l=1}^{H-2}(1-v_l^*),\, \omega_H = \prod_{l=1}^{H-1}(1-v_l^*),$$
$$v_l^* \sim Beta(1-\alpha_1, \alpha_2 + l\alpha_1),\, l = 1, .., H-1;\; \alpha_1 \in [0,1],\; \alpha_2 > (-\alpha_1), \tag{4.3}$$

where H is an upper bound on the number of clusters. As $H \to \infty$, this choice leads to the classical Pitman-Yor process [75]. Choosing $\alpha_1 = 0$ in the representation leads to the classical Dirichlet process prior. A useful method for selecting H is to choose a value that yields a marginal density for $y = (y_1, ..., y_n)'$ close to its limit as $H \to \infty$. Of course the adequacy of this truncation will also depend upon α_1, α_2, but even if they are unknown parameters, we can still monitor the marginal density by looking at the value for α_1, α_2 in our MCMC iterations (see [75] for more details). For implementation of our approach, we start our analysis with a bigger value of H, so that a lot of clusters are unoccupied. Parameters α_1 and α_2 are assigned $Beta(a_{\alpha_1}, b_{\alpha_1})$ and $Gamma(a_{\alpha_2}, b_{\alpha_2})$ prior distributions, respectively.

Using (4.1) and (4.2), the conditional distribution of y_i can be written as

$$f(y_i|A_i, \tau^2) = \sum_{d=1}^{H} \omega_d \mathrm{N}(y_i|\mu_d^* + \langle A_i, B_d^* \rangle, \tau^2). \tag{4.4}$$

The model presented in (4.4) acknowledges more flexible distribution in modeling $y_i|A_i, \tau^2$. Introducing a cluster index $z_i \in \{1, .., H\}$ corresponding to the individual i, we obtain $y_i|A_i, z_i, \tau^2 \sim \mathrm{N}(y_i|\mu_{z_i}^* + \langle A_i, B_{z_i}^* \rangle, \tau^2)$, with $P(z_i = d) = \omega_d$, for $d = 1, ..., H$. This conditional independence structure, given the cluster indices of the individuals, facilitates computation, while still allowing a flexible dependence structure between the different components marginally.

To develop prior distributions on μ_d^*, B_d^* and τ^2, we adopt the network shrinkage prior framework developed in Chapter 2. More precisely, let $u_{1,d}, ..., u_{V,d} \in \mathbb{R}^R$ be a collection of R-dimensional d-th mixture specific latent variables, one for each node, such that $u_{k,d}$ corresponds to node k in the d-th mixture component. Let the vectorized upper triangular part of B_d^* be given by γ_d. Each $\gamma_{k,l,d}$ is assumed to be conditionally independent with a density that can be represented as a location and scale mixture of normals as described in Chapter 2,

$$\gamma_{k,l,d}|s_{k,l,d}, u_{k,d}, u_{l,d}, \tau^2 \sim N(u_{k,d}' \Lambda_d u_{l,d}, \tau^2 s_{k,l,d}),\ s_{k,l,d} \sim Exp(\theta_d^2),\ \theta_d^2 \sim Gamma(\zeta, \iota), \tag{4.5}$$

where $s_{k,l,d}$ is the scale parameter corresponding to each $\gamma_{k,l,d}$ and $\Lambda_d = \mathrm{diag}(\lambda_{1,d}, ..., \lambda_{R,d})$ is an $R \times R$ diagonal matrix. In the same spirit as Chapters 2 and 3, we assign a spike and slab prior distribution [76] on the latent factor $u_{k,d}$ as below

$$u_{k,d} \sim \begin{cases} N(0,Q_d), & \text{if } \xi_{k,d}=1 \\ \delta_0, & \text{if } \xi_{k,d}=0 \end{cases}, \ \xi_{k,d} \sim Ber(\Delta_d),\ Q_d \sim IW(\nu,I),\ \Delta_d \sim Beta(a,b). \quad (4.6)$$

Here Q_d is a covariance matrix of order $R \times R$. The parameter Δ_d corresponds to the probability of the nonzero mixture component. Importantly, $\xi_{k,d}=0$ implies that $u_{k,d}$ has no influence in predicting the response. The location parameter μ_d is assigned a standard normal distribution. We assign a hierarchical prior $\lambda_{r,d} \sim Ber(\pi_{r,d})$, $\pi_{r,d} \sim Beta(1,r^{\eta})$, $\eta > 1$, and τ^2 is assigned a flat prior. With the construction specified as above, the form of the base measure G_0 can be expressed as $G_0(B_d^*,\mu_d^*|\tau^2) = G_{0,1}(B_d^*|\tau^2)G_{0,2}(\mu_d^*|\tau^2)$, where $G_{0,2}(\mu_d^*|\tau^2) = N(0,1)$, and $G_{0,1}(B_d^*|\tau^2)$ is expressed as follows:

$$G_{0,1}(B_d^*|\tau^2) = \int \left\{ \prod_{k<l} N(u'_{k,d}\Lambda_d u_{l,d}, \tau^2 s_{k,l,d}) p(s_{k,l}) \right\} \prod_{k=1}^{V} p(u_{k,d}) \prod_{r=1}^{R} p(\lambda_{r,d}) \prod_{r=1}^{R} d\lambda_{r,d} \prod_{k=1}^{V} du_k \prod_{k<l} ds_{k,l}.$$

The model and prior specification allows clustering of individuals into a number of classes less or equal to H. In each class, the response and network predictor is represented by separate network regression structures. Recall that in Chapters 2 and 3, all subjects share the same set of network nodes and edges actively related to the response. In the framework introduced here, subjects belonging to different clusters may have different sets of nodes and edges significantly related to the response. In the context of the brain connectome application in Section 4.5, it boils down to assuming that the relationship between the response and network predictors may vary from group to group [26].

4.3 Posterior Computations

The full posterior of parameters is intractable, hence posterior inference is carried out using MCMC. Similar to earlier chapters, all parameters except α_1 and α_2 have full conditional posterior distributions lying in standard families of distributions, as described in Appendix G. Hence Gibbs sampling with Metropolis can be readily implemented. All simulations and real data analysis results are presented with $a = 1, b = 1, \zeta = 2, \iota = 2$ and $\nu = 20$. Detailed justification for this specific choices of a, b and ν have already been provided in Chapter 3, Section 3.5. The hyperparameters ζ and ι imply a prior mean for the scale parameters $s_{k,l,d}$'s that is not too small or too large. Detailed sensitivity analyses with choices of a, b, ζ, ι, ν for both simulation studies and real data are presented subsequently. Finally, the hyperparameters a_{α_1}, b_{α_1}, a_{α_2} and b_{α_2} are chosen so that the number of clusters a priori becomes close to the eyeball estimate of the number of clusters from the plot of the response variable. We will offer more discussion on the prior number of components implied by our choice of hyperparameters in each simulation case and in the real data examples.

To assess inference of the proposed mixture model, we find the point estimate of clustering denoted by $\hat{z}$ (not reported), heat maps of the posterior probability of two samples belonging to the same cluster, $P(z_i = z_j|y)$ (which provide a measure of the uncertainty associated with the clustering), and a histogram of the posterior distribution of the number of identified clusters. The point estimate $\hat{z}$ is obtained by minimizing (using iterative componentwise optimization) the expected loss function discussed in [90],

$$F(\hat{z}) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} 1(\hat{z}_i = \hat{z}_j) \left[\frac{w_2}{w_1 + w_2} - P(z_i = z_j | y) \right]. \tag{4.7}$$

The ratio w_1/w_2 controls the relative loss of incorrectly clustering or separating a pair of samples. In our illustrations we set $w_1 = w_2 = 1$.

The posterior inference is based on 5000 suitably thinned samples from the MCMC sampler after a burn in of 20000 samples. The results are robust to small to moderate changes in the prior specification, and the convergence of parameters has been checked using convergence diagnostics available in the `coda` package in `R`.

4.4 Simulation Studies

This section considers synthetic datasets to assess the performance of our model, referred to as the Nonparametric Bayesian network regression (NBNR) model, along with a competitor.

4.4.1 Simulation Settings

To study all competitors under various data generation schemes, we simulate the response y_i depending on the network predictor A_i from the finite mixture model given by

$$y_i | A_i \sim \sum_{d=1}^{H_0} \omega_{d,0} N(\mu_{d,0} + \langle A_i, B_{d,0} \rangle_F, \tau_0^2), \tag{4.8}$$

where $B_{d,0}$, $d = 1, \ldots, H_0$ are mixture specific symmetric network coefficient matrices, each having zero diagonal entries. The network predictor matrices $A_i = ((a_{i,l,l'}))_{l,l'=1}^{V}$ are simulated by drawing the edges $a_{i,l,l'}$ i.i.d from N(0,1), for $l < l'$, $a_{i,l,l'} = a_{i,l',l}$ and $a_{i,l,l} = 0$.

To simulate the network coefficients, we draw V latent variables $u_{d,k,0}$, each of dimension R_g, from a mixture distribution given by

$$u_{d,k,0} \sim \pi_0 N_{R_g}(u_{d,m,g}, u_{d,s,g}^2) + (1-\pi_0)\delta_0; k \in \{1,...,V\}, \tag{4.9}$$

where δ_0 is the Dirac-delta function and π_0 is the probability of any $u_{d,k,0}$ being nonzero in the truth, $d = 1,...,H_0$ (fixed at 0.5 for all simulations). We then consider six different cases as following:

Cases 1-5: In Cases 1-5, the (l,l')th element of the network predictor coefficient $B_{d,0}$ corresponding to the d-th mixture component is constructed using a low-rank approach $b_{d,0,l,l'} = u'_{d,l,0}u_{d,l',0}/2$, accounting for the interaction between nodes l and l', for all $d = 1,...,H_0$. The 5 different cases are obtained by varying the number of true mixture components (H_0), the number of mixture components in the fitted model (H) and sample size (n) , as summarized in Table 4.1.

Case 6: In Case 6, we consider $H_0 = 3$, $\omega_{1,0} = \omega_{2,0} = \omega_{3,0} = 1/3$, and $B_{1,0}$, $B_{2,0}$ and $B_{3,0}$ are simulated in different ways as following:

Simulating $B_{1,0}$: The (l,l')th element of the network predictor coefficient $B_{1,0}$ is constructed using a low-rank approach $b_{1,0,l,l'} = u'_{1,l,0}u_{1,l',0}/2$.

Simulating $B_{2,0}$: $B_{2,0}$ is simulated as a block diagonal matrix with three 5×5 symmetric diagonal blocks, each having 0 diagonal entries. The upper triangular entries of the three block matrices are simulated from $N(1,1)$, $N(2,2)$ and $N(3,3)$ distributions, respectively.

Simulating $B_{3,0}$: If both $u_{3,l,0}, u_{3,l',0} \neq 0$, $b_{3,0,l,l'}$ is simulated from $N(0,1)$, otherwise $b_{3,0,l,l'}$ is set to 0. Model is fitted with $H = 15$ clusters for the data generated in Case 6. All cases fixes the number of network nodes at $V = 15$, as summarized in Table 4.1.

The intercept $\mu_{d,0}$, $d = 1,...,H_0$ in each mixture component is drawn from $N(0.5, 0.2)$, while τ_0^2 is fixed at 0.1.

Cases	H_0	H	V	n
Case - 1	2	15	15	400
Case - 2	2	15	15	200
Case - 3	3	15	15	400
Case - 4	4	20	15	500
Case - 5	1	20	15	200
Case - 6	3	15	15	450

Table 4.1: Table presents different cases in the simulation study. The parameters H_0, H refer to the true and fitted number of mixture components in the nonparametric Bayesian network regression model. Different cases also present various combinations of the number of network nodes V and sample size n.

Note that **Cases 1-5** represent the true model being included in the class of fitted models. On the other hand, **Case 6** shows departure of the true model from the class of fitted models. For each of the six cases, each component of the mean vector $u_{d,m,g}$ is randomly generated to lie between $(-2, 2)$ and the standard deviation $u_{d,s,g}$ is generated between 0.3 and 2.

4.4.2 Competitors and Metrics of Evaluation

As a competitor of our model, we employ the Bayesian network shrinkage prior (BNSP) regression model proposed in Chapter 2. BNSP assumes (a) the same set of influential nodes and edges affect a neurological response for every individual, and, (b) normality of the error distribution. Hence, comparison with BNSP will help assess the inferential advantage of our proposed model over BNSP when the data supports different relationships between response and network predictor for different groups of individuals and non-normality of the error distribution.

The competitors are assessed based on their ability to estimate the true regression function $E_0[y_i|A_i] = \sum_{d=1}^{H_0} \omega_{d,0}(\mu_{d,0} + \langle A_i, B_{d,0}\rangle)$. In particular, we compute mean squared error (MSE) of estimating the true regression function over all data points given by $\frac{1}{n}\sum_{i=1}^{n}(E_0[y_i|A_i] - \widehat{E[y_i|A_i]})^2$, where $\widehat{E[y_i|A_i]}$ denotes the posterior mean of the regression function from a competing method. While MSE offers an evaluation of point estimation by both competitors, the uncertainty in estimating the true regression function is measured using the coverage and length of 95% credible intervals obtained from the competing methods.

We also compare between BNSP and NBNR in terms of a popular model fitting statistic, referred to as the posterior predictive loss criterion (PPLC) [51]. PPLC is described as the sum of two quantities G and P, where G represents the quality of model fitting and P represents the complexity of the model. The resulting quantity $D = G + P$ strikes a balance between model fit and model complexity.

In addition to reporting the posterior distribution of the number of clusters and the uncertainty associated with clustering through $P(z_i = z_j|y)$ in the simulation studies, we also evaluate the ability of the models to identify clusters using the Adjusted Rand Index (ARI) [73] of the posterior cluster configurations with respect to the known cluster configuration. The ARI evaluates the agreement in cluster assignment between two cluster configurations. For any two partitions $\mathcal{C}_1$ and $\mathcal{C}_2$ of $\{1, ..., n\}$, the Rand index calculates the ratio of agreement between $\mathcal{C}_1$ and $\mathcal{C}_2$ of $\{1, ..., n\}$. Three quantities denoted as c_1, c_2 and c_3 are calculated: c_1 represents the number of pairs of objects that are placed in the same cluster in $\mathcal{C}_1$ and the same cluster in $\mathcal{C}_2$, c_2 are the pairs that are in different clusters in both partitions, and c_3 is the total number of pairs equaling $\binom{n}{2}$. The Rand index (RI) is RI= $\frac{c_1+c_2}{c_3}$. The adjusted Rand index (ARI) is corrected for chance. It ranges between -1 and 1, with larger values indicating agreement between cluster configurations.

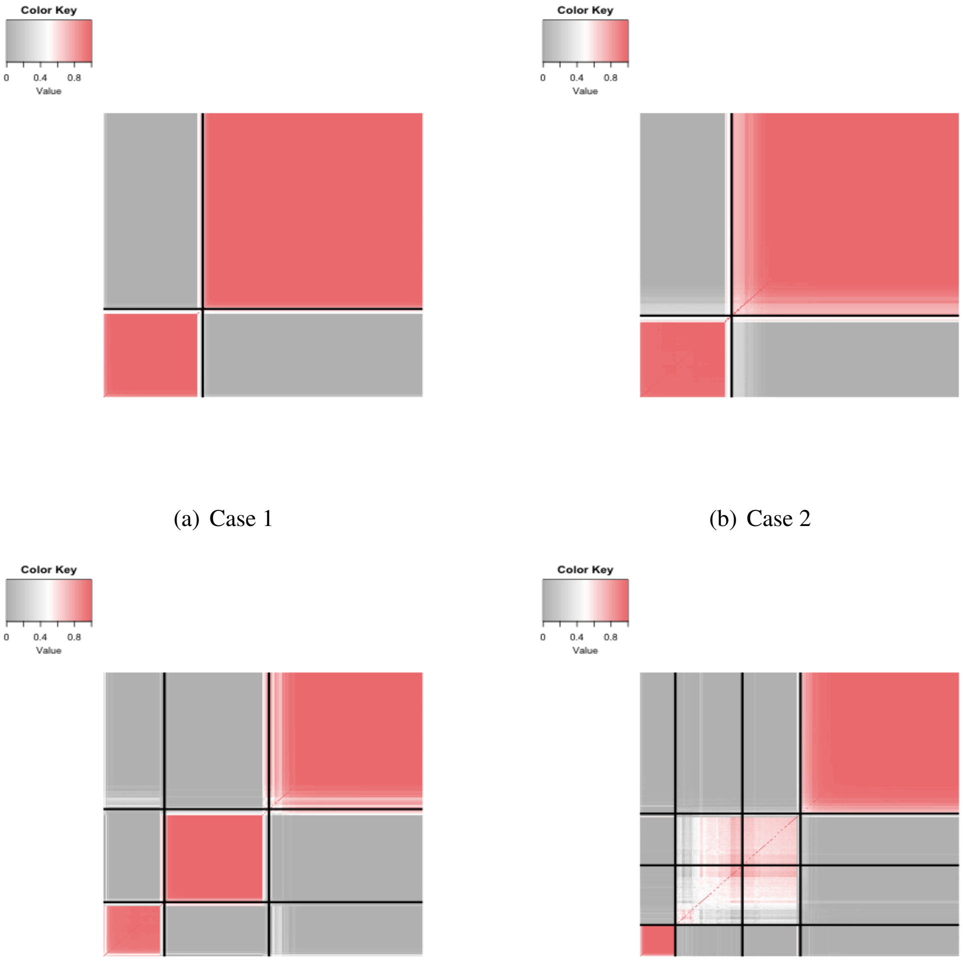
4.4.3 Simulation results

We note that our choice of hyperparameters a_{α_1}, b_{α_1}, a_{α_2} and b_{α_2} ensures mean number of clusters to be approximately 3.97, 3.62, 3.97, 4.17, 3.62 and 3.02 respectively for the 6 simulation cases a priori. Figures 4.4, 4.2 and 4.3 provide insight into the estimates of the cluster structure and associated uncertainty by displaying discrepancy between true and estimated number of clusters and heat maps of posterior probabilities of pairs of subjects belonging to the same cluster. To facilitate visualization, regions are ordered according to the true cluster configuration in the heatmap. In cases 1-3, the model recovers the true cluster structure, with little uncertainty associated with the estimator. In case 4, it appears that clusters 2 and 3 are not well identifiable, and hence the estimation as well as the uncertainty characterization suffer. Specifically, the two middle clusters (clusters 2 and 3) show much higher uncertainties. In case 5, the model identifies the true single cluster quite well. The most challenging case among all is case 6, which corresponds to model mis-specification. With model mis-specification, estimation of clusters becomes more challenging, with the posterior distribution of ARI concentrating below the other cases. Further, in case 6, there appears to be higher uncertainty with elements in the third cluster, where the model is a bit uncertain about whether to include some samples in the 2nd or the 3rd cluster.

The posterior distributions of the number of identified clusters are also presented in the form of barplots in Figure 4.5. Consistent with the story presented so far, the posterior distribution of the number of clusters appears to concentrate around the true number of clusters H_0 in cases 1-5. Cases 1, 2, 3 and 5 show clear modes at the true number of clusters. The posterior distribution of the number of clusters in Case 4 also finds mode at the truth $H_0 = 4$, though posterior probability of the number of clusters equalling 5 turns out to be high. The most difficult case is case 6, with model mis-specification, where the model mildly overestimates the posterior probability of the number of clusters. Note that out of H assigned clusters, most are not populated in each case. Hence the choice of H is sufficient in each case.

Table 4.2 presents mean squared errors (MSE) for the estimates of the regression mean function under each of the competitors. Further, coverage and average length of 95% credible intervals are provided to assess how well calibrated the estimates are. A few interesting observations emerge from Table 4.2. Comparing cases 1 and 2, it turns out that NBNR offers smaller MSE and narrower credible intervals when the sample size is smaller. Also, comparing cases 3 and 4, it appears that increasing the true number of mixture components H_0 results in a considerable increase in MSE and the length of 95% credible interval. Except for case 5, NBNR demonstrates coverage more than nominal in every other case.

Given that the data have been generated from a mixture of network regression distributions except for case 5, BNSP is expected to perform inferior to NBNR. Indeed, Table 4.2 shows substantially inferior MSE and much wider credible intervals offered by BNSP compared to NBNR in all other cases, except case 5. In case 5, when data has been generated from the



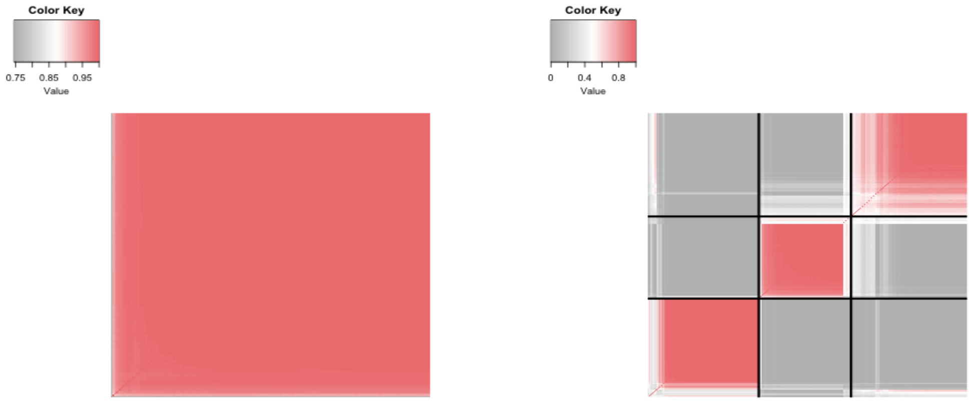
(a) Case 1

(b) Case 2

(c) Case 3

(d) Case 4

Figure 4.2: Plots showing uncertainty in estimating the clusters in the simulation cases 1-4. Boldfaced horizontal and vertical lines indicate the true clustering.

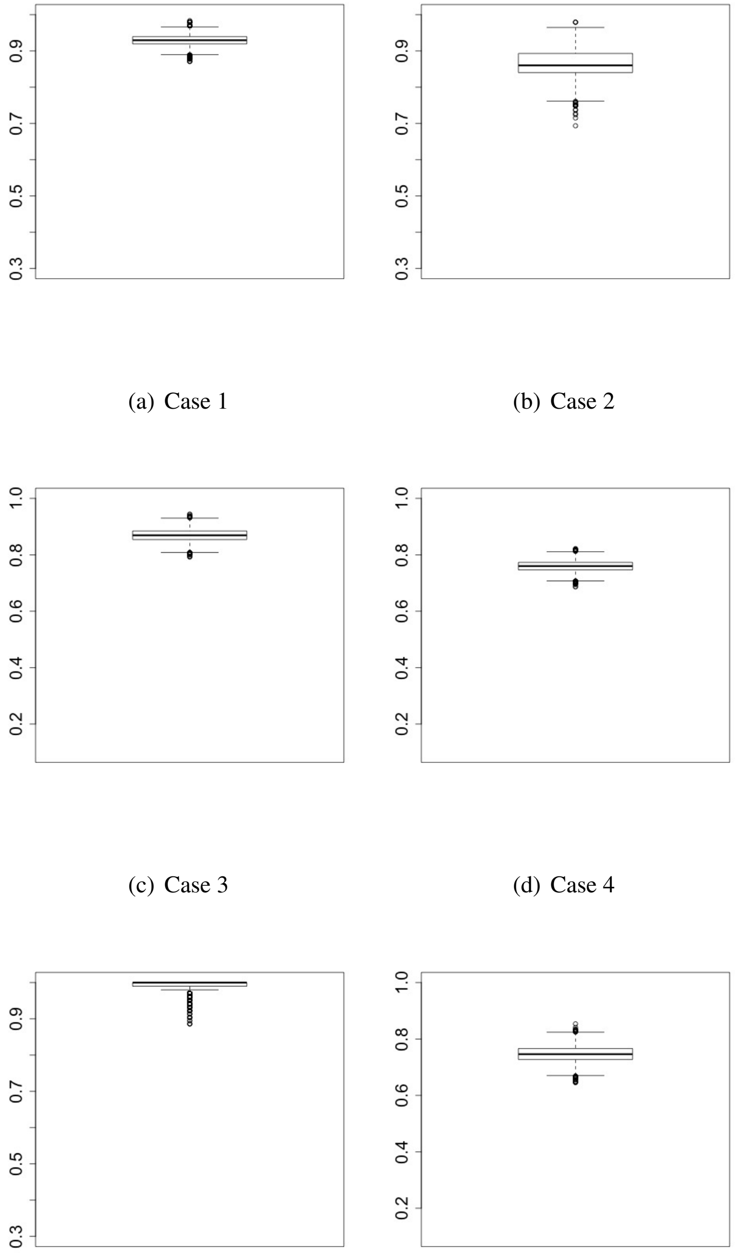


(a) Case 5 (b) Case 6

Figure 4.3: Plots showing uncertainty in estimating the clusters in the simulation cases 5-6. Boldfaced horizontal and vertical lines indicate the true clustering.

BNSP model (i.e., NBNR model with $H_0 = 1$), BNSP is found to provide mildly better MSE than NBNR. While the coverage of both competitors are close to nominal in case 5, BNSP displays shorter credible interval than NBNR.

Similar to the MSE values, the G values representing fit of the two models (Table 4.3) show superior performance of NBNR when the data are simulated from a mixture model with more than one mixture component. In case 5, with the true data generating model being the BNSP model, BNSP demonstrates better performance than NBNR. The P values increase when the number of true mixture components grow. We also find a sharp increase in the P value for case 4 which represents both higher H_0 and H. Overall, the model fitting statistics reveal advantages of fitting NBNR over BNSP in presence of data generated from a mixture distribution.



(a) Case 1

(b) Case 2

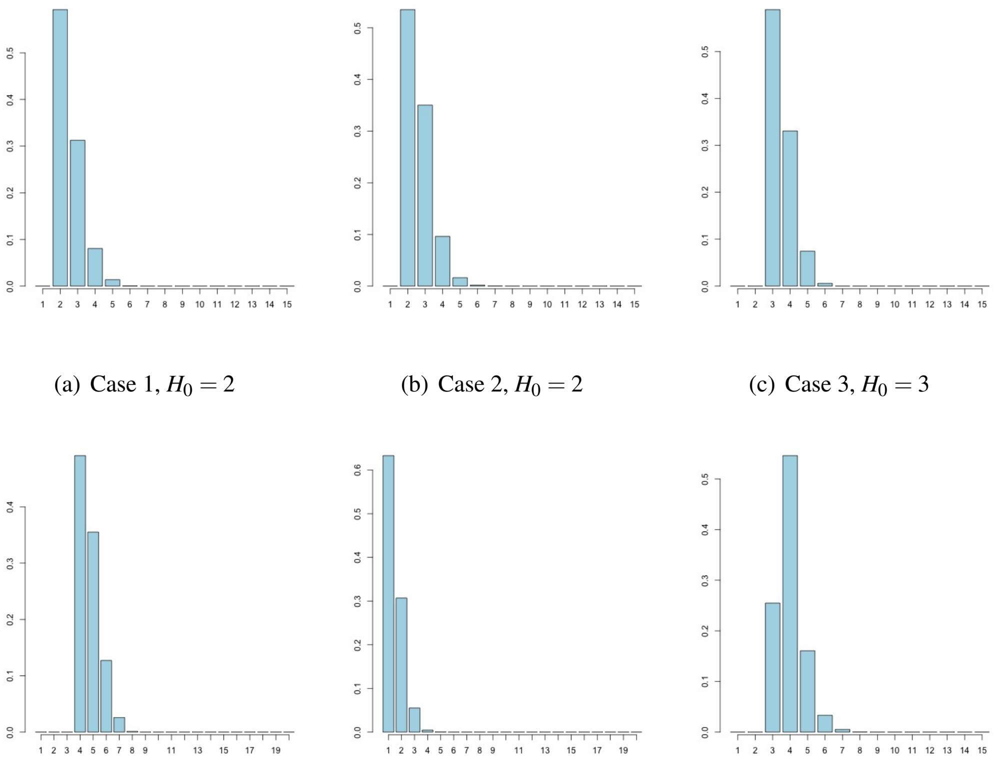
(c) Case 3

(d) Case 4

(e) Case 5

(f) Case 6

Figure 4.4: Posterior distribution of ARI in the 6 simulation cases.



(a) Case 1, $H_0 = 2$

(b) Case 2, $H_0 = 2$

(c) Case 3, $H_0 = 3$

(d) Case 4, $H_0 = 4$

(e) Case 5, $H_0 = 1$

(f) Case 6, $H_0 = 3$

Figure 4.5: Bar plots showing the posterior distribution of the number of chosen clusters by the model in the 6 simulation cases. The true number of clusters H_0 is also mentioned in each case.

Mean Squared Error (MSE)

Method	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
NBNR	**0.02**	**0.07**	**0.65**	**2.62**	0.36	**0.07**
BNSP	18.74	10.97	48.38	37.96	**0.34**	16.77

Coverage of 95% Credible Interval (CI)

Method	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
NBNR	0.99	0.99	0.99	0.98	0.95	0.98
BNSP	0.94	0.93	0.94	0.94	0.93	0.93

Average Length of 95% Credible Interval (CI)

Method	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
NBNR	1.70	2.05	2.87	7.98	2.68	1.83
BNSP	16.97	12.19	25.52	23.44	2.03	15.02

Table 4.2: Mean squared error (MSE), coverage and length of 95% credible intervals in estimating the regression function for NBNR and BNSP are provided for all the cases.

G

Method	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
NBNR	18.88	3.93	14.72	55.64	10.85	10.42
BNSP	29047.54	3498.95	69409.94	92226.81	9.49	30595.55

P

Method	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
NBNR	51.22	35.63	257.09	1343.33	21.01	98.67
BNSP	40099.77	6441.24	92223.09	115729.9	18.72	39420.5

D

Method	Case 1	Case 2	Case 3	Case 4	Case 5	Case 6
NBNR	70.11	39.53	271.81	1398.95	31.87	109.09
BNSP	69147.30	9940.19	161633.03	207956.7	28.22	70016.05

Table 4.3: G, P, D values for BNSP and NBNR for all six cases.

4.4.4 Sensitivity to the choice of hyperparameters in simulations

To check sensitivity of inference to the choice of hyper-parameters, we consider a representative case (case 3) and re-analyze the same simulated data with different combinations of hyper-parameters. In particular, we consider five different hyper-parameter settings for case 3 and compare the inference with the results on case 3 presented earlier. The five combinations are given by, (i) $a = 1, b = 9, \nu = 20, \frac{\zeta}{\iota} = 1$; (ii) $a = 1, b = 1, \nu = 20, \frac{\zeta}{\iota} = 0.2$; (iii) $a = 1, b = 1, \nu = 20, \frac{\zeta}{\iota} = 5$; (iv) $a = 1, b = 1, \nu = 50, \frac{\zeta}{\iota} = 0.2$; (v) $a = 1, b = 1, \nu = 50, \frac{\zeta}{\iota} = 5$. Notice that (i) presents a priori mean of 0.1 for each Δ_d. Again, (ii), (iv) represent low prior means for θ_d, while (iii) and (v) represent higher prior means for θ_d. The various combinations also present variations of the hyperparameter ν in the Inverse-Wishart distribution of Q_d.

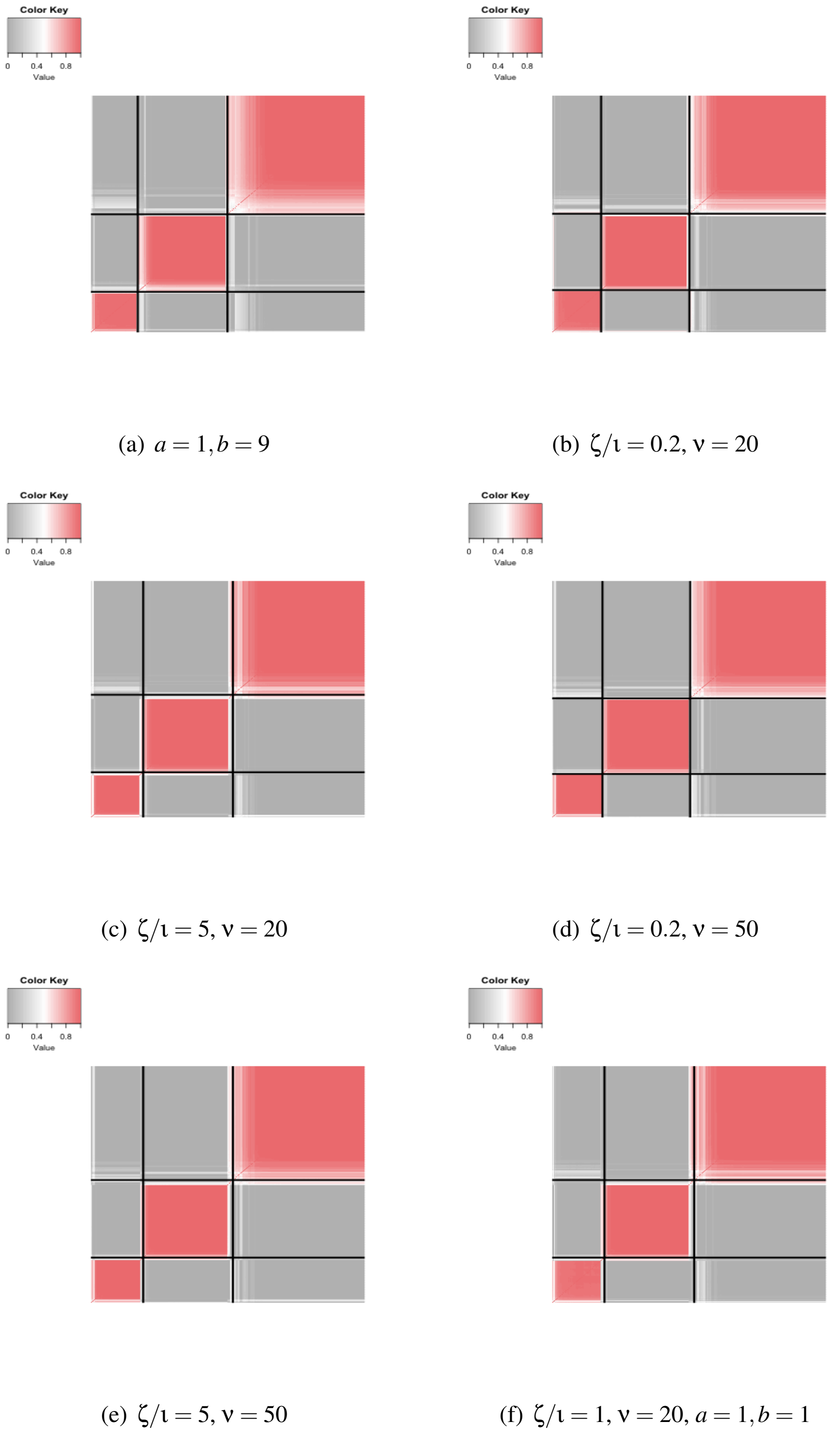
Figure 4.6 shows the uncertainty quantification associated with clustering for the five different settings and compares them with Figure 4.6(f) (the original setting). Of all the parameters, only variations in a and b seem to have an effect in the inferences, but this effect is found to be very small. The posterior distributions of the number of clusters presented in Figure 4.7 for different settings also show mildly sensitive results with changes in hyper-parameters a and b, though the distribution is generally much less affected by changes in other hyper-parameters. The posterior mean of the number of clusters in five combinations are presented in Table 4.2. The posterior mean of the number of clusters in the original case 3 is 3.52 and the corresponding results from combinations (ii)-(v) are very close. Only combination (i) shows an overestimation in the posterior mean number of clusters. A similar trend appears in the posterior distribution of ARI, as presented in Figure 4.8. The MSE, coverage and length of 95% credible intervals

	$a = 1, b = 9$	$\frac{\zeta}{\iota} = 0.2, \nu = 20$	$\frac{\zeta}{\iota} = 5, \nu = 20$	$\frac{\zeta}{\iota} = 0.2, \nu = 50$	$\frac{\zeta}{\iota} = 5, \nu = 50$
MSE	0.70	0.55	0.50	0.66	0.82
Coverage	0.99	0.99	0.99	0.99	0.98
Length	3.29	2.90	2.98	2.90	2.92
M.C.	4.2	3.75	3.42	3.44	3.43

Table 4.4: Mean squared error (MSE), coverage and length of 95% credible intervals in estimating the regression function for NBNR under different hyper-parameter settings. The last row of the table shows the posterior mean of the number of clusters (M.C. or *mean number of clusters*) in the five different hyperparameter combinations.

for different hyper-parameter combinations are also presented in Table 4.4 and are compared with corresponding results from case 3 in Table 4.2. The results appear to be of the same order in different hyper-parameter settings with NBNR maintaining significant advantage in terms of point estimation and uncertainty over BNSP under all these hyper-parameter settings.

In addition to investigating sensitivity of inference with the choice of hyperparameters, we also check sensitivity with the choice of prior distribution on ω_1,...,ω_H. As discussed earlier, the Pitman-Yor process is derived using a stick breaking construction of ω_1,...,ω_H. We also draw inference in case 3 using an alternative construction of the prior on $(\omega_1,...,\omega_H)$ that specifies $(\omega_1,...,\omega_H) \sim Dir(\tilde{\alpha}/H,...,\tilde{\alpha}/H)$, where $\tilde{\alpha} > 0$ and $\tilde{\alpha}$ follows a Gamma distribution with parameters implying a prior mean of the number of clusters ≈ 3.32. The plots for prior distribution of the number of clusters for the Pitman-Yor prior in case 3 and this truncated DP prior are shown in Figure 4.9. As $H \to \infty$, this prior converges to the Dirichlet process prior. The ARI, posterior distribution of the number of clusters and uncertainty in clustering are presented in Figure 4.10. The sensitivity of the results to this different prior choice on $\omega_1,...,\omega_H$



(a) $a = 1, b = 9$

(b) $\zeta/\iota = 0.2$, $\nu = 20$

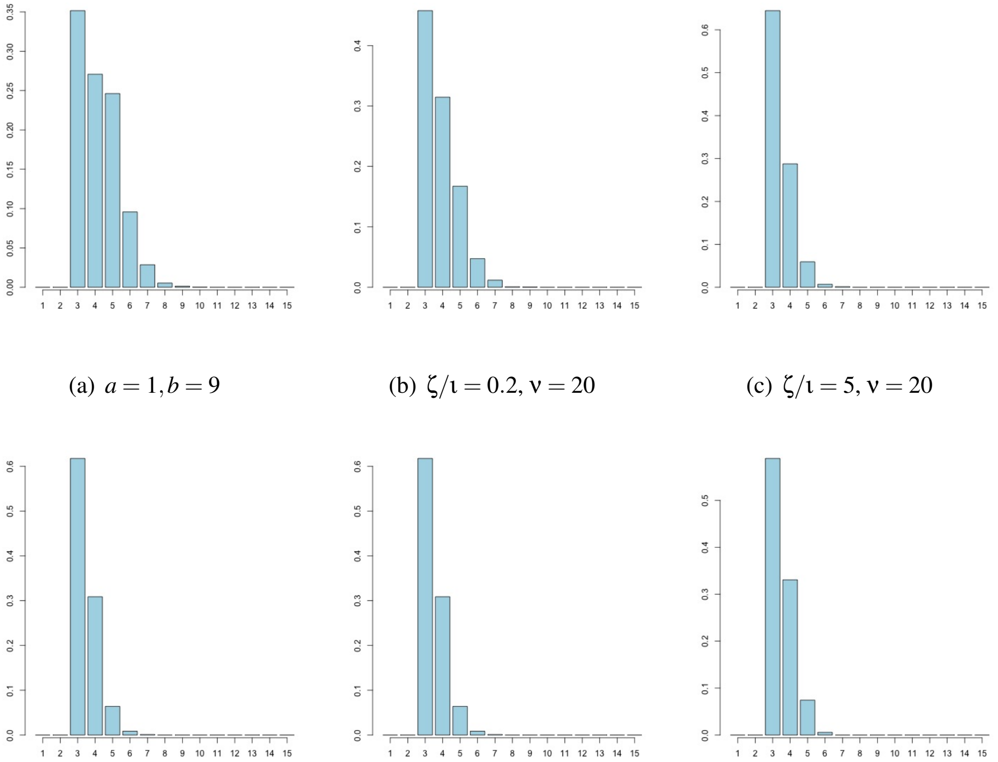
(c) $\zeta/\iota = 5$, $\nu = 20$

(d) $\zeta/\iota = 0.2$, $\nu = 50$

(e) $\zeta/\iota = 5$, $\nu = 50$

(f) $\zeta/\iota = 1$, $\nu = 20$, $a = 1, b = 1$

Figure 4.6: Plots showing uncertainty in estimating the clusters under various hyper-parameter settings in Case 3.



(a) $a = 1, b = 9$

(b) $\zeta/\iota = 0.2$, $\nu = 20$

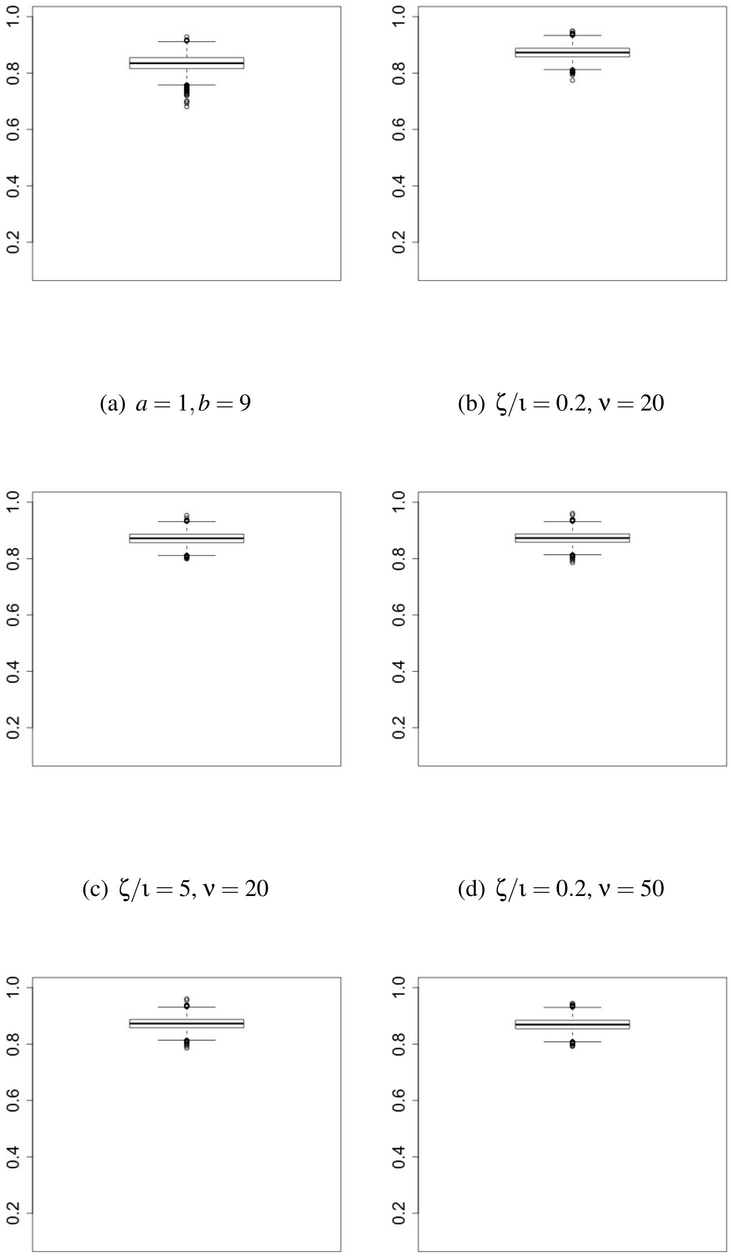
(c) $\zeta/\iota = 5$, $\nu = 20$

(d) $\zeta/\iota = 0.2$, $\nu = 50$

(e) $\zeta/\iota = 5$, $\nu = 50$

(f) $\zeta/\iota = 1$, $\nu = 20$, $a = 1, b = 1$

Figure 4.7: Bar plots showing the posterior distribution of the number of chosen clusters by the model under various hyper-parameter settings in Case 3.



(a) $a = 1, b = 9$

(b) $\zeta/\iota = 0.2$, $\nu = 20$

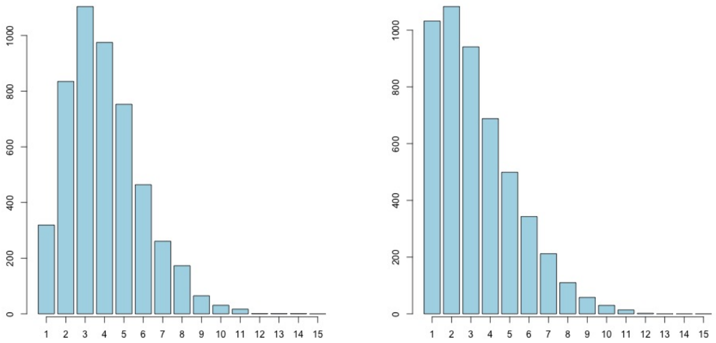
(c) $\zeta/\iota = 5$, $\nu = 20$

(d) $\zeta/\iota = 0.2$, $\nu = 50$

(e) $\zeta/\iota = 5$, $\nu = 50$

(f) $\zeta/\iota = 1$, $\nu = 20$, $a = 1, b = 1$

Figure 4.8: Posterior distribution of ARI in various hyper-parameter combinations for sensitivity analysis in simulation.



(a) PY prior

(b) truncated DP prior

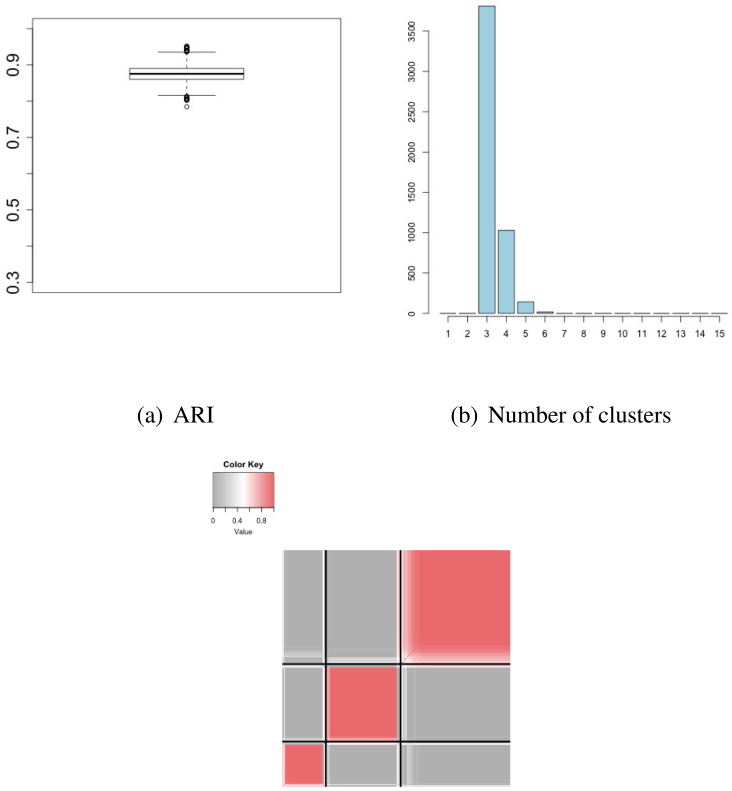
Figure 4.9: Prior distribution of the number of clusters for our choice of PY prior in case 3 and the truncated Dirichlet process prior.

is very small. The posterior mean of the number of clusters is 3.34, which is close to the what has been obtained in the original analysis of case 3. The MSE, coverage and length of 95% CI for the posterior mean function turn out to be 0.61, 0.99 and 2.93 which are very close to the numbers corresponding to case 3 in Table 4.2.

4.5 Brain Connectome Data Application

This section reports analysis of the OCEAN brain connectome dataset described in Section 4.1.1. We fit NBNR for $H = 20$ to allow a sufficient number of clusters to be identified. Table 4.5 shows that the model fit corresponding to NBNR improves over BNSP by a considerable margin, indicating the need for fitting the Bayesian mixture model to this data.

The left panel in the first row of Figure 4.11 shows the distribution of the number of clusters implied by our choice of prior hyperparameters. The distribution is bimodal in 2 and 3 and there is a considerable mass at 4. The right panel in the first row shows the posterior

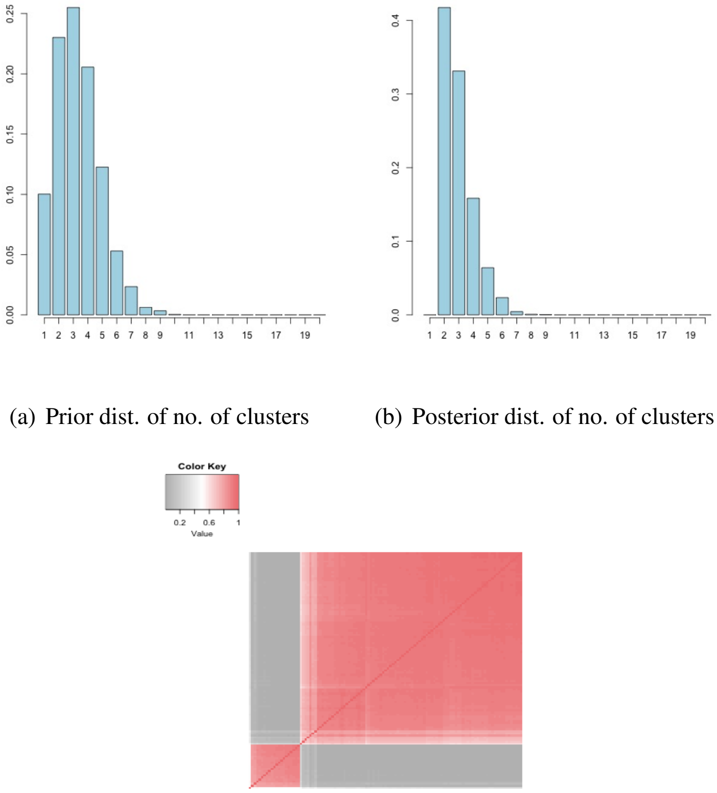


(a) ARI

(b) Number of clusters

(c) Uncertainty in clustering

Figure 4.10: Posterior distribution of ARI, the number of clusters and the uncertainty related to clustering are presented for the choice $(\omega_1,\ldots,\omega_H) \sim Dir(\alpha/H,\ldots,\alpha/H)$.



(a) Prior dist. of no. of clusters

(b) Posterior dist. of no. of clusters

(c) Uncertainty

Figure 4.11: OCEAN Data: 4.11(a) shows the distribution of the number of clusters implied by our choice of prior hyperparameters. 4.11(c) shows the uncertainty in estimating the clusters. 4.11(b) shows a barplot for the posterior dist. of the estimated number of clusters. The inference is presented for $H = 20$.

Methods	NBNR with $H = 20$			BNSP		
Comparison Metric	G	P	D	G	P	D
Values	15.35	31.69	47.05	101.98	114.11	216.09

Table 4.5: Model fitting statistics for NBNR and BNSP for the OCEAN data.

distribution of the number of clusters (figure 4.11) has a clear mode at 2. Figure 4.11 in the second row displays the heat map of posterior probabilities of any pair of individuals lying in the same cluster. The model fit with $H = 20$ shows two prominent clusters a posteriori. Importantly, there is no posterior probability of having more than 8 clusters, suggesting that $H = 20$ is sufficiently large and appropriate for this analysis.

Influential nodes for Group 1	
Left Hemisphere Lobes	Temporal, Cingulate, Frontal, Occipital
Right Hemisphere Lobes	Parietal, Insula
Influential nodes for Group 2	
Left Hemisphere Lobes	Temporal, Frontal, Parietal, Insula
Right Hemisphere Lobes	Cingulate, Frontal, Occipital

Table 4.6: Brain regions (ROIs) detected as influential for the two detected clusters of individuals in the OCEAN dataset.

We supply the model with the estimated cluster indicators and run it again to draw further inference on the influential nodes and edges in the two clusters. Table 4.6 presents the influential nodes identified in the two clusters. Note that 6 and 7 nodes (out of 12) are identified

Influential nodes	
Left Hemisphere Lobes	Temporal, Frontal, Occipital, Parietal
Right Hemisphere Lobes	Frontal, Occipital, Parietal, Cingulate

Table 4.7: Brain regions (ROIs) detected as influential by BNSP in the OCEAN dataset.

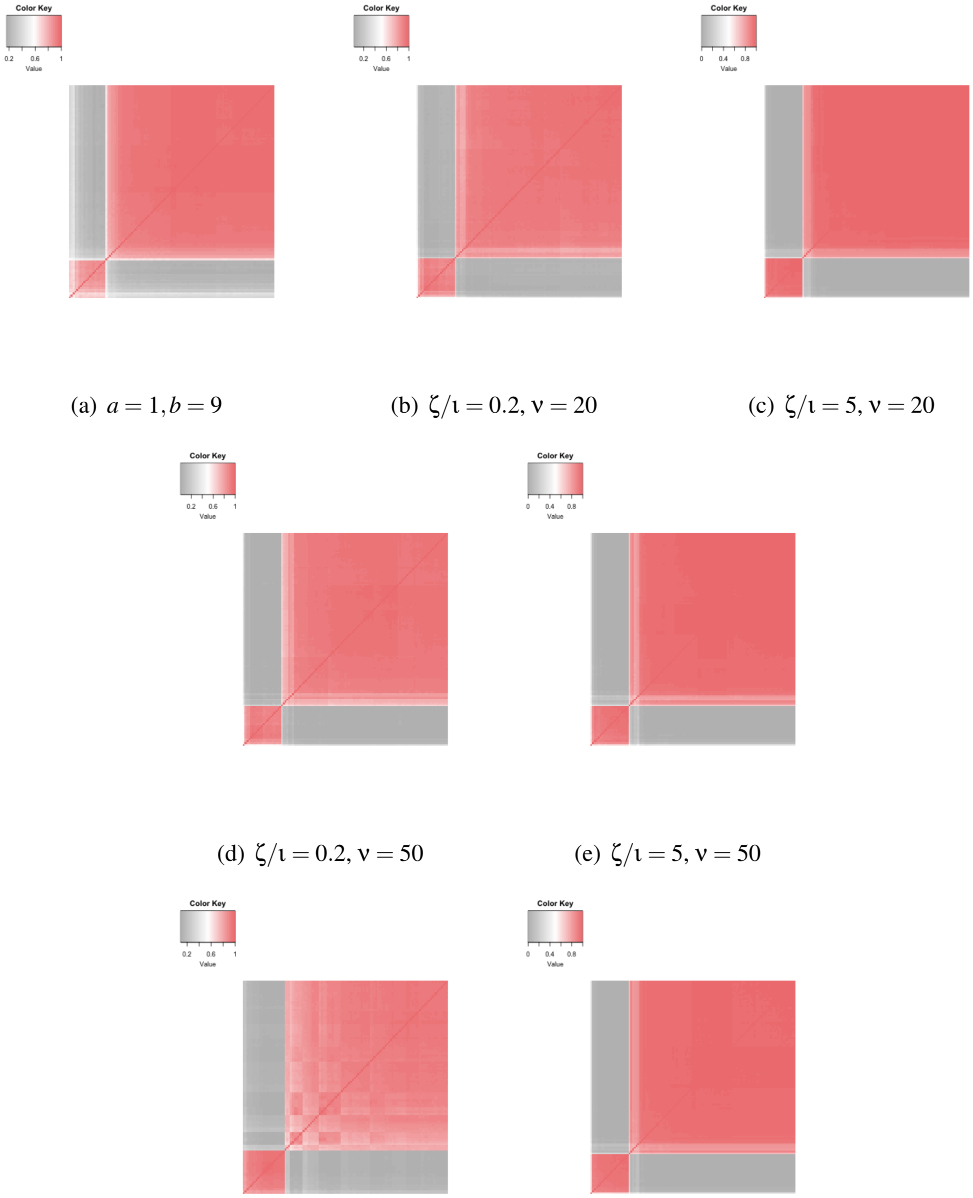
for the two groups of individuals as influential, respectively. Both groups identify the temporal and frontal lobes as influential in the left hemisphere, but only the second group identifies frontal lobe as influential in the right hemisphere. We also identify 7 and 18 influential edges connecting the influential nodes for the two groups of individuals. Among these, there are 2 common edges connected to the frontal lobe and 1 common edge connected to the temporal lobe. We also fit BNSP to this dataset which identifies 4 lobes in each hemisphere as influential. These 8 lobes include the temporal and frontal lobes in the left hemisphere, and the frontal lobe in the right hemisphere. BNSP identifies 16 influential edges connecting the influential lobes.

4.5.1 Sensitivity to the choice of hyperparameters in the OCEAN data

Similar to simulation studies, we also present sensitivity of inference in the OCEAN data analysis to different choices of the hyperparameters. We stick to the five combinations of hyperparameters presented in Section 4.4.4. Additionally, we consider two more combinations. In one of them, we set $\alpha_1 = 0$, so that the Pitman-Yor prior becomes equivalent to a Dirichlet process prior. In the other combination, we change the hyperparameters of the Pitman-Yor process so that the prior distribution of the number of clusters is concentrated much higher than what we have used in our analysis. In fact, the prior mean of the number of clusters is ≈ 10 with this choice of hyperparameters. The second column of Table 4.8 presents the discrepancy between optimal clusters identified by each combination and the optimal clusters identified by our original analysis. Except for combination $a = 1, b = 9$, and for $\alpha_1 = 0$, they turn out to be perfect matches. Figure 4.12 presents heat maps of posterior probabilities of pairs of individuals lying in the same cluster under all combinations. The uncertainty tends to be similar under all combinations. Additionally, the posterior distribution of the number of clusters displays mode at 2 for all combinations of hyperparameters (see Figure 4.13). With a higher prior mean of the number of clusters, we might expect the inference to deteriorate, but are pleasantly surprised to see the inference not being affected. Perhaps the larger sample size is responsible for good inference under this setting. We also report model fitting statistics (G, P, D) for all these combinations, which can be compared with the results presented in Table 4.5. The model fitting turns out to be very similar under all the combinations, except for somewhat inferior performance in terms of PPLC in (i) with $a = 1, b = 9$.

4.5.2 Analysis of a Brain Connectome Dataset with Composite Creativity Index (CCI) as the Response

In this section, we analyze the brain connectome data described in Chapter 2 using the NBNR model proposed in this chapter, with one exception. Recall that in Chapter 2, corresponding to every individual, we have a brain network predictor of dimension 68×68 with 68 nodes in the network representing 68 ROIs. However, working with ROI level data is computationally challenging in the context of nonparametric mixture models. Also, from various simulation studies we realize that the performance of the method deteriorates considerably when



(a) $a = 1, b = 9$

(b) $\zeta/\iota = 0.2,\ \nu = 20$

(c) $\zeta/\iota = 5,\ \nu = 20$

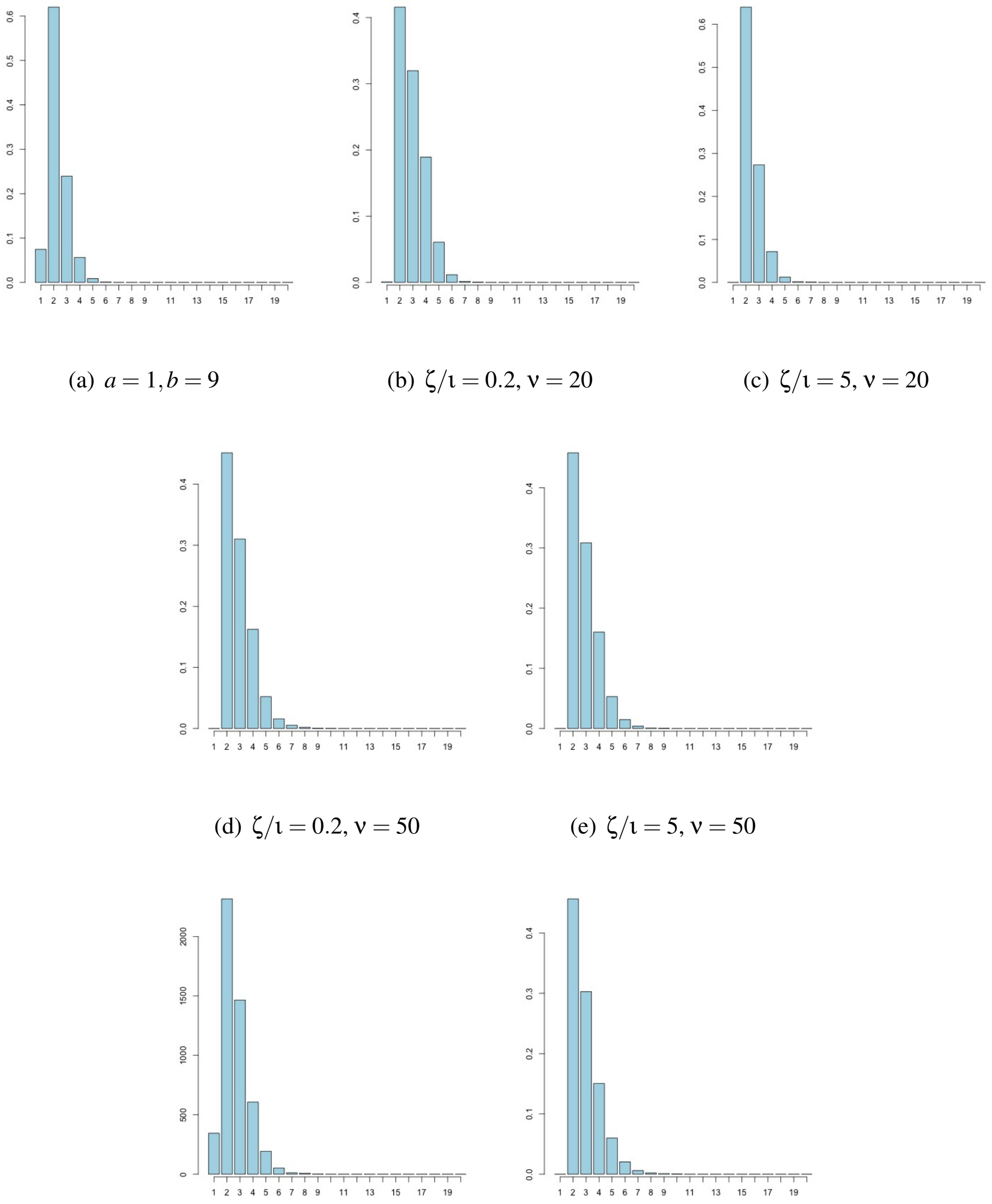
(d) $\zeta/\iota = 0.2,\ \nu = 50$

(e) $\zeta/\iota = 5,\ \nu = 50$

(f) $\alpha_1 = 0$

(g) PY: higher mean

Figure 4.12: Plots showing uncertainty in estimating the clusters under various hyperparameter settings in the OCEAN data.



(a) $a = 1, b = 9$

(b) $\zeta/\iota = 0.2$, $\nu = 20$

(c) $\zeta/\iota = 5$, $\nu = 20$

(d) $\zeta/\iota = 0.2$, $\nu = 50$

(e) $\zeta/\iota = 5$, $\nu = 50$

(f) $\alpha_1 = 0$

(g) PY: higher mean

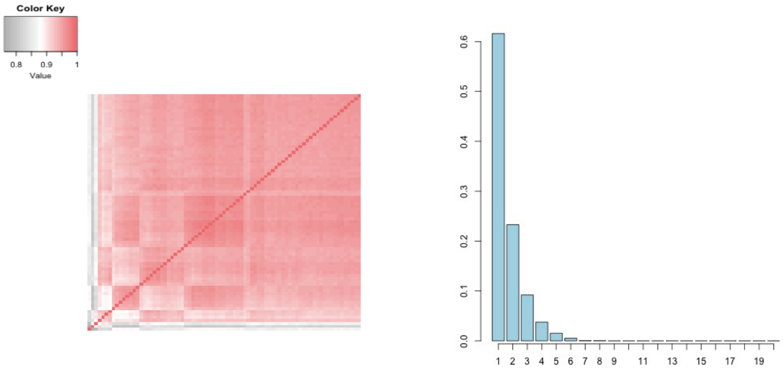
Figure 4.13: Barplots showing the posterior distribution of the number of chosen clusters by the model under various hyperparameter settings in the OCEAN data.

Methods	NBNR with $H = 20$			
Comparison Metric	ARI	G	P	D
$a = 1, b = 9$	0.96	19.54	53.43	72.97
$\frac{\zeta}{\iota} = 0.2, \nu = 20$	1.00	16.13	31.45	47.59
$\frac{\zeta}{\iota} = 5, \nu = 20$	1.00	17.83	32.26	50.09
$\frac{\zeta}{\iota} = 0.2, \nu = 50$	1.00	16.66	31.66	48.32
$\frac{\zeta}{\iota} = 5, \nu = 50$	1.00	17.30	32.99	50.29
$\alpha_1 = 0$	0.92	20.34	41.07	61.41
PY: higher mean	1.00	18.25	32.70	50.95

Table 4.8: Performance of NBNR under different hyperparameter choices for the OCEAN data. The first column presents different combinations to check sensitivity. the second column presents ARI between optimal clusters obtained from each combination and the optimal clusters obtained by the original analysis of the OCEAN data.

the V/n ratio increases. Hence, we use lobe level network data for every individual rather than the ROI level data. This means that the network predictor corresponding to every individual is of dimension 12×12 in this analysis. As in Chapter 2, CCI is taken to be the response.

We start by fitting the NBNR model with $H = 20$ for this data. Figure 4.14 presents the posterior distribution of the number of clusters, as well as the uncertainty in estimating two subjects in the same cluster. Both Figures 4.14(a) and 4.14(b) indicate the existence of one cluster in the data. The optimal clustering configuration $\hat{z}$ estimated from (4.7) also includes all subjects in the same cluster. Given that the NBNR places all posterior probability on a single cluster, the model fitting statistics with the G, P and D values demonstrate marginally better



(a) $H = 20$

(b) $H = 20$

Figure 4.14: CCI Data: The left plot shows uncertainty in estimating the clusters. The plot on the right is a barplot for the posterior distribution of the estimated number of clusters. The inference is presented for $H = 20$.

performance of BNSP over NBNR (see Table 4.9).

Our analysis identifies 9 lobes to be influential to predict CCI, out of which five are in the left hemisphere and four in the right hemisphere (see table 4.10). These include the temporal, frontal, cingulate and occipital lobes in both hemispheres. As discussed in Chapter 2, there is considerable literature suggesting close association of creativity with the frontal and temporal lobes. Findings from this analysis also suggest the same. We also find 7 influential edges among all edges connecting between two influential nodes.

Methods	NBNR with $H = 20$			BNSP		
Comparison Metric	G	P	D	G	P	D
Values	60.84	77.01	137.86	56.21	78.18	134.39

Table 4.9: Model fitting statistics for NBNR and BNSP for the brain connectome CCI data application.

Left Hemisphere Lobes	Temporal, Cingulate, Frontal, Occipital, Parietal
Right Hemisphere Lobes	Temporal, Cingulate, Frontal, Occipital

Table 4.10: Brain regions (ROIs) detected as influential for the composite creativity index.

4.6 Summary

This chapter develops a Bayesian mixture model of network regressions. The proposed model allows groups of subjects sharing similar relationships between the scalar response and the network predictor. Unlike Chapter 2, the framework developed in this chapter is able to incorporate the neuroscientific phenomenon that different sets of individuals may have different relationships between brain lobes and a specific phenotype. Our proposed model also allows clustering of individuals into groups showing similar relationships between the phenotype and the brain connectome. Simulation studies and the brain connectome data analysis reveal superior performance of the proposed model over the BNSP model devised in Chapter 2.

Chapter 5

Conclusion

This dissertation develops novel regression frameworks for scalar responses and network predictors. Chapter 2 introduces a novel approach that develops a regression framework of a continuous phenotypic response on a brain network (represented in the form of a symmetric matrix). We propose a novel network shrinkage prior on the network predictor coefficient matrix. The proposed framework is able to identify nodes or functional regions in the brain network and interconnections between different regions, significantly related to the phenotypic response. To the best of our knowledge, our framework is the first principled Bayesian framework that enables identification of network nodes and edges significantly related to the response. The performance of the proposed model is evaluated with respect to a wide range of existing competitors available in the high dimensional frequentist and Bayesian literature using a variety of simulation studies. The proposed model identifies important brain regions and interconnections significantly associated with creativity for a group of subjects.

Next, in chapter 3 we extend our model to build network classifiers when a brain connectome network along with a binary response is provided for a group of individuals. Here we develop a broader class of global-local network shrinkage priors which includes the novel prior distribution specified earlier as a special case. We specifically consider two different global-local network shrinkage priors from this class of priors and investigate them using simulation studies. In particular, we assess their performance in terms of network classification and identifying influential network nodes and edges for the purpose of classification. We also demonstrate superior performance of our proposed network classifiers over state-of-the-art high dimensional classification techniques. Another major contribution remains developing theoretical conditions to guarantee asymptotically consistent classification for the proposed framework. In particular, we derive conditions on the number of network nodes, sparsity in the network coefficient matrix as a function of the sample size to achieve asymptotically optimal classification. While theoretical results on high dimensional binary regression with ordinary shrinkage priors have emerged recently, developing theory for our network classifier model involves several additional challenges due to the complex nature of the global local shrinkage prior developed here. The framework is used to classify individuals into high and low IQ groups based on their brain connectomes.

In chapter 4, we have developed a Bayesian network mixture regression model. The model allows the relationship between the scalar response and the network predictor to vary between groups of subjects. Simulation studies and analysis of the brain connectome dataset demonstrate superior performance of the proposed approach over the approach described in Chapter 2. Simulation studies are also used to evaluate the performance of the proposed approach by varying the true and fitted number of clusters, size of the network and sample size.

Chapter 6

Future Work

A number of future directions emerge from this work. The present framework develops network shrinkage priors to detect ineffective nodes and edges. Instead, one may cast this problem as a model selection problem in high dimensional network regression and develop non-local priors [78] for identifying influential nodes and edges. Another important direction appears to be the development of Gaussian process regression with the network as an input and the scalar response as the output. The problem is challenging since it requires developing a covariance kernel on network predictors which are not in the standard Euclidean space. One may also extend the current approaches to multivariate settings where, corresponding to each network predictor, there are multiple responses. For example, one may jointly model the big personality traits, such as *agreeableness*, *conscientiousness*, *openness*, *extraversion* and *neuroticism*, as a multivariate response and regress it on the brain network predictor for a subject. To elaborate, let $P_{i,1}, P_{i,2}, P_{i,3}, P_{i,4}$ and $P_{i,5}$ be the five personality traits with the corresponding network predictor A_i. One may consider employing a series of Bayesian network regression models with the network coefficient for the jth regression as B_j,

$$P_{i,j} = \mu_j + \langle A_i, B_j \rangle_F + \varepsilon_{i,j},\ i = 1, ..., n;\ j = 1, 2, 3, 4, 5.$$

The errors are correlated $\varepsilon_i = (\varepsilon_{i,1}, ..., \varepsilon_{i,5})' \sim N(0, \Sigma)$ and B_1, B_2, B_3, B_4 and B_5 are also modeled jointly to borrow information across different responses. Borrowing information may improve identification of influential nodes and edges.

Another important research direction we aim to pursue is to exploit the hierarchical structure of lobes and ROIs and develop multi-scale network regression models. To elaborate, we propose to develop a multi-scale network regression model

$$y_i = \mu + \langle A_i, B_1 \rangle_F + \langle R_i, B_2 \rangle_F + \varepsilon_i,\ \varepsilon_i \sim F,$$

where $F(\cdot)$ is some symmetric error distribution, A_i is the 68×68 network predictor matrix representing the number of neuron connections between 68 ROIs, and R_i is the 12×12 network predictor matrix representing the number of neuron connections between 12 lobes. The matrices B_1 and B_2 are the network predictor coefficients corresponding to A_i and R_i, respectively. We plan to develop network shrinkage priors on B_1 and B_2 in such a way that a-priori ensures all ROIs to be uninfluential if the lobe containing the ROI is uninfluential. Finally, we also propose to extend our theoretical results to general global-local shrinkage priors. Some of these constitute our present work.

Chapter 7

Appendix

7.1 Appendix A

This section shows the posterior propriety of the parameters in the BNR model. Without loss of generality, we set $\mu = 0$ while proving the posterior propriety. To begin with, we state a number of useful lemmas.

Preliminary Results

Lemma 7.1.1 *If C is an $h \times h$ non-negative definite matrix, then $|C+I| \geq 1$.*

Proof The eigenvalues of $(C+I)$ are given by $\varphi_1 + 1, ..., \varphi_h + 1$, where $\varphi_1, ..., \varphi_h$ are eigenvalues of C. Since C is non-negative definite, $\varphi_1 \geq 0, ..., \varphi_h \geq 0$. The result follows from the fact that $|C+I| = \prod_{l=1}^{h}(\varphi_l + 1)$ is the product of eigenvalues.

Lemma 7.1.2 *Let C be an $h \times h$ diagonal matrix with diagonal entries $c_1, ..., c_h$ all greater* *than* 0. *Suppose A is an $n \times h$ matrix with the largest eigenvalue of AA' given by $\mu_{AA'}$. Then $ACA' + I \leq \left(\mu_{AA'} \sum_{l=1}^{h} c_l + 1\right) I$, where $H_1 \leq H_2$ implies $H_2 - H_1$ is a positive definite matrix.*

Proof Since $c_1, ..., c_h > 0$, $ACA' \leq (\sum_{l=1}^{h} c_l) AA'$. Consider the spectral decomposition of the matrix AA'. Let the eigen-decomposition of $AA' = \Lambda H \Lambda'$, where Λ is the matrix of eigenvectors and H is a diagonal matrix with diagonal entries $\mu_1, ..., \mu_n$. Since each $\mu_i \leq \mu_{AA'}$, $AA' \leq \mu_{AA'} \Lambda \Lambda' = \mu_{AA'} I$. Thus, $ACA' \leq (\sum_{l=1}^{h} c_l) \mu_{AA'} I$. Hence $ACA' + I \leq \left(\mu_{AA'} \sum_{l=1}^{h} c_l + 1\right) I$.

Lemma 7.1.3 *Suppose z is an $h \times 1$ vector and A is an $h \times h$ symmetric positive definite matrix. Let B be another $h \times h$ positive definite matrix such that $A \geq B$ (where $A \geq B$ implies $A - B$ is non-negative definite). Then $z'A^{-1}z \leq z'B^{-1}z$.*

Proof $A \geq B$ implies $B^{-1/2} A B^{-1/2} \geq I$. Thus all eigenvalues of $B^{-1/2} A B^{-1/2} = B^{-1/2} A^{1/2} A^{1/2} B^{-1/2}$ are greater than or equal to 1. Since commuting the product of two matrices does not change the eigenvalues, $A^{1/2} B^{-1} A^{1/2}$ has all eigenvalues greater than or equal to 1. Thus $A^{1/2} B^{-1} A^{1/2} \geq I$, which implies $A^{-1} \leq B^{-1}$. Then $z'A^{-1}z \leq z'B^{-1}z$.

Main Result

Note that the posterior distribution of the parameters is given by

$$
\begin{aligned}
& p(\gamma, \tau^2, u_1, .., u_V, \xi_1, .., \xi_V, \lambda_1, .., \lambda_R, \theta^2, \Delta, \{s_{k,l}\}_{k<l}, \pi_1, ..., \pi_R, M \,|\, y, X) \\
& \qquad \propto \mathrm{N}(y | X\gamma, \tau^2 I) \times \mathrm{N}(\gamma | W, \tau^2 D) \times \frac{1}{\tau^2} \times \prod_{k=1}^{V} [\xi_k N(u_k \,|\, 0, M) + (1 - \xi_k)\delta_0] \\
& \qquad \times \prod_{k<l} Exp(s_{k,l} \,|\, \theta^2/2) \times Gamma(\theta^2 \,|\, \zeta, \iota) \times IW(M \,|\, S, \nu) \times Beta(\Delta \,|\, a_\Delta, b_\Delta) \\
& \qquad\qquad\qquad \times \prod_{r=1}^{R} [Ber(\lambda_r \,|\, \pi_r) \times Beta(\pi_r \,|\, 1, r^\eta)] \times \prod_{k=1}^{V} Ber(\xi_k \,|\, \Delta).
\end{aligned}
$$

Integrating over $\xi_1,...,\xi_V$

$$p(\gamma,\tau^2,u_1,..,u_V,\lambda_1,..,\lambda_R,\theta^2,\Delta,\{s_{k,l}\}_{k<l},\pi_1,...,\pi_R,M\,|\,y,X) \propto \mathrm{N}(y\,|\,X\gamma,\tau^2 I)\times$$
$$\mathrm{N}(\gamma|W,\tau^2 D)\times\frac{1}{\tau^2}\times\prod_{k=1}^{V}[\Delta N(u_k\,|\,0,M)+(1-\Delta)\delta_0]\times\prod_{k<l}Exp(s_{k,l}\,|\,\theta^2/2)\times$$
$$Gamma(\theta^2\,|\,\zeta,\iota)\times IW(M\,|\,S,\nu)\times Beta(\Delta\,|\,a_\Delta,b_\Delta)\times\prod_{r=1}^{R}[Ber(\lambda_r\,|\,\pi_r)\times Beta(\pi_r\,|\,1,r^\eta)]\,.$$

Further integrating over $\pi_1,...,\pi_R$ yields,

$$p(\gamma,\tau^2,u_1,..,u_V,\lambda_1,..,\lambda_R,\theta^2,\Delta,\{s_{k,l}\}_{k<l},M\,|\,y,X) \propto \mathrm{N}(y\,|\,X\gamma,\tau^2 I)\times\mathrm{N}(\gamma|W,\tau^2 D)\times$$
$$\frac{1}{\tau^2}\times\prod_{k=1}^{V}[\Delta N(u_k\,|\,0,M)+(1-\Delta)\delta_0]\times\prod_{k<l}Exp(s_{k,l}\,|\,\theta^2/2)\times Gamma(\theta^2\,|\,\zeta,\iota)\times$$
$$IW(M\,|\,S,\nu)\times Beta(\Delta\,|\,a_\Delta,b_\Delta)\times\prod_{r=1}^{R}\frac{\Gamma(\lambda_r+1)\Gamma(1-\lambda_r+r^\eta)\Gamma(r^\eta+1)}{\Gamma(r^\eta+2)\Gamma(r^\eta)}.$$

The prior specifications on Δ enable it to be bounded within a finite interval of (0,1). Thus in showing the posterior propriety of parameters with unbounded range, it is enough to treat Δ as constant. We treat it as fixed henceforth.

Note that each $\lambda_r\in\{0,1\}$, hence marginalizing out λ_r gives

$$p(\gamma,\Lambda,\tau^2,u_1,..,u_V,\theta^2,\{s_{k,l}\}_{k<l},M\,|\,y,X) \propto \sum_{\lambda_r\in\{0,1\}}\Big[\mathrm{N}(y\,|\,X\gamma,\tau^2 I)\times\mathrm{N}(\gamma|W,\tau^2 D)\times$$
$$\frac{1}{\tau^2}\times\prod_{k=1}^{V}[\Delta N(u_k\,|\,0,M)+(1-\Delta)\delta_0]\times\prod_{k<l}Exp(s_{k,l}\,|\,\theta^2/2)\times Gamma(\theta^2\,|\,\zeta,\iota)\times$$
$$IW(M\,|\,S,\nu)\times\prod_{r=1}^{R}\frac{\Gamma(\lambda_r+1)\Gamma(1-\lambda_r+r^\eta)\Gamma(r^\eta+1)}{\Gamma(r^\eta+2)\Gamma(r^\eta)}\Big].$$

Integrating over γ, we obtain,

$$p(u_1,..,u_V,\tau^2,\theta^2,\{s_{k,l}\}_{k<l},M\,|\,y,X) \propto \sum_{\lambda_r\in\{0,1\}}\Big[\frac{1}{(\tau^2)^{n/2+1}|XDX'+I|^{1/2}}\times$$
$$\exp\left\{-\frac{(y-XW)'(XDX'+I)^{-1}(y-XW)}{2\tau^2}\right\}\times\prod_{k=1}^{V}[\Delta N(u_k\,|\,0,M)+(1-\Delta)\delta_0]\times$$
$$\prod_{k<l}Exp(s_{k,l}\,|\,\theta^2/2)\times Gamma(\theta^2\,|\,\zeta,\iota)\times IW(M\,|\,S,\nu)\times$$
$$\prod_{r=1}^{R}\frac{\Gamma(\lambda_r+1)\Gamma(1-\lambda_r+r^{\eta})\Gamma(r^{\eta}+1)}{\Gamma(r^{\eta}+2)\Gamma(r^{\eta})}\Big].$$

Next, we integrate w.r.t. θ^2 to obtain

$$p(u_1,..,u_V,\tau^2,\{s_{k,l}\}_{k<l},M\,|\,y,X) \propto \sum_{\lambda_r\in\{0,1\}}\Big[\frac{1}{(\tau^2)^{n/2+1}|XDX'+I|^{1/2}}\times$$
$$\exp\left\{-\frac{(y-XW)'(XDX'+I)^{-1}(y-XW)}{2\tau^2}\right\}\times\prod_{k=1}^{V}[\Delta N(u_k\,|\,0,M)+(1-\Delta)\delta_0]\times$$
$$\frac{1}{(\iota+\sum_{k<l}s_{k,l})^{q+\zeta}}\times IW(M\,|\,S,\nu)\times\prod_{r=1}^{R}\frac{\Gamma(\lambda_r+1)\Gamma(1-\lambda_r+r^{\eta})\Gamma(r^{\eta}+1)}{\Gamma(r^{\eta}+2)\Gamma(r^{\eta})}\Big]. \tag{7.1}$$

(7.1) is a discrete sum of 2^R terms with different combinations of $\lambda_1,...,\lambda_r$. The sum integrated out over all the parameters is finite if the individual summands are finite when integrated out w.r.t all parameters.

Denote a representative summand by $p^*(u_1,..,u_V,\tau^2,\{s_{k,l}\}_{k<l},M\,|\,y,X)$, where

$$p^*(u_1,..,u_V,\tau^2,\{s_{k,l}\}_{k<l},M\,|\,y,X) \propto \frac{1}{(\tau^2)^{n/2+1}|XDX'+I|^{1/2}}\times$$
$$\exp\left\{-\frac{(y-XW)'(XDX'+I)^{-1}(y-XW)}{2\tau^2}\right\}\times\prod_{k=1}^{V}[\Delta N(u_k\,|\,0,M)+(1-\Delta)\delta_0]\times$$
$$\frac{1}{(\iota+\sum_{k<l}s_{k,l})^{q+\zeta}}\times IW(M\,|\,S,\nu)\times\prod_{r=1}^{R}\frac{\Gamma(\lambda_r+1)\Gamma(1-\lambda_r+r^{\eta})\Gamma(r^{\eta}+1)}{\Gamma(r^{\eta}+2)\Gamma(r^{\eta})}.$$

Note the fact that D is a diagonal matrix with all positive diagonal entries. Thus XDX'

is non-negative definite and by using Lemma 7.1.2

$$XDX' + I \leq XX' \sum_{k<l} s_{k,l} + I \leq \left(\mu_{XX'} \sum_{k<l} s_{k,l} + 1 \right) I,$$

where $A \leq B$ implies $A - B$ is a non-negative definite matrix and $\mu_{XX'}$ is the largest eigenvalue of XX'. Using Lemma 7.1.3, the above inequality implies

$$(y - XW)'(XDX' + I)^{-1}(y - XW) \geq \frac{||y - XW||^2}{\mu_{XX'} \sum_{k<l} s_{k,l} + 1}.$$

Let

$$\begin{aligned} \tilde{p}(u_1, .., u_V, \tau^2, \{s_{k,l}\}_{k<l}, M) = & \frac{1}{(\tau^2)^{n/2+1} |XDX' + I|^{1/2}} \times \\ & \exp\left\{ -\frac{(y - XW)'(XDX' + I)^{-1}(y - XW)}{2\tau^2} \right\} \times \prod_{k=1}^{V} N(u_k \,|\, 0, M) \times \\ & \frac{1}{(\iota + \sum\limits_{k<l} s_{k,l})^{q+\zeta}} \times IW(M \,|\, S, \nu) \times \prod_{r=1}^{R} \frac{\Gamma(\lambda_r + 1)\Gamma(1 - \lambda_r + r^{\eta})\Gamma(r^{\eta} + 1)}{\Gamma(r^{\eta} + 2)\Gamma(r^{\eta})}. \end{aligned} \tag{7.2}$$

With little algebra it can be shown that

$$\begin{aligned} & p^*(u_1, .., u_V, \tau^2, \{s_{k,l}\}_{k<l}, M \,|\, y, X) \\ & = \text{constant} \times \sum_{1 \leq j_1, \ldots, j_l \leq V, 0 \leq l \leq V} \Delta^l (1 - \Delta)^{V-l} \tilde{p}(u_{j_1}, .., u_{j_l}, u_{j_{l+1}} = 0, .., u_{j_V} = 0, \tau^2, \{s_{k,l}\}_{k<l}, M). \end{aligned}$$

Therefore, the integral of (7.1) w.r.t. all parameters is finite if and only if

$$\int \tilde{p}(u_1, .., u_V, \tau^2, \{s_{k,l}\}_{k<l}, M) du_1 \cdots du_V d\tau^2 dM d \prod_{k<l} s_{k,l} < \infty.$$

Henceforth, we will proceed to show that this integral is finite.

With little algebra, we have that

$$\int IW(M|\nu, S) \prod_{k=1}^{V} N(u_k \,|\, 0, M) dM \propto \frac{1}{|S + \sum_{k=1}^{V} u_k u_k'|^{(\nu+V)/2}}.$$

Hence,

$$
\begin{aligned}
\tilde{p}(u_1,..,u_V,\tau^2,\{s_{k,l}\}_{k<l}) \le \text{constant} \times \frac{1}{|S+\sum_{k=1}^{V} u_k u_k'|^{(\nu+V)/2}} \frac{1}{(\tau^2)^{n/2+1}} \times \\
\exp\left\{-\frac{||y-XW||^2}{2\tau^2(\mu_{XX'}\sum_{k<l}s_{k,l}+1)}\right\} \times \frac{1}{(\iota+\sum\limits_{k<l}s_{k,l})^{q+\zeta}} \frac{1}{|XDX'+I|^{1/2}} \times \\
\prod_{r=1}^{R} \frac{\Gamma(\lambda_r+1)\Gamma(1-\lambda_r+r^{\eta})\Gamma(r^{\eta}+1)}{\Gamma(r^{\eta}+2)\Gamma(r^{\eta})}.
\end{aligned}
$$

Define $\mathcal{A} = \{(u_1,...,u_V) : ||y-XW||^2 > 1\}$. Then

$$
\begin{aligned}
\int \tilde{p}(u_1,..,u_V,\tau^2,\{s_{k,l}\}_{k<l}) du_1 \cdots du_V d\tau^2 d\prod_{k<l} s_{k,l} \\
= \int_{\mathcal{A}} \tilde{p}(u_1,..,u_V,\tau^2,\{s_{k,l}\}_{k<l}) du_1 \cdots du_V d\tau^2 d\prod_{k<l} s_{k,l} + \\
\int_{\mathcal{A}^c} \tilde{p}(u_1,..,u_V,\tau^2,\{s_{k,l}\}_{k<l}) du_1 \cdots du_V d\tau^2 d\prod_{k<l} s_{k,l}.
\end{aligned}
$$

Now,

$$
\begin{aligned}
\int_{\mathcal{A}} \tilde{p}(u_1,..,u_V,\tau^2,\{s_{k,l}\}_{k<l}) d\tau^2 d\prod_{k<l} s_{k,l} du_1 \cdots du_V \le \text{constant} \int_{\mathcal{A}} \frac{du_1 \cdots du_V}{|S+\sum_{k=1}^{V} u_k u_k'|^{(\nu+V)/2}} \times \\
\int \frac{1}{(\tau^2)^{n/2+1}} \exp\left\{-\frac{1}{2\tau^2(\mu_{XX'}\sum_{k<l}s_{k,l}+1)}\right\} \times \frac{1}{(\iota+\sum\limits_{k<l}s_{k,l})^{q+\zeta}} \frac{d\tau^2 d\prod\limits_{k<l} s_{k,l}}{|XDX'+I|^{1/2}} \times \\
\prod_{r=1}^{R} \frac{\Gamma(\lambda_r+1)\Gamma(1-\lambda_r+r^{\eta})\Gamma(r^{\eta}+1)}{\Gamma(r^{\eta}+2)\Gamma(r^{\eta})} \\
\le \text{constant} \left\{ \int_{\mathcal{A}} \frac{1}{|S+\sum_{k=1}^{V} u_k u_k'|^{(\nu+V)/2}} du_1 \cdots du_V \right\} \times \\
\left\{ \int \frac{(\mu_{XX'}\sum_{k<l}s_{k,l}+1)^{n/2}}{|XDX'+I|^{1/2}(\iota+\sum\limits_{k<l}s_{k,l})^{q+\zeta}} d\prod_{k<l} s_{k,l} \right\} \times \prod_{r=1}^{R} \frac{\Gamma(\lambda_r+1)\Gamma(1-\lambda_r+r^{\eta})\Gamma(r^{\eta}+1)}{\Gamma(r^{\eta}+2)\Gamma(r^{\eta})}.
\end{aligned}
$$

Note that

$$\int\limits_{\mathcal{A}} \frac{1}{|S+\sum_{k=1}^{V} u_k u_k'|^{(\nu+V)/2}} du_1 \cdots du_V \leq \int\limits_{\mathcal{A}} \frac{1}{\prod_{k=1}^{V} |S+u_k u_k'|^{\nu/2V+1/2}} du_1 \cdots du_V$$

$$\leq \prod_{k=1}^{V} \left(\int\limits_{\mathcal{A}} \frac{1}{|S+u_k u_k'|^{\nu/V+1}} du_k \right)^{1/2},$$

where the first inequality follows from the fact that $|S+\sum_{k=1}^{V} u_k u_k'| \geq |S+u_k u_k'|$ for all k. The second inequality is a direct application of the Cauchy-Schwarz inequality. By the ratio test of integrals, this integral is finite if $\int \frac{1}{[(1+u_{k,1})^2 \cdots (1+u_{k,R})^2]^{\nu/V+1}} du_k$ is finite. Now use the fact that $\int \frac{1}{x^{1+c}} dx < \infty$ for any $c > 0$ to argue that $\int \frac{1}{[(1+u_{k,1})^2 \cdots (1+u_{k,R})^2]^{2\nu/V+1}} du_k$ is finite.

Similarly,
$\left\{ \int \frac{(\mu_{XX'} \sum_{k<l} s_{k,l}+1)^{n/2}}{|XDX'+I|^{1/2}(\iota+\sum\limits_{k<l} s_{k,l})^{q+\zeta}} d\prod_{k<l} s_{k,l} \right\} \leq \left\{ \int \frac{(\mu_{XX'} \sum_{k<l} s_{k,l}+1)^{n/2}}{(\mu_{XX',min} \min_{k<l} s_{k,l}+1)^{n/2}(\iota+\sum\limits_{k<l} s_{k,l})^{q+\zeta}} d\prod_{k<l} s_{k,l} \right\}$, where $\mu_{XX',min}$ is the minimum eigenvalue of XX'. The last inequality follows from the fact that $XX' \geq \mu_{XX',min} \min_{k<l} s_{k,l} I$. $\left\{ \int \frac{(\mu_{XX'} \sum_{k<l} s_{k,l}+1)^{n/2}}{(\mu_{XX',min} \min_{k<l} s_{k,l}+1)^{n/2}(\iota+\sum\limits_{k<l} s_{k,l})^{q+\zeta}} d\prod_{k<l} s_{k,l} \right\}$ is finite if and only if $\left\{ \int \frac{(\mu_{XX'} \sum_{k<l} s_{k,l}+1)^{n/2}}{(\mu_{XX',min} \sum\limits_{k<l} s_{k,l}+1)^{n/2}(\iota+\sum\limits_{k<l} s_{k,l})^{q+\zeta}} d\prod_{k<l} s_{k,l} \right\} < \infty$, by ratio test of integrals. Since the latter integral is finite, $\int\limits_{\mathcal{A}} \tilde{p}(u_1,..,u_V,\tau^2,\{s_{k,l}\}_{k<l}) du_1 \cdots du_V d\tau^2 d\prod\limits_{k<l} s_{k,l} \leq \infty$.

Now consider the expression $\int\limits_{\mathcal{A}^c} \tilde{p}(u_1,..,u_V,\tau^2,\{s_{k,l}\}_{k<l}) d\tau^2 d\prod\limits_{k<l} s_{k,l} du_1 \cdots du_V$. It is easy to see that $\mathcal{A}^c = \{(u_1,...,u_V) : ||y-XW||^2 \leq 1\}$ is a bounded set, so that the bounded function

$\exp\left\{-\frac{||y-XW||^2}{2\tau^2(\mu_{XX'}\sum_{k<l}s_{k,l}+1)}\right\}$ achieves the maximum value at $W=W^*$. Thus,

$$\begin{aligned}
&\int_{\mathcal{A}^c}\tilde{p}(u_1,..,u_V,\tau^2,\{s_{k,l}\}_{k<l})du_1\cdots du_V d\tau^2 d\prod_{k<l}s_{k,l}\leq \text{constant}\int_{\mathcal{A}^c}\frac{1}{|S+\sum_{k=1}^{V}u_ku_k'|^{(\nu+V)/2}}\times\\
&\qquad\exp\left\{-\frac{||y-XW^*||^2}{2\tau^2(\mu_{XX'}\sum_{k<l}s_{k,l}+1)}\right\}\times\int\frac{1}{(\iota+\sum\limits_{k<l}s_{k,l})^{q+\zeta}}\frac{d\tau^2 d\prod\limits_{k<l}s_{k,l}}{(\tau^2)^{n/2+1}}\frac{1}{|XDX'+I|^{1/2}}\\
&\leq\frac{\text{constant}}{||y-XW^*||^n}\left\{\int_{\mathcal{A}^c}\frac{1}{|S+\sum_{k=1}^{V}u_ku_k'|^{(\nu+V)/2}}du_1\cdots du_V\right\}\times\\
&\qquad\left\{\int\frac{(\mu_{XX'}\sum_{k<l}s_{k,l}+1)^{n/2}}{|XDX'+I|^{1/2}(\iota+\sum\limits_{k<l}s_{k,l})^{q+\zeta}}d\prod_{k<l}s_{k,l}\right\}<\infty,
\end{aligned}$$

where the last step follows from earlier discussions.

7.2 Appendix B

This section provides details of posterior computation for all the parameters in the Bayesian network regression with a continuous response.

Let $x_i=(a_{i,1,2},a_{i,1,3},...,a_{i,1,V},a_{i,2,3},a_{i,2,4},...,a_{i,2,V},....,a_{i,V-1,V})'$ be of dimension $q\times 1$, where $q=\frac{V\times(V-1)}{2}$. Assume $y=(y_1,...,y_n)'\in\mathbb{R}^n$ and $X=(x_1:\cdots:x_n)'$ is an $n\times q$ matrix. Further, assume $W=(u_1'\Lambda u_2,...,u_1'\Lambda u_V,....,u_{V-1}'\Lambda u_V)'$, $D=diag(s_{1,2},...,s_{V-1,V})$ and $\gamma=(\gamma_{1,2},...,\gamma_{V-1,V})'$. Thus, with n data points, the hierarchical model with the Bayesian Network Lasso prior can be written as

$$y \sim \mathrm{N}(\mu + X\gamma, \tau^2 I)$$

$$\gamma \sim \mathrm{N}(W, \tau^2 D),\ (\mu, \tau^2) \sim \pi(\mu, \tau^2) \propto \frac{1}{\tau^2},\ u_k|\xi_k = 1 \sim N(u_k\,|\,0, M),\, u_k|\xi_k = 0 \sim \delta_0,\, \mu \sim flat()$$

$$s_{k,l} \sim Exp(\theta^2/2),\ \ \theta^2 \sim Gamma(\zeta, \iota),\ M \sim IW(S, \nu),\ \Delta \sim Beta(a_\Delta, b_\Delta), \xi_k \sim Ber(\Delta)$$

$$\lambda_r \sim Ber(\pi_r),\ \pi_r \sim Beta(1, r^\eta),\, \eta > 1.$$

The hierarchical model specified above leads to straightforward Gibbs sampling with full conditionals obtained as following:

- $\mu\,|- \sim N\left(\frac{1'(y-X\gamma)}{n}, \frac{\tau^2}{n}\right)$

- $\gamma\,|- \sim N(\mu_{\gamma|\cdot}, \Sigma_{\gamma|\cdot})$, where $\mu_{\gamma|\cdot} = (X'X + D^{-1})^{-1}(X'(y-\mu 1) + D^{-1}W)$ and $\Sigma_{\gamma|\cdot} = \tau^2 (X'X + D^{-1})^{-1}$

- $\tau^2\,|- \sim IG\left[(\frac{n}{2} + \frac{V(V-1)}{4}), \frac{(y-\mu 1 - X\gamma)'(y-\mu 1 - X\gamma) + (\gamma - W)' D^{-1}(\gamma - W)}{2}\right]$

- $s_{k,l}\,|- \sim GIG\left[\frac{1}{2}, \frac{(\gamma_{k,l} - u_k' \Lambda u_l)^2}{\tau^2}, \theta^2\right]$, where GIG denotes the generalized inverse Gaussian distribution.

- $\theta^2\,|- \sim Gamma\left[\left(\zeta + \frac{V(V-1)}{2}\right), \left(\iota + \sum_{k<l} \frac{s_{k,l}}{2}\right)\right]$

- $u_k\,|- \sim w_{u_k}\,\delta_0(u_k) + (1 - w_{u_k})\,N(u_k\,|\,m_{u_k}, \Sigma_{u_k})$, where $U_k^* = (u_1 : \cdots : u_{k-1} : u_{k+1} : \cdots : u_V)'\Lambda,\ H_k = diag(s_{1,k}, ..., s_{k-1,k}, s_{k,k+1}, ..., s_{k,V}),\ \gamma_k = (\gamma_{1,k}, ..., \gamma_{k-1,k}, \gamma_{k,k+1}, ..., \gamma_{k,V})$, and

$$\Sigma_{u_k} = \left(U_h^{*'} H_k^{-1} U_k^* / \tau^2 + M^{-1}\right)^{-1},\ m_{u_k} = \Sigma_{u_k} U_k^{*'} H_k^{-1} \gamma_k / \tau^2$$

$$w_{u_k} = \frac{(1-\Delta) N(\gamma_k\,|\,0, \tau^2 H_k)}{(1-\Delta) N(\gamma_k\,|\,0, \tau^2 H_k) + \Delta N(\gamma_k\,|\,0, \tau^2 H_k + U_k^* M U_k^{*'})}$$

- $\xi_k | - \sim Ber(1 - w_{u_k})$

- $\Delta | - \sim Beta\left[(a_\Delta + \sum_{k=1}^{V} \xi_k), (b_\Delta + \sum_{k=1}^{V}(1-\xi_k))\right]$.

- $M | - \sim IW[(S + \sum_{k:u_k \neq 0} u_k \Lambda u_k'), (\nu + \{\#k : u_k \neq 0\})]$.

- $\lambda_r | - \sim Ber(p_{\lambda_r})$, where $p_{\lambda_r} = \frac{\pi_r N(\gamma | W_1, \tau^2 D)}{\pi_r N(\gamma | W_1, \tau^2 D) + (1-\pi_r) N(\gamma | W_0, \tau^2 D)}$. Here $W_1 = (u_1' \Lambda_1 u_2, ..., u_1' \Lambda_1 u_V, ..., u_{V-1}' \Lambda_1 u_V)'$, $W_0 = (u_1' \Lambda_0 u_2, ..., u_1' \Lambda_0 u_V, ..., u_{V-1}' \Lambda_0 u_V)'$, $\Lambda_1 = diag(\lambda_1, .., \lambda_{r-1}, 1, \lambda_{r+1}, .., \lambda_R)$, $\Lambda_0 = diag(\lambda_1, .., \lambda_{r-1}, 0, \lambda_{r+1}, .., \lambda_R)$, for $r = 1, .., R$.

- $\pi_r | - \sim Beta(\lambda_r + 1, 1 - \lambda_r + r^\eta)$, for $r = 1, .., R$.

As noted in Section 2.3 of the main text, naively sampling from the full conditional of γ above faces substantial computational difficulties. We now state Lemma 7.2.1 that provides a computational strategy to draw posterior samples of γ efficiently. Proof of Lemma 7.2.1 is given below.

Lemma 7.2.1 *Let γ_W be a random variable such that*

$$\gamma_W | - \sim \mathrm{N}\left[(D^{-1} + X^T X)^{-1} X^T (y - \mu \underline{1} - XW), \tau^2 (D^{-1} + X^T X)^{-1}\right]. \tag{7.3}$$

Then the following results hold.

(a) $\gamma \overset{D}{=} \gamma_W + W$

(b) Let, $\Delta_{\gamma_1} \sim \mathrm{N}(\mathbf{0}, \tau^2 D)$, $\Delta_{\gamma_2} \sim \mathrm{N}(\mathbf{0}, I)$, $\Delta_{\gamma_3} = \frac{X}{\tau} \Delta_{\gamma_1} + \Delta_{\gamma_2}$,

$\gamma_W = \Delta_{\gamma_1} + (\tau^2 D) \frac{X^T}{\tau} (X D X^T + I)^{-1} \left[\frac{(y - \mu \underline{1} - XW)}{\tau} - \Delta_{\gamma_3}\right]$.

Remark: This algorithm ensures that samples from the posterior full conditionals of γ can be obtained by sampling from the posterior full conditionals of γ_W. Lemma 7.2.1 shows that obtaining samples from the full conditional of γ_W only requires inverting an $n \times n$ matrix. Assuming $n << q$, which is typically encountered in the real data applications, the computational complexity of the proposed approach is substantially mitigated.

Proof of Lemma 7.2.1

(a) Note that

$$\begin{aligned}
E(\gamma_W + W) &= W + (D^{-1} + X^T X)^{-1} X^T (y - \mu \underline{1} - XW) \\
&= W - (D^{-1} + X^T X)^{-1} X^T XW + (D^{-1} + X^T X)^{-1} X^T (y - \mu \underline{1}) \\
&= W - (D^{-1} + X^T X)^{-1} (D^{-1} + X^T X - D^{-1}) W + (D^{-1} + X^T X)^{-1} X^T (y - \mu \underline{1}) \\
&= W - (I - (D^{-1} + X^T X)^{-1} D^{-1}) W + (D^{-1} + X^T X)^{-1} X^T (y - \mu \underline{1}) \\
&= (D^{-1} + X^T X)^{-1} D^{-1} W + (D^{-1} + X^T X)^{-1} X^T (y - \mu \underline{1}) \\
&= (D^{-1} + X^T X)^{-1} (D^{-1} W + X^T (y - \mu \underline{1})) = E(\gamma).
\end{aligned}$$

Also note that $Var(\gamma_W + W) = Var(\gamma)$ trivially since W is a given in the Gibbs step.

(b) Note that

$$\begin{aligned}
E(\gamma_W) &= E\left(\Delta_{\gamma_1} + (\tau^2 D)\frac{X^T}{\tau}(XDX^T + I)^{-1}\left[\frac{(y - \mu \underline{1} - XW)}{\tau} - \Delta_{\gamma_3}\right]\right) \\
&= 0 + (\tau^2 D)\frac{X^T}{\tau}(XDX^T + I)^{-1}\left[\frac{(y - \mu \underline{1} - XW)}{\tau} - 0\right] \\
&= DX^T (XDX^T + I)^{-1} (y - \mu \underline{1} - XW).
\end{aligned}$$

Using the Sherman-Morrison-Woodbury matrix identity, we have that $(D^{-1} + X^T X)^{-1} = D - DX^T (XDX^T + I)^{-1} XD$.

Hence

$$E\left(\Delta_{\gamma_1}+(\tau^2 D)\frac{X^T}{\tau}(XDX^T+I)^{-1}\left[\frac{(y-\mu\underline{1}-XW)}{\tau}-\Delta_{\gamma_3}\right]\right)$$

$$=DX^T(XDX^T+I)^{-1}(y-\mu\underline{1}-XW)$$

$$=(DX^T-DX^T+DX^T(XDX^T+I)^{-1})(y-\mu\underline{1}-XW)$$

$$=(D-DX^T(XDX^T+I)^{-1}XD)X^T(y-\mu\underline{1}-XW)$$

$$=(D^{-1}+X^TX)^{-1}X^T(y-\mu\underline{1}-XW)$$

$$=E(\gamma_W),$$

where the last step follows from equation (5) in the main text under Lemma 2.1.

Using the fact that $Var(\Delta_{\gamma_1})=\tau^2 D, Var(\Delta_{\gamma_2})=I, Var(\Delta_{\gamma_3})=(XDX^T+I)$ and $Cov(\Delta_{\gamma_1},\Delta_{\gamma_3})=\tau XD$, we have

$$Var\left(\Delta_{\gamma_1}+(\tau^2 D)\frac{X^T}{\tau}(XDX^T+I)^{-1}\left[\frac{(y-\mu\underline{1}-XW)}{\tau}-\Delta_{\gamma_3}\right]\right)$$

$$=Var(\Delta_{\gamma_1})+\tau DX^T(XDX^T+I)^{-1}Var(\Delta_{\gamma_3})(\tau DX^T(XDX^T+I)^{-1})^T$$

$$+\tau DX^T(XDX^T+I)^{-1}Cov(\Delta_{\gamma_1},\Delta_{\gamma_3})$$

$$=\tau^2 D+\tau^2 DX^T(XDX^T+I)^{-1}(XDX^T+I)[DX^T(XDX^T+I)^{-1}]^T$$

$$-2\tau^2 DX^T(XDX^T+I)^{-1}$$

$$=\tau^2 D+\tau^2 DX^T(XDX^T+I)^{-1}(XDX^T+I)[DX^T(XDX^T+I)^{-1}]^T$$

$$-2\tau^2 DX^T(XDX^T+I)^{-1}$$

$$=\tau^2\left[D-DX^T(XDX^T+I)^{-1}XD\right]=\tau^2(D^{-1}+X^TX)^{-1}=Var(\gamma_W),$$

where the last step follows from equation (5) in the main text under Lemma 2.1.

7.3 Appendix C

In this section, we describe the procedure for edge selection in our model, taking into account multiplicity correction. It is well acknowledged that the problem of selecting important coefficients is a challenging task when γ is assigned a continuous shrinkage prior, since none of the coefficients is zero in any MCMC iteration. Recently, [92] proposed an approach that aims to address the problem of identifying influential edge coefficients through a novel method of post processing of posterior samples. The approach is based on first obtaining a posterior distribution of the number of signals by clustering the signal and the noise coefficients and then estimating the signals from the posterior median. While [92] addresses the problem of variable selection from posterior samples of coefficients, the procedure does not necessarily address the problem of multiple comparisons.

Here we propose a novel procedure that is inspired by [92] that explicitly allows to generate accurate estimates of the false discovery rate (FDR) associated with the procedure. Our approach also relies on fitting a mixture model to the logarithm of the absolute value of the point estimates of the coefficients using an Expectation-Maximization algorithm, but one more flexible than the one implicitly used in [92]. The probability that each coefficient is generated by the mixture component with the lowest mean (which is a natural byproduct of the EM algorithm) provides an estimate of the local FDR associated with that coefficient [101], from which an estimate of the FDR curve can be easily generated. The details of the algorithm are as follows:

1. Obtain posterior mean of all edge coefficients from post burn-in MCMC samples.

2. Cluster the logarithm of absolute values of the posterior mean of coefficients into two groups using a either a two-component mixture of Gaussian distributions or a two component mixture of skewed t-distributions. We use the `R` library `mclust` for when we use Gaussian mixture [48], and library `EMMIXcskew` while using mixture of skewed t-distribution.

3. Using the probability that each coefficient is generated by the mixture component with the lowest mean as an estimate of the local FDR, compute for every H the FDR associated with the H largest coefficients, $FDR(H)$, as the sum of their local FDR values divided by H.

4. Given a value α of the FDR that we are aiming to control for (say, for example, 0.05), pick as significant the H^* largest coefficients, where H^* is the largest value of H such that $FDR(H) \leq \alpha$

7.4 Appendix D

This section provides full conditionals for all the parameters in the Bayesian binary network regression with network lasso shrinkage prior on γ described in Chapter 3. Assume $W = (u_1'\Lambda u_2, ..., u_1'\Lambda u_V, ..., u_{V-1}'\Lambda u_V)'$, $D = diag(s_{1,2}^2, ..., s_{V-1,V}^2)$ and $\gamma = (\gamma_{1,2}, ..., \gamma_{V-1,V})'$. Thus, with n data points, the hierarchical model with the *network lasso prior* in the binary setting can be written as

$$t \sim \mathrm{N}(\mu + X\gamma, \Omega^{-1})$$

$$\gamma \sim \mathrm{N}(W, D),\ u_k|\xi_k = 1 \sim N(u_k\,|\,0, Q),\ u_k|\xi_k = 0 \sim \delta_0,\ \xi_k \sim Ber(\Delta),\ \mu \sim flat()$$

$$s^2_{k,l} \sim Exp(\theta^2/2),\ \ \theta^2 \sim Gamma(\zeta, \iota),\ \ Q \sim IW(\nu, I),\ \ \Delta \sim Beta(a_\Delta, b_\Delta)$$

$$p(\omega_i) \sim PG(1,0),\ \lambda_r \sim Ber(\pi_r),\ \pi_r \sim Beta(1, r^\eta),\ \eta > 1.$$

The full conditional distributions of the model parameters are given below.

- $\mu\,|-\sim N\left(\frac{1'\Omega(t-X\gamma)}{1'\Omega 1}, \frac{1}{1'\Omega 1}\right)$

- $\gamma\,|-\sim N(\mu_{\gamma|\cdot}, \Sigma_{\gamma|\cdot})$, where $\mu_{\gamma|\cdot} = (X'\Omega X + D^{-1})^{-1}(X'\Omega(t-\mu 1) + D^{-1}W)$ and $\Sigma_{\gamma|\cdot} = (X'\Omega X + D^{-1})^{-1}$

- $s^2_{k,l}\,|-\sim GIG\left[\frac{1}{2}, (\gamma_{k,l} - u_k'\Lambda u_l)^2, \theta^2\right]$, where GIG denotes the generalized inverse Gaussian distribution.

- $\theta^2\,|-\sim Gamma\left[\left(\zeta + \frac{V(V-1)}{2}\right), \left(\iota + \sum_{k<l}\frac{s^2_{k,l}}{2}\right)\right]$

- $u_k\,|-\sim w_{u_k}\,\delta_0(u_k) + (1-w_{u_k})\,N(u_k\,|\,m_{u_k}, \Sigma_{u_k})$, where $U_k^* = (u_1 : \cdots : u_{k-1} : u_{k+1} : \cdots : u_V)'\Lambda$, $H_k = diag(s^2_{1,k}, ..., s^2_{k-1,k}, s^2_{k,k+1}, ..., s^2_{k,V})$, $\gamma_k = (\gamma_{1,k}, ..., \gamma_{k-1,k}, \gamma_{k,k+1}, ..., \gamma_{k,V})$, and

$$\Sigma_{u_k} = \left(U_h^{*'}H_k^{-1}U_k^* + Q^{-1}\right)^{-1},\ m_{u_k} = \Sigma_{u_k}U_k^{*'}H_k^{-1}\gamma_k$$
$$w_{u_k} = \frac{(1-\Delta)N(\gamma_k\,|\,0, H_k)}{(1-\Delta)N(\gamma_k\,|\,0, H_k) + \Delta N(\gamma_k\,|\,0, H_k + U_k^*QU_k^{*'})}$$

- $\xi_k|-\sim Ber(1-w_{u_k})$

- $\Delta\,|-\sim Beta\left[(a_\Delta + \sum_{k=1}^V \xi_k), (b_\Delta + \sum_{k=1}^V (1-\xi_k))\right]$.

- $Q\,|-\sim IW[(\nu+\{\#k: u_k\neq 0\}),(I+\sum_{k:u_k\neq 0}u_k\Lambda u_k')]$.

- $\lambda_r\,|-\sim Ber(p_{\lambda_r})$, where $p_{\lambda_r}=\frac{\pi_r N(\gamma|W_1,D)}{\pi_r N(\gamma|W_1,D)+(1-\pi_r)N(\gamma|W_0,D)}$. Here

  $W_1=(u_1'\Lambda_1 u_2,...,u_1'\Lambda_1 u_V,....,u_{V-1}'\Lambda_1 u_V)'$, $W_0=(u_1'\Lambda_0 u_2,...,u_1'\Lambda_0 u_V,....,u_{V-1}'\Lambda_0 u_V)'$,

  $\Lambda_1=diag(\lambda_1,..,\lambda_{r-1},1,\lambda_{r+1},..,\lambda_R)$, $\Lambda_0=diag(\lambda_1,..,\lambda_{r-1},0,\lambda_{r+1},..,\lambda_R)$, for $r=1,..,R$.

- $\pi_r\,|-\sim Beta(\lambda_r+1,1-\lambda_r+r^{\eta})$, for $r=1,..,R$.

  Using the relationship, $PG(x\,|\,b,c)\propto \exp(-\frac{c^2x}{2})PG(x\,|\,1,0)$ [111], we obtain

- $\omega_i\,|-\sim PG(1,\mu+x_i'\gamma)$, for $i=1,..,n$.

7.5 Appendix E

This section provides full conditionals for all the parameters in the Bayesian network classifier model introduced in Chapter 3 with Bayesian network horseshoe prior. Assume $W=(u_1'\Lambda u_2,...,u_1'\Lambda u_V,....,u_{V-1}'\Lambda u_V)'$, $D=diag(\sigma^2 s_{1,2}^2,...,\sigma^2 s_{V-1,V}^2)$ and $\gamma=(\gamma_{1,2},...,\gamma_{V-1,V})'$. Thus, with n data points, the hierarchical model with the network horseshoe prior in the binary setting can be written as

$$t\sim \mathrm{N}(\mu+X\gamma,\Omega^{-1})$$

$$\gamma\sim \mathrm{N}(W,D),\ u_k|\xi_k=1\sim N(u_k\,|\,0,Q),\ u_k|\xi_k=0\sim\delta_0,\ \xi_k\sim Ber(\Delta),\ \mu\sim flat()$$

$$s_{k,l}\sim C^+(0,1),\ \ \sigma\sim C^+(0,1),\ \ Q\sim IW(\nu,I),\ \ \Delta\sim Beta(a_\Delta,b_\Delta)$$

$$p(\omega_i)\sim PG(1,0),\ \lambda_r\sim Ber(\pi_r),\ \pi_r\sim Beta(1,r^{\eta}),\ \eta>1.$$

Note that, following [97],

$$s_{k,l} \sim C^{+}(0,1), \;\; \sigma \sim C^{+}(0,1)$$

can be written in an augmented form as

$$s^2_{k,l}\,|\,\nu_{k,l} \sim IG\left(\frac{1}{2},\frac{1}{\nu_{k,l}}\right), \;\; \nu_{k,l} \sim IG\left(\frac{1}{2},1\right), \;\; \sigma^2\,|\,\sigma_2 \sim IG\left(\frac{1}{2},\frac{1}{\sigma_2}\right), \;\; \sigma_2 \sim IG\left(\frac{1}{2},1\right).$$

With the model formulation described above, the full conditional distributions of the model parameters are given by the following distributions:

- $\mu\,|\,- \sim N\left(\frac{1'\Omega(t-X\gamma)}{1'\Omega 1}, \frac{1}{1'\Omega 1}\right)$

- $\gamma | - \sim N(\mu_{\gamma|.}, \Sigma_{\gamma|.})$, where $\mu_{\gamma|.} = (X'\Omega X + D^{-1})^{-1}(X'\Omega(t-\mu 1) + D^{-1}W)$ and $\Sigma_{\gamma|.} = (X'\Omega X + D^{-1})^{-1}$

- $s^2_{k,l}\,|\,- \sim IG\left[1, \left(\frac{1}{\nu_{k,l}} + \frac{(\gamma_{k,l} - u'_k \Lambda u_l)^2}{2\sigma^2}\right)\right]$

- $\sigma^2\,|\,- \sim IG\left[\left(\frac{1}{2} + \frac{V(V-1)}{4}\right), \left(\frac{1}{\sigma_2} + \sum_{k<l} \frac{(\gamma_{k,l} - u'_k \Lambda u_l)^2}{2s^2_{k,l}}\right)\right]$

- $\nu_{k,l}\,|\,- \sim IG\left[1, \left(1 + \frac{1}{s^2_{k,l}}\right)\right]$

- $\sigma_2\,|\,- \sim IG\left[1, \left(1 + \frac{1}{\sigma^2}\right)\right]$

- $u_k\,|\,- \sim w_{u_k}\,\delta_0(u_k) + (1 - w_{u_k})\,N(u_k\,|\,m_{u_k}, \Sigma_{u_k})$, where $U^*_k = (u_1 : \cdots : u_{k-1} : u_{k+1} : \cdots : u_V)'\Lambda$, $H_k = diag(s^2_{1,k}, ..., s^2_{k-1,k}, s^2_{k,k+1}, ..., s^2_{k,V})$, $\gamma_k = (\gamma_{1,k}, ..., \gamma_{k-1,k}, \gamma_{k,k+1}, ..., \gamma_{k,V})$, and

$$\Sigma_{u_k} = \left(U^{*'}_h H^{-1}_k U^*_k/\sigma^2 + Q^{-1}\right)^{-1}, \; m_{u_k} = \Sigma_{u_k} U^{*'}_k H^{-1}_k \gamma_k/\sigma^2$$

$$w_{u_k} = \frac{(1-\Delta)N(\gamma_k\,|\,0, \sigma^2 H_k)}{(1-\Delta)N(\gamma_k\,|\,0, \sigma^2 H_k) + \Delta N(\gamma_k\,|\,0, \sigma^2 H_k + U^*_k Q U^{*'}_k)}$$

- $\xi_k|- \sim Ber(1-w_{u_k})$

- $\Delta\,|- \sim Beta\left[(a_\Delta + \sum_{k=1}^{V}\xi_k), (b_\Delta + \sum_{k=1}^{V}(1-\xi_k))\right]$.

- $Q\,|- \sim IW[(\nu + \{\#k : u_k \neq 0\}), (I + \sum_{k:u_k\neq 0} u_k \Lambda u_k')]$.

- $\lambda_r\,|- \sim Ber(p_{\lambda_r})$, where $p_{\lambda_r} = \frac{\pi_r N(\gamma|W_1,\sigma_2^2 D)}{\pi_r N(\gamma|W_1,\sigma_2^2 D)+(1-\pi_r)N(\gamma|W_0,\sigma_2^2 D)}$. Here

  $W_1 = (u_1'\Lambda_1 u_2, ..., u_1'\Lambda_1 u_V, ..., u_{V-1}'\Lambda_1 u_V)'$, $W_0 = (u_1'\Lambda_0 u_2, ..., u_1'\Lambda_0 u_V, ..., u_{V-1}'\Lambda_0 u_V)'$,

  $\Lambda_1 = diag(\lambda_1, .., \lambda_{r-1}, 1, \lambda_{r+1}, .., \lambda_R)$, $\Lambda_0 = diag(\lambda_1, .., \lambda_{r-1}, 0, \lambda_{r+1}, .., \lambda_R)$, for $r = 1, .., R$.

- $\pi_r\,|- \sim Beta(\lambda_r + 1, 1 - \lambda_r + r^\eta)$, for $r = 1, .., R$.

  Using the relationship, $PG(x\,|\,b,c) \propto \exp(-\frac{c^2x}{2})PG(x\,|\,b,0)$ [111], we obtain

- $\omega_i\,|- \sim PG(1, \mu + x_i'\gamma)$, for $i = 1, .., n$.

7.6 Appendix F

Similar to the assumptions made by [143] in their proof of posterior consistency for binary logistic regression, we prove our results assuming that the centering parameter $\mu = 0$ in both the true and the data generating models. We note that the main structure of the proof will remain unchanged with this assumption and the result proved in this chapter can be trivially extended to the setting with nonzero μ.

We begin by defining some notations. In the proof, $\Pi(\cdot)$ will be used to denote the generic probability notation. We define the notation of the log-likelihood function by

$$w_{\gamma,n}(y_n) = \sum_{i=1}^{n} [(x_i'\gamma)y_i - z(x_i'\gamma)],\ z(x_i'\gamma) = \log(1+\exp(x_i'\gamma)). \tag{7.4}$$

We also introduce the function $C_{y_n,n}(\cdot)$ to quantify the curvature of $w_{\gamma,n}(y_n)$ around $\gamma^{(0)}$,

$$C_{y_n,n}(\gamma) = w_{\gamma,n}(y_n) - w_{\gamma^{(0)},n}(y_n) - \nabla w_{\gamma^{(0)},n}(y_n)'(\gamma - \gamma^{(0)}), \tag{7.5}$$

where $\nabla w_{\gamma^{(0)},n}(y_n)$ is the derivative of $w_{\gamma^{(0)},n}(y_n)$ w.r.t. γ, evaluated at $\gamma^{(0)}$. Also the likelihood $p_\gamma(y_n)$ can be written using the above notations as $p_\gamma(y_n) = \prod_{i=1}^{n} \exp(w_{\gamma,n}(y_i))$. The notations $E_\gamma(\cdot)$ and $E_{\gamma^{(0)}}(\cdot)$ have been reserved to denote expectation w.r.t the distribution of $y_n|\gamma$ and $y_n|\gamma^{(0)}$ respectively.

The proof of Theorem 3.3.1 relies in part on the existence of exponentially consistent sequence of tests.

Definition An exponentially consistent sequence of test functions Φ_n for testing $H_0 : \gamma = \gamma^0$ vs. $H_1 : \gamma \in \mathcal{A}_n^c$ satisfies

$$E_{\gamma^0}(\Phi_n) \leq d_1 \exp(-h_1 n), \qquad \sup_{\gamma \in \mathcal{A}_n^c} E_\gamma(1-\Phi_n) \leq d_2 \exp(-h_2 n)$$

for some $d_1, d_2, h_1, h_2 > 0$.

Lemma 7.6.1 *For some $h > 0$, there exists a sequence of test functions for testing $H_0 : \gamma = \gamma^0$ vs. $H_1 : \gamma \in \mathcal{A}_n^c$, which satisfy*

$$E_{\gamma^0}(\Phi_n) \leq \exp(-hn), \qquad \sup_{\gamma \in \mathcal{A}_n^c} E_\gamma(1-\Phi_n) \leq \exp(-hn). \tag{7.6}$$

Proof The construction of the test is provided in the proof of Theorem 2 and Lemma 4 in [55].

We also state another result which will be subsequently used in the proof.

Lemma 7.6.2 *Let* $u_k^{(0)} = (u_{k,1}^{(0)}, ..., u_{k,R}^{(0)})'$ *for* $k = 1, .., V_n$*, and* $\upsilon_{k,l}$ *be the only positive root of the equation*

$$x^2 + x(||u_k^{(0)}||_2 + ||u_l^{(0)}||_2) - \eta_1 = 0,\ k < l. \tag{7.7}$$

Assume $\upsilon = \min_{k,l} \upsilon_{k,l}$*. Then, for* $W = (u_1' u_2, ..., u_{V_n-1}' u_{V_n})'$ *and* $W^{(0)} = (u_1^{(0)'} u_2^{(0)}, ..., u_{V_n-1}^{(0)'} u_{V_n}^{(0)})'$

$$\Pi(||W - W^{(0)}||_\infty < \eta_1) \geq \Pi(||u_k - u_k^{(0)}||_2 \leq \upsilon, \forall\, k = 1, .., V_n). \tag{7.8}$$

Proof for $k < l$,

$$\begin{aligned}
|u_k' u_l - u_k^{(0)'} u_l^{(0)}| &= |\sum_{r=1}^{R} u_{k,r} u_{l,r} - \sum_{r=1}^{R} u_{k,r}^{(0)} u_{l,r}^{(0)}| \\
&\leq |\sum_{r=1}^{R} (u_{k,r} - u_{k,r}^{(0)}) u_{lr}| + |\sum_{r=1}^{R} (u_{l,r} - u_{l,r}^{(0)}) u_{k,r}^{(0)}| \\
&\leq ||u_k - u_k^{(0)}||_2 ||u_l||_2 + ||u_l - u_l^{(0)}||_2 ||u_k^{(0)}||_2 \\
&\leq ||u_k - u_k^{(0)}||_2 \left[||u_l - u_l^{(0)}||_2 + ||u_l^{(0)}||_2 \right] + ||u_l - u_l^{(0)}||_2 ||u_k^{(0)}||_2.
\end{aligned}$$

If $||u_k - u_k^{(0)}||_2 \leq \upsilon$, $\forall\, k = 1, .., V_n$, the above inequality implies

$$|u_k' u_l - u_k^{(0)'} u_l^{(0)}| \leq \upsilon(\upsilon + ||u_l^{(0)}||_2) + \upsilon ||u_k^{(0)}||_2 \leq \eta_1, \forall\, k < l.$$

Hence $\Pi(||W - W^{(0)}||_\infty < \eta_1) \geq \Pi(||u_k - u_k^{(0)}||_2 \leq \upsilon, \forall\, k = 1, .., V_n)$.

Proof of Theorem 3.3.1

Suppose $\mathcal{E}_n = \left\{ y : ||\nabla w_{\gamma^{(0)},n}(y)||_\infty \leq 2\sqrt{nq_n} \right\}$. Then the probability of the vector y_n belonging to the set $\mathcal{E}_n$ is given by,

$$P_{\gamma^{(0)}}(y_n \in \mathcal{E}_n) \geq 1 - P_{\gamma^{(0)}}(\max_{1 \leq j \leq q_n} |\sum_{i=1}^{n} (y_i - \nabla z(x_i'(\gamma - \gamma^{(0)}))) x_{ij}| > 2\sqrt{nq_n}) \geq 1 - \frac{2}{q_n},$$

where the last step follows from the Hoeffding inequality. Note that as $n \to \infty$, $q_n \to \infty$, hence $P_{\gamma^{(0)}}(y_n \in \mathcal{E}_n) \to 1$. Hence, in the subsequent proof we can assume without loss of generality that $y_n \in \mathcal{E}_n$. It can be observed that

$$\Pi_n(\mathcal{A}_n^c) = \frac{\int_{\mathcal{A}_n^c} p_\gamma(y_n)\pi_n(\gamma)}{\int p_\gamma(y_n)\pi_n(\gamma)} = \frac{\int_{\mathcal{A}_n^c} \frac{p_\gamma(y_n)}{p_{\gamma^{(0)}}(y_n)}\pi_n(\gamma)}{\int \frac{p_\gamma(y_n)}{p_{\gamma^{(0)}}(y_n)}\pi_n(\gamma)} = \frac{\mathcal{N}_n}{\mathcal{D}_n} \leq \Phi_n + (1-\Phi_n)\frac{\mathcal{N}_n}{\mathcal{D}_n}, \tag{7.9}$$

where Φ_n is the exponentially consistent sequence of tests given in Lemma 7.6.1. The above equation is true as $\mathcal{N}_n/\mathcal{D}_n \leq 1$. This is in turn true as both are integrals of the same nonnegative functions, $\mathcal{D}_n$ is the integral of that function over the entire set of possible γ's, while $\mathcal{N}_n$ is the integral over a subset $\mathcal{A}_n^c$. In proving Theorem 3.3.1, we will proceed in three steps as following.

(a) Step 1 shows that $\Phi_n \to 0$, as $n \to \infty$, almost surely.

(b) Step 2 shows that $\exp(hn/2)(1-\Phi_n)\mathcal{N}_n \to 0$, as $n \to \infty$, almost surely.

(c) Finally, step 3 shows that $\exp(hn/2)\mathcal{D}_n \to \infty$, as $n \to \infty$.

Here h is the one as defined in Lemma 7.6.1. By (7.9), (a)-(c) implies $\Pi_n(\mathcal{A}_n^c) \to 0$. We will now proceed proving (a)-(c).

(a) Step 1

An application of the Markov inequality and (7.6) in Lemma 7.6.1 yield,

$$P_{\gamma^{(0)}}(\Phi_n > \exp(-nh/2)) \leq E_{\gamma^{(0)}}(\Phi_n)\exp(nh/2) \leq \exp(-nh/2).$$

Therefore $\sum_{n=1}^{\infty} P_{\gamma^{(0)}}(\Phi_n > \exp(-nh/2)) < \infty$.

Applying Borel-Cantelli lemma, Thus, $P_{\gamma^{(0)}}(\Phi_n > \exp(-nh/2)$ happens infinitely often$) = 0$. This means that $\exists n_0$ and a set Ω with $P_{\gamma^{(0)}}(\Omega) = 0$, s.t. for all $n > n_0$, $\Phi_n(\omega) < \exp(-nh/2)$, for all $\omega \in \Omega^c$. Since $\exp(-nh/2) \to 0$, this means that $\Phi_n \to 0$ almost surely.

Thus,

$$\Phi_n \to 0 \quad a.s. \tag{7.10}$$

<u>(b) Step 2</u>

We have

$$\begin{aligned}
E_{\gamma^{(0)}}((1-\Phi_n)\mathcal{N}_n) &= \int (1-\Phi_n) \int_{\mathcal{A}_n^c} \frac{p_\gamma(y_n)}{p_{\gamma^{(0)}}(y_n)} \pi_n(\gamma) p_{\gamma^{(0)}}(y_n) \\
&= \int_{\mathcal{A}_n^c} \int (1-\Phi_n) p_\gamma(y_n) \pi_n(\gamma) \\
&= \int_{\mathcal{A}_n^c} E_\gamma(1-\Phi_n)\pi_n(\gamma) \\
&\leq \sup_{\gamma \in \mathcal{A}_n^c} E_\gamma(1-\Phi_n \Pi(\mathcal{A}_n^c) \\
&\leq \sup_{\gamma \in \mathcal{A}_n^c} E_\gamma(1-\Phi_n) \leq \exp(-nh) \leq \exp(-nh/2).
\end{aligned}$$

Consider the set $\mathcal{G}_{n,h,2} = \{(1-\Phi_n)\mathcal{N}_n \exp(nh/2) > \exp(-nh/4)\}$. The above inequality implies that $\sum_{n=1}^{\infty} P_{\gamma^{(0)}}(\mathcal{G}_{n,h,2}) < \infty$. Again since h is fixed, applying Borel-Cantelli lemma $P_{\gamma^{(0)}}(limsup_{n\to\infty} \mathcal{G}_{n,h,2}) = 0$. Using the definition of limsup of the sets $\mathcal{G}_{n,h,2}$ [85], $P_{\gamma^{(0)}}(\mathcal{G}_{n,h,2}$ happens infinitely often$) = 0$. Thus, $P_{\gamma^{(0)}}((1-\Phi_n)\mathcal{N}_n \exp(nh/2) > \exp(-nh/4)$ happens infinitely often$) = 0$. Let Ω_2 be the set s.t. $P_{\gamma^{(0)}}(\Omega) = 0$ and $(1-\Phi_n(\omega))\mathcal{N}_n \exp(nh/2) > \exp(-nh/4)$ happens infinitely often for all $\omega \in \Omega_2$. This means that $\exists n_{0,2}$ s.t. for all $n > n_{0,2}$, $(1-\Phi_n(\omega))\mathcal{N}_n \exp(nh/2) < \exp(-nh/4)$, for all $\omega \in \Omega_2^c$. Since $\exp(-nh/4) \to 0$, this means that $\exp(nh/2)(1-\Phi_n)\mathcal{N}_n \to 0$ almost surely.

$$\exp(nh/2)(1-\Phi_n)\mathcal{N}_n \to 0 \quad a.s.. \tag{7.11}$$

(c) Step 3

$$\begin{aligned}
\int \frac{p_\gamma(y_n)}{p_{\gamma^{(0)}}(y_n)}\pi(\gamma) &= \int \exp\left(\nabla w_{\gamma^{(0)},n}(y_n)'(\gamma-\gamma^{(0)}) + C_{y_n,n}(\gamma)\right)\pi(\gamma) \\
&\geq \int \exp\left(-||\nabla w_{\gamma^{(0)},n}(y_n)||_\infty ||\gamma-\gamma^{(0)}||_2 - \frac{n}{8}||\gamma-\gamma^{(0)}||_2^2\right)\pi(\gamma) \\
&\geq \int \exp\left(-2\sqrt{nq_n}||\gamma-\gamma^{(0)}||_2 - \frac{n}{8}||\gamma-\gamma^{(0)}||_2^2\right)\pi(\gamma) \\
&\geq \exp\left(-2\sqrt{nq_n}\frac{\eta_1}{n^{\rho/2}} - \frac{n\eta_1^2}{8n^\rho}\right)\Pi\left(||\gamma-\gamma^{(0)}||_2 < \frac{\eta_1}{n^{\rho/2}}\right),
\end{aligned}$$

where ρ is the one defined in the statement of the theorem and the inequality in the second line follows from the Taylor series expansion after taking into account that $\nabla^2 z(\cdot) \leq 1/4$ ($z(\cdot)$ defined in (7.4)), which is true as $\frac{d^2}{df^2}\log\left(1+e^f\right) = \frac{e^f}{(1+e^f)^2} \leq 1/4$. The inequality in the third line follows from the fact that $y_n \in \mathcal{E}_n$.

First, observe that, given all the hierarchical parameters, the Bayesian network lasso prior distribution on γ can be written as $\gamma = W + \gamma_2$, where γ_2 follows the ordinary Bayesian lasso shrinkage prior. With this observation, one can see

$$\Pi\left(||\gamma-\gamma^{(0)}||_2 < \frac{\eta_1}{n^{\rho/2}}\right) \geq \Pi\left(||\gamma_2-\gamma_2^{(0)}||_2 < \frac{\eta_1}{2n^{\rho/2}}\right)\Pi\left(||W-W^{(0)}||_2 < \frac{\eta_1}{2n^{\rho/2}}\right),$$

where W and $W^{(0)}$ are as defined in Lemma 7.6.2. We will show sequentially

(i) $-\log\Pi\left(||W-W^{(0)}||_2 < \frac{\eta_1}{2n^{\rho/2}}\right) = o(n)$ and

(ii) $-\log\left\{\Pi\left(||\gamma_2-\gamma_2^{(0)}||_2 < \frac{\eta_1}{2n^{\rho/2}}\right)\right\} = o(n)$.

(i) Note that, with R (dimensions of the latent variables) and Δ (probability of a node being influential) as defined before we obtain,

$$\begin{aligned}\Pi(||W-W^{(0)}||_2<\frac{\eta_1}{2n^{\rho/2}}) &\geq \Pi(||u_k-u_k^{(0)}||_2\leq \upsilon_n, \forall k=1,..,V_n)\\ &\geq \mathrm{E}\left[\Pi(||u_k-u_k^{(0)}||_2\leq \upsilon_n, \forall k=1,..,V_n|\Delta)\right]\\ &\geq \mathrm{E}\left[\prod_{k=1}^{V_n}\left\{\exp\left(-\frac{1}{2}u_k^{(0)'}u_k^{(0)}\right)\Pi(||u_k||_2\leq \upsilon_n|\Delta)\right\}\right], \qquad (7.12)\end{aligned}$$

where the first inequality follows from Lemma 7.6.2 by replacing η_1 with $\frac{\eta_1}{2n^{\rho/2}}$ with a slight abuse of notation, and υ_n is defined accordingly. The last inequality follows from the Anderson's Lemma. We will now make use of the fact that $\int_{-a}^{a}\exp(-x^2/2)dx\geq \exp(-a^2)2a$ to conclude

$$\Pi(||u_k||_2\leq \upsilon_n|\Delta)\geq \prod_{r=1}^{R}\Pi\left(|u_{k,r}|\leq \frac{\upsilon_n}{R}|\Delta\right)=\prod_{r=1}^{R}\left((1-\Delta)+\frac{\Delta}{\sqrt{2\pi}}\int_{-\upsilon_n/R}^{\upsilon_n/R}\exp(-x^2/2)\right)$$
$$\geq \prod_{r=1}^{R}\left((1-\Delta)+\frac{\Delta}{\sqrt{2\pi}}\exp(-\upsilon_n^2/R^2)\frac{2\upsilon_n}{R}\right)\geq \left[(1-\Delta)+\frac{\Delta}{\sqrt{2\pi}}\exp(-\upsilon_n^2/R^2)\frac{2\upsilon_n}{R}\right]^R.$$

$$\begin{aligned}\prod_{k=1}^{V_n}\Pi(||u_k||_2\leq \upsilon_n) &\geq \mathrm{E}\left[(1-\Delta)+\frac{\Delta}{\sqrt{2\pi}}\exp(-\upsilon_n^2/R^2)\frac{2\upsilon_n}{R}\right]^{RV_n}\\ &= \mathrm{E}\left[\sum_{h_1=1}^{RV_n}\binom{RV_n}{h_1}(1-\Delta)^{h_1}\Delta^{RV_n-h_1}\left(\frac{2\upsilon_n}{R}\right)^{RV_n-h_1}\exp\left(-(RV_n-h_1)\upsilon_n^2/R^2\right)\right]\\ &\geq \sum_{h_1=1}^{RV_n}\binom{RV_n}{h_1}Beta(RV_n-h_1+1,h_1+1)\\ &\left(\frac{2\upsilon_n}{R}\right)^{RV_n-h_1}\exp\left(-(RV_n-h_1)\upsilon_n^2/R^2\right)\\ &\geq \sum_{h_1=1}^{RV_n}\frac{(RV_n)!}{h_1!(RV_n-h_1)!}\frac{h_1!(RV_n-h_1)!}{(RV_n+1)!}\\ &\left(\frac{2\upsilon_n}{R}\right)^{RV_n-h_1}\exp\left(-(RV_n-h_1)\upsilon_n^2/R^2\right)\\ &\geq \frac{RV_n}{RV_n+1}\left(\frac{2\upsilon_n}{R}\right)^{RV_n}\exp(-V_n\upsilon_n^2/R).\end{aligned}$$

Where the last inequality follows from Lemma 7.6.2 by considering the fact that, $\upsilon_n = \min_{k,l} \frac{-[||u_k^{(0)}||+||u_l^{(0)}||]+\sqrt{[||u_k^{(0)}||+||u_l^{(0)}||]^2+2\eta_1/n^{\rho/2}}}{2} \leq \frac{\sqrt{\eta_1}}{\sqrt{2}n^{\rho/4}}$. Hence, $0 < \frac{2\upsilon_n}{R} < 1$ for large n. It now follows from (7.12) that

$$-\log \Pi\left(||W - W^{(0)}||_2 < \frac{\eta_1}{2n^{\rho/2}}\right) \leq \sum_{k=1}^{V_n} \frac{u_k^{(0)'} u_k^{(0)}}{2} + \frac{V_n \eta_1}{2Rn^{\rho/2}} - (RV_n)\log\left(\frac{2\sqrt{\eta_1}}{\sqrt{2}Rn^{\rho/4}}\right) + \log(RV_n + 1)$$
$$- \log(RV_n) = o(n),$$

by the assumptions (A) and (B). This proves (i).

We will now prove (ii). Let $\mathcal{S}^0 = \{j : \gamma_{2,j}^{(0)} \neq 0\}$. Define s as the vector of upper triangular part of the matrix with (k,l)th entry $s_{k,l}$. It follows that

$$\Pi\left(||\gamma_2 - \gamma_2^{(0)}||_2 < \frac{\eta_1}{2n^{\rho/2}}\right) \geq \Pi\left(|\gamma_{2,j} - \gamma_{2,j}^{(0)}| < \frac{\eta_1}{2\sqrt{q_n}n^{\rho/2}}, j \in \mathcal{S}^0\right) \Pi\left(\sum_{j \notin \mathcal{S}^0} |\gamma_{2,j}|^2 < \frac{(q_n - s_{2,n}^0)\eta_1^2}{4q_n n^\rho}\right). \tag{7.13}$$

We will lower bound two components of the product in (7.13) individually. By Chebyshev's inequality

$$\Pi\left(\sum_{j \notin \mathcal{S}^0} |\gamma_{2,j}|^2 < \frac{(q_n - s_{2,n}^0)\eta_1^2}{4q_n n^\rho}\right) \geq \left(1 - \frac{E[\sum_{j \notin \mathcal{S}^0} |\gamma_{2,j}|^2] 4q_n n^\rho}{(q_n - s_{2,n}^0)\eta_1^2}\right)$$
$$= \left(1 - \frac{2\theta_n q_n n^\rho}{\eta_1^2}\right). \tag{7.14}$$

$$\Pi\left(|\gamma_{2,j} - \gamma_{2,j}^{(0)}| < \frac{\eta_1}{2\sqrt{q_n}n^{\rho/2}}, j \in \mathcal{S}^0\right) = E\left[\Pi\left(|\gamma_{2,j} - \gamma_{2,j}^{(0)}| < \frac{\eta_1}{2\sqrt{q_n}n^{\rho/2}}, j \in \mathcal{S}^0 | s_{\mathcal{S}^0}\right)\right]$$
$$= E\left[\prod_{j \in \mathcal{S}^0} \Pi\left(|\gamma_{2,j} - \gamma_{2,j}^{(0)}| < \frac{\eta_1}{2\sqrt{q_n}n^{\rho/2}} | s_{\mathcal{S}^0}\right)\right].$$

Using the fact that $\int_a^b e^{-x^2/2}dx \geq e^{-(a^2+b^2)/2}(b-a)$, one obtains

$$\prod_{j\in\mathcal{S}^0}\Pi\left(|\gamma_{2,j}-\gamma_{2,j}^{(0)}|<\frac{\eta_1}{2\sqrt{q_n}n^{\rho/2}}|s_{\mathcal{S}^0}\right)\geq\prod_{j\in\mathcal{S}^0}\left\{\left(\frac{\eta_1}{\sqrt{2q_nn^{\rho}\pi s_j^2}}\right)\exp\left(-\frac{|\gamma_{2,j}^0|^2+\eta_1^2/(4q_nn^{\rho})}{s_j^2}\right)\right\}.$$

Thus

$$\begin{aligned}&\Pi\left(|\gamma_{2,j}-\gamma_{2,j}^{(0)}|<\frac{\eta_1}{2\sqrt{q_n}n^{\rho/2}},j\in\mathcal{S}^0\right)\\&\geq E\left[\prod_{j\in\mathcal{S}^0}\left\{\left(\frac{\eta_1}{\sqrt{2q_nn^{\rho}\pi s_j^2}}\right)\exp\left(-\frac{|\gamma_{2,j}^0|^2+\eta_1^2/(4q_nn^{\rho})}{s_j^2}\right)\right\}\right]\\&\geq\left(\frac{\eta_1\theta_n}{\sqrt{2q_nn^{\rho}\pi}}\right)^{s_{2,n}^0}\prod_{j\in\mathcal{S}^0}\int_{s_j}\left\{\frac{1}{\sqrt{s_j^2}}\exp\left(-\frac{|\gamma_{2,j}^0|^2+\eta_1^2/(4q_nn^{\rho})}{s_j^2}-\frac{\theta_ns_j^2}{2}\right)ds_j^2\right\}.\end{aligned}$$

Use the change of variable $\frac{1}{s_j^2}=z_j$ and the normalizing constant from the inverse Gaussian density to deduce

$$\begin{aligned}&\int_{s_j}\left\{\frac{1}{\sqrt{s_j^2}}\exp\left(-\frac{|\gamma_{2,j}^0|^2+\eta_1^2/(4q_nn^{\rho})}{s_j^2}-\frac{\theta_ns_j^2}{2}\right)ds_j^2\right\}\\&=\int_{z_j}\left\{\frac{1}{\sqrt{z_j^3}}\exp\left(-(|\gamma_{2,j}^0|^2+\eta_1^2/(4q_nn^{\rho})z_j-\frac{\theta_n}{2z_j}\right)dz_j\right\}\\&=\sqrt{\left(\frac{2\pi}{\theta_n}\right)}\exp\left(-\theta_n\sqrt{2\left(|\gamma_{2,j}^0|^2+\eta_1^2/(4q_nn^{\rho})\right)}\right).\end{aligned}$$

Therefore,

$$\Pi\left(|\gamma_{2,j}-\gamma_{2,j}^{(0)}|<\frac{\eta_1}{2\sqrt{q_n}n^{\rho/2}},j\in\mathcal{S}^0\right)\geq\left(\frac{\eta_1\sqrt{\theta_n}}{\sqrt{q_nn^{\rho}}}\right)^{s_{2,n}^0}\exp\left(-\theta_n\sum_{j\in\mathcal{S}^0}\sqrt{2\left(|\gamma_{2,j}^0|^2+\eta_1^2/(4q_nn^{\rho})\right)}\right).\tag{7.15}$$

Combining results from (7.14) and (7.15)

$$\Pi\left(||\gamma_2-\gamma_2^{(0)}||_2<\frac{\eta_1}{2n^{\rho/2}}\right)\geq\left(\frac{\eta_1\sqrt{\theta_n}}{\sqrt{q_nn^{\rho}}}\right)^{s_{2,n}^0}\exp\left(-\theta_n\sum_{j\in\mathcal{S}^0}\sqrt{2\left(|\gamma_{2,j}^0|^2+\eta_1^2/(4q_nn^{\rho})\right)}\right)\left(1-\frac{2\theta_nq_nn^{\rho/2}}{\eta_1^2}\right).$$

Referring to Assumption (F),

$$-\log\Pi\left(||\gamma_2-\gamma_2^{(0)}||_2<\frac{\eta_1}{2n^{\rho/2}}\right)\leq s_{2,n}^0[\eta+\log(q_n)+(3\rho/4)\log(n)+\log(\log(n))/2]$$

$$+\frac{\sqrt{2\left(|\gamma_{2,j}^0|^2+\eta_1^2/(4q_nn^\rho)\right)}}{q_nn^{\rho/2}\log(n)}-\log\left(1-\frac{2}{\eta^2\log(n)}\right)=o(n), \tag{7.16}$$

under assumptions (B)-(F).

Finally,

$$-\log(\mathcal{D}_n)\leq 2\sqrt{nq_n}\frac{\eta_1}{n^{\rho/2}}+\frac{n\eta^2}{8n^\rho}-\log\Pi\left(||\gamma-\gamma^{(0)}||_2<\frac{\eta_1}{n^{\rho/2}}\right)$$

$$=2\eta\sqrt{q_n}n^{(1-\rho)/2}+\frac{\eta_1^2}{8}n^{1-\rho}-\log\Pi\left(||\gamma-\gamma^{(0)}||_2<\frac{\eta_1}{n^{\rho/2}}\right).$$

Using (7.16), the fact that $(1-\rho)/2\in(-1/2,0)$ and assumption (B), we obtain $-\log(\mathcal{D}_n)=o(n)$. Thus (c) follows.

7.7 Appendix G

This section provides full conditionals for all the parameters in the Nonparametric Bayesian network regression described in Chapter 4.

Let $x_i=(a_{i,1,2},a_{i,1,3},...,a_{i,1,V},a_{i,2,3},a_{i,2,4},...,a_{i,2,V},....,a_{i,V-1,V})'$ be of dimension $q\times 1$, where $q=\frac{V\times(V-1)}{2}$. Assume $y=(y_1,...,y_n)'\in\mathbb{R}^n$ and $X=(x_1:\cdots:x_n)'$ is an $n\times q$ matrix. Further, assume $W_d=(u_{1,d}'\Lambda_d u_{2,d},...,u_{1,d'}\Lambda_d u_{V,d},....,u_{(V-1),d}'\Lambda_d u_{V,d})'$, $D_d=diag(s_{1,2,d},...,s_{V-1,V,d})$ and $\gamma_d=(\gamma_{1,2,d},...,\gamma_{V-1,V,d})'$.

With n data points, the hierarchical model is written as

$$y_i \,|\, (z_i = d) \sim N(\mu_d + X\gamma_d, \tau^2 I);\, d = 1,...,H;\, i = 1,...,N$$

$$\gamma_d \sim \mathrm{N}(W_d, \tau^2 D_d),\ \pi(\tau^2) \propto \frac{1}{\tau^2},\ \mu_d \sim N(0,1),\ u_{k,d}|\xi_{k,d} = 1 \sim N(u_{k,d}\,|\,0, Q_d),\, u_{k,d}|\xi_{k,d} = 0 \sim \delta_0,$$

$$s_{k,l,d} \sim Exp(\theta_d^2/2),\ \theta_d^2 \sim Gamma(\zeta, \iota),\ Q_d \sim IW(S, \nu),\ \Delta_d \sim Beta(a,b),$$

$$\xi_{k,d} \sim Ber(\Delta_d),\ \lambda_{r,d} \sim Ber(\pi_{r,d}),\ \pi_{r,d} \sim Beta(1, r^{\eta}),\, \eta > 1,\ P(z_i = d) = \omega_d$$

$$\omega_1 = v_1^*,\, \omega_2 = v_2^*(1 - v_1^*), .., \omega_{H-1} = v_{H-1}^* \prod_{l=1}^{H-2}(1 - v_l^*),\, \omega_H = \prod_{l=1}^{H-1}(1 - v_l^*),$$

$$v_l^* \sim Beta(1 - \alpha_1, \alpha_2 + l\alpha_1),\, l = 1, .., H-1;\, \alpha_1 \sim U(0,1),\, \alpha_2 \sim Gamma(a_\alpha, b_\alpha).$$

The model computation proceeds using the popular Markov Chain Monte Carlo algorithm with the full conditional distributions of parameters are given as following:

- $\mu_d \,|\, - \sim N\left(\frac{1_{E_d}{}'(y_{E_d} - X_{E_d}\gamma_d)}{1_{E_d}{}'1_{E_d}}, \frac{\tau^2}{1_{E_d}{}'1_{E_d}}\right)$
- $\gamma_d \,|\, - \sim N(\mu_{\gamma_d|\cdot}, \Sigma_{\gamma_d|\cdot})$, where $\mu_{\gamma_d|\cdot} = (X'_{E_d}X_{E_d} + D_d^{-1})^{-1}(X'_{E_d}(y_{E_d} - \mu_d 1_{E_d}) + D_d^{-1}W_d)$ and $\Sigma_{\gamma_d|\cdot} = \tau^2(X'_{E_d}X_{E_d} + D_d^{-1})^{-1}$
- $\tau^2 \,|\, - \sim IG\left[(\frac{N}{2} + \frac{V(V-1)H}{4}), \sum_{d=1}^{H} \frac{||(y_{E_d} - \mu_d 1_{E_d} - X_{E_d}\gamma_d)||^2 + (\gamma_d - W_d)' D_d^{-1}(\gamma_d - W_d)}{2}\right]$
- $s_{k,l,d} \,|\, - \sim GIG\left[\frac{1}{2}, \frac{(\gamma_{k,l,d} - u'_{k,d}\Lambda_d u_{l,d})^2}{\tau^2}, \theta_d^2\right]$, where GIG denotes the generalized inverse Gaussian distribution.
- $\theta_d^2 \,|\, - \sim Gamma\left[\left(\zeta + \frac{V(V-1)}{2}\right), \left(\iota + \sum_{k<l}\frac{s_{k,l,d}}{2}\right)\right]$
- $u_{k,d} \,|\, - \sim w_{u_{k,d}}\delta_0(u_{k,d}) + (1 - w_{u_{k,d}})N(u_{k,d}\,|\,m_{u_{k,d}}, \Sigma_{u_{k,d}})$, where

  $U^*_{k,d} = (u_{1,d} : \cdots : u_{k-1,d} : u_{k+1,d} : \cdots : u_{V,d})'\Lambda_d,$

$bH_{k,d} = diag(s_{1,k,d},...,s_{k-1,k,d},s_{k,k+1,d},...,s_{k,V,d})$,

$\gamma_{k,d} = (\gamma_{1,k,d},...,\gamma_{k-1,k,d},\gamma_{k,k+1,d},...,\gamma_{k,V,d})$, and

$$\Sigma_{u_{k,d}} = \left(U^{*'}_{h,d}H^{-1}_{k,d}U^{*}_{k,d}/\tau^2 + Q_d^{-1}\right)^{-1},\ m_{u_{k,d}} = \Sigma_{u_{k,d}}U^{*'}_{k,d}H^{-1}_{k,d}\gamma_{k,d}/\tau^2$$
$$w_{u_{k,d}} = \frac{(1-\Delta_d)N(\gamma_{k,d}\,|\,0,\tau^2 H_{k,d})}{(1-\Delta_d)N(\gamma_{k,d}\,|\,0,\tau^2 H_{k,d}) + \Delta_d N(\gamma_{k,d}\,|\,0,\tau^2 H_{k,d} + U^{*}_{k,d}Q_d U^{*'}_{k,d})}$$

- $\xi_{k,d}|- \sim Ber(1-w_{u_{k,d}})$

- $\Delta_d\,|- \sim Beta\left[(a+\sum_{k=1}^{V}\xi_{k,d}),(b+\sum_{k=1}^{V}(1-\xi_{k,d}))\right]$.

- $Q_d\,|- \sim IW[(S+\sum_{k:u_{k,d}\neq 0}u_{k,d}\Lambda_d u'_{k,d}),(\nu+\{\#k : u_{k,d}\neq 0\})]$.

- $\lambda_{r,d}\,|- \sim Ber(p_{\lambda_{r,d}})$, where $p_{\lambda_{r,d}} = \frac{\pi_{r,d}N(\gamma_d\,|\,W_{1,d},\tau^2 D_d)}{\pi_{r,d}N(\gamma_d\,|\,W_{1,d},\tau^2 D_d)+(1-\pi_{r,d})N(\gamma_d\,|\,W_{0,d},\tau^2 D_d)}$. Here

  $W_{1,d} = (u'_{1,d}\Lambda_{1,d}u_{2,d},...,u'_{1,d}\Lambda_{1,d}u_{V,d},....,u'_{V-1,d}\Lambda_{1,d}u_{V,d})'$,

  $W_{0,d} = (u'_{1,d}\Lambda_{0,d}u_{2,d},...,u'_{1,d}\Lambda_{0,d}u_{V,d},....,u'_{V-1,d}\Lambda_{0,d}u_{V,d})'$,

  $\Lambda_{1,d} = diag(\lambda_{1,d},..,\lambda_{r-1,d},1,\lambda_{r+1,d},..,\lambda_{R,d})$,

  $\Lambda_{0,d} = diag(\lambda_{1,d},..,\lambda_{r-1,d},0,\lambda_{r+1,d},..,\lambda_{R,d})$, for $r = 1,..,R$.

- $\pi_{r,d}\,|- \sim Beta(\lambda_{r,d}+1,1-\lambda_{r,d}+r^{\eta})$, for $r = 1,..,R$.

- $P(z_i = d\,|-) = \frac{\omega_d N(y_i\,|\,x_i'\gamma_d+\mu_d,\tau^2)}{\sum_{d'=1}^{H}\omega_{d'}N(y_i\,|\,x_i'\gamma_{d'}+\mu_{d'},\tau^2)}$, for $d = 1,..,H$. $v_l^*\,|-Beta(1-\alpha_1+\#\{i : z_i = l\},\alpha_2+l\alpha_1+\sum_{ss=l+1}^{H}\#\{i : z_i = ss\})$, $l = 1,...,H-1$,

  $\omega_1 = v_1^*$, $\omega_2 = v_2^*(1-v_1^*),..,\omega_{H-1} = v_{H-1}^*\prod_{l=1}^{H-2}(1-v_l^*)$, $\omega_H = \prod_{l=1}^{H-1}(1-v_l^*)$

- Parameters α_1 and α_2 are updated using Metropolis Hastings algorithm.

Bibliography

[1] Felix Abramovich and Vadim Grinshtein. High-dimensional classification by sparse logistic regression. *arXiv preprint arXiv:1706.08344*, 2017.

[2] James H Albert and Siddhartha Chib. Bayesian analysis of binary and polychotomous response data. *Journal of the American Statistical Association*, 88(422):669–679, 1993.

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

[4] David J Aldous. Exchangeability and related topics. In *École d'Été de Probabilités de Saint-Flour XIII 1983*, pages 1–198. Springer, 1985.

[5] Artin Armagan, David B Dunson, and Jaeyong Lee. Generalized double Pareto shrinkage. *Statistica Sinica*, 23(1):119–143, 2013.

[6] Artin Armagan, David B Dunson, Jaeyong Lee, Waheed U Bajwa, and Nate Strawn. Posterior consistency in linear models under shrinkage priors. *Biometrika*, 100(4):1011–1018, 2013.

[7] Eduard Belitser and Nurzhan Nurushev. Needles and straw in a haystack: robust confidence for possibly sparse sequences. *arXiv preprint arXiv:1511.01803*, 2015.

[8] Anirban Bhattacharya, Antik Chakraborty, and Bani K Mallick. Fast sampling with gaussian scale mixture priors in high-dimensional regression. *Biometrika*, pages 985–991, 2016.

[9] Monica Billio, Mila Getmansky, Andrew W Lo, and Loriana Pelizzon. Econometric measures of connectedness and systemic risk in the finance and insurance sectors. *Journal of financial economics*, 104(3):535–559, 2012.

[10] Thomas E Brown, Philipp C Reichel, and Donald M Quinlan. Executive function impairments in high iq adults with adhd. *Journal of Attention Disorders*, 13(2):161–167, 2009.

[11] Ed Bullmore and Olaf Sporns. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews. Neuroscience*, 10(3):186–198, 2009.

[12] Wray Buntine, Lan Du, and Petteri Nurmi. Bayesian networks on dirichlet distributed vectors. *On Probabilistic Graphical Models*, page 33, 2010.

[13] Gareth Butland, José Manuel Peregrín-Alvarez, Joyce Li, Wehong Yang, Xiaochun Yang, Veronica Canadien, Andrei Starostine, Dawn Richards, Bryan Beattie, Nevan Krogan, et al. Interaction network containing conserved and essential protein complexes in escherichia coli. *Nature*, 433(7025):531–537, 2005.

[14] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis. *Journal of the ACM (JACM)*, 58(3):11, 2011.

[15] Bruce Caplan, Jeffrey S Kreutzer, and John DeLuca. *Encyclopedia of Clinical Neuropsychology; With 199 Figures and 139 Tables.* Springer, 2011.

[16] Shelley H Carson, Jordan B Peterson, and Daniel M Higgins. Decreased latent inhibition is associated with increased creative achievement in high-functioning individuals. *Journal of personality and social psychology*, 85(3):499, 2003.

[17] Carlos M Carvalho, Nicholas G Polson, and James G Scott. The horseshoe estimator for sparse signals. *Biometrika*, 97(2):465–480, 2010.

[18] Ismaël Castillo, Judith Rousseau, et al. A bernstein–von mises theorem for smooth functionals in semiparametric models. *The Annals of Statistics*, 43(6):2353–2383, 2015.

[19] Ismaël Castillo, Aad van der Vaart, et al. Needles and straw in a haystack: Posterior concentration for possibly sparse sequences. *The Annals of Statistics*, 40(4):2069–2101, 2012.

[20] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A Parrilo, and Alan S Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

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

[22] Arindam Chatterjee and Soumendra Nath Lahiri. Bootstrapping lasso estimators. *Journal of the American Statistical Association*, 106(494):608–625, 2011.

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

[24] Nicholas A Christakis and James H Fowler. The spread of obesity in a large social network over 32 years. *n engl j med*, 2007(357):370–379, 2007.

[25] Nicholas A Christakis and James H Fowler. The collective dynamics of smoking in a large social network. *New England Journal of Medicine*, 358(21):2249–2258, 2008.

[26] Comparing connectomes across subjects and populations at different scales. Meskaldji, djalel eddine and fischi-gomez, elda and griffa, alessandra and hagmann, patric and morgenthaler, stephan and thiran, jean-philippe. *NeuroImage*, 80:416–425, 2013.

[27] R Cameron Craddock, Paul E Holtzheimer III, Xiaoping P Hu, and Helen S Mayberg. Disease state prediction from resting state functional connectivity. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 62(6):1619–1628, 2009.

[28] Jeffrey M Cucina, Nicholas L Vasilopoulos, and Arwen H DeCostanza. Using principal component scores to enhance the validity and reliability of big five personality measures. *Journal of Individual Differences*, 2017.

[29] Madelaine Daianu, Neda Jahanshad, Talia M Nir, Arthur W Toga, Clifford R Jack Jr, Michael W Weiner, and Paul M Thompson, for the Alzheimer's Disease Neuroimaging Initiative. Breakdown of brain connectivity between normal aging and alzheimer's disease: a structural k-core network analysis. *Brain connectivity*, 3(4):407–422, 2013.

[30] Kayla De la Haye, Garry Robins, Philip Mohr, and Carlene Wilson. Obesity-related behaviors in adolescent friendship networks. *Social Networks*, 32(3):161–167, 2010.

[31] Mukund Deshpande, Michihiro Kuramochi, Nikil Wale, and George Karypis. Frequent substructure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering*, 17(8):1036–1050, 2005.

[32] Rahul S Desikan, Florent Ségonne, Bruce Fischl, Brian T Quinn, Bradford C Dickerson, Deborah Blacker, Randy L Buckner, Anders M Dale, R Paul Maguire, Bradley T Hyman, et al. An automated labeling system for subdividing the human cerebral cortex on MRI scans into gyral based regions of interest. *Neuroimage*, 31(3):968–980, 2006.

[33] Patrick Doreian. Causality in social network analysis. *Sociological Methods & Research*, 30(1):81–114, 2001.

[34] Daniele Durante and David B Dunson. Nonparametric bayes dynamic modeling of relational data. *Biometrika*, 101(4):883–898, 2014.

[35] Daniele Durante and David B. Dunson. Bayesian inference and testing of group differences in brain networks. *Bayesian Analysis*, doi:10.1214/16-BA1030, 2017. Advance publication.

[36] Daniele Durante, David B Dunson, et al. Bayesian inference and testing of group differences in brain networks. *Bayesian Analysis*, 2017.

[37] Daniele Durante, David B Dunson, et al. Bayesian inference and testing of group differences in brain networks. *Bayesian analysis*, 13(1):29–58, 2018.

[38] Daniele Durante, David B Dunson, and Joshua T Vogelstein. Nonparametric bayes modeling of populations of networks. *Journal of the American Statistical Association*, 112(520):1516–1530, 2017.

[39] Paul Erdos and Alfréd Rényi. On the evolution of random graphs. *Publication of the Mathematical Institute of the Hungarian Academy of Sciences*, 5(1):17–60, 1960.

[40] Michael D Escobar and Mike West. Bayesian density estimation and inference using mixtures. *Journal of the American Statistical Association*, 90(430):577–588, 1995.

[41] Maryam Fazel, Haitham Hindi, and Stephen P Boyd. Log-det heuristic for matrix rank minimization with applications to hankel and euclidean distance matrices. In *American Control Conference, 2003. Proceedings of the 2003*, volume 3, pages 2156–2162. IEEE, 2003.

[42] Hongliang Fei and Jun Huan. Boosting with structure information in the functional space: an application to graph classification. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 643–652. ACM, 2010.

[43] Yoram Finkelstein, Jacob Vardi, and Israel Hod. Impulsive artistic creativity as a presentation of transient cognitive alterations. *Behavioral Medicine*, 17(2):91–94, 1991.

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

[45] Alex Fornito, Andrew Zalesky, and Michael Breakspear. Graph analysis of the human connectome: promise, progress, and pitfalls. *Neuroimage*, 80:426–444, 2013.

[46] Bailey K Fosdick and Peter D Hoff. Testing and modeling dependencies between a network and nodal attributes. *Journal of the American Statistical Association*, 110(511):1047–1056, 2015.

[47] James H Fowler and Nicholas A Christakis. Dynamic spread of happiness in a large social network: longitudinal analysis over 20 years in the framingham heart study. *British Medical Journal*, 337:a2338, 2008.

[48] Chris Fraley, Adrian E Raftery, T Brendan Murphy, and Luca Scrucca. mclust version 4 for r: normal mixture modeling for model-based clustering, classification, and density estimation. Technical report, 2012.

[49] Ove Frank and David Strauss. Markov graphs. *Journal of the American Statistical Association*, 81(395):832–842, 1986.

[50] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software*, 33(1):1–22, 2010.

[51] Alan E Gelfand and Sujit K Ghosh. Model choice: a minimum posterior predictive loss approach. *Biometrika*, 85(1):1–11, 1998.

[52] Andrew Gelman, John B Carlin, Hal S Stern, David B Dunson, Aki Vehtari, and Donald B Rubin. *Bayesian data analysis*, volume 2. CRC press Boca Raton, FL, 2014.

[53] Andrew Gelman, Jessica Hwang, and Aki Vehtari. Understanding predictive information criteria for bayesian models. *Statistics and computing*, 24(6):997–1016, 2014.

[54] Edward I George and Robert E McCulloch. Variable selection via gibbs sampling. *Journal of the American Statistical Association*, 88(423):881–889, 1993.

[55] Subhashis Ghosal, Anindya Roy, et al. Posterior consistency of gaussian process prior for nonparametric binary regression. *The Annals of Statistics*, 34(5):2413–2429, 2006.

[56] Edgar N Gilbert. Random graphs. *The Annals of Mathematical Statistics*, 30(4):1141–1144, 1959.

[57] Cedric E Ginestet, Arnaud P Fournel, and Andrew Simmons. Statistical network analysis for functional mri: summary networks and group comparisons. *Frontiers in computational neuroscience*, 8:51, 2014.

[58] Matthew F Glasser, Stamatios N Sotiropoulos, J Anthony Wilson, Timothy S Coalson, Bruce Fischl, Jesper L Andersson, Junqian Xu, Saad Jbabdi, Matthew Webster, Jonathan R Polimeni, et al. The minimal preprocessing pipelines for the human connectome project. *Neuroimage*, 80:105–124, 2013.

[59] Robert B Gramacy. `R` package `monomvn`. 2013.

[60] Sharmistha Guha and Abel Rodriguez. Bayesian regression with undirected network predictors with an application to brain connectome data. *arXiv preprint arXiv:1803.10655*, 2018.

[61] R Guhaniyogi and A Rodriguez. Joint modeling of longitudinal relational data and exogenous variables. *https://www.soe.ucsc.edu/sites/default/files/technical-reports/UCSC-SOE-17-17.pdf*, 2017.

[62] Rajarshi Guhaniyogi, Shaan Qamar, and David B Dunson. Bayesian tensor regression. *Journal of Machine Learning Research*, 18(79):1–31, 2017.

[63] Steve Hanneke, Wenjie Fu, Eric P Xing, et al. Discrete temporal models of social networks. *Electronic Journal of Statistics*, 4:585–605, 2010.

[64] Chris Hans. Bayesian lasso regression. *Biometrika*, 96(4):835–845, 2009.

[65] Christoph Helma, Ross D. King, Stefan Kramer, and Ashwin Srinivasan. The predictive toxicology challenge 2000–2001. *Bioinformatics*, 17(1):107–108, 2001.

[66] Peter Hoff. Modeling homophily and stochastic equivalence in symmetric relational data. In *Advances in neural information processing systems*, pages 657–664, 2008.

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

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

[69] Peter D Hoff. Multiplicative latent factor models for description and prediction of social networks. *Computational and mathematical organization theory*, 15(4):261, 2009.

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

[71] Peter D Hoff. Multilinear tensor regression for longitudinal relational data. *The Annals of Applied Statistics*, 9(3):1169, 2015.

[72] Peter D Hoff, Adrian E Raftery, and Mark S Handcock. Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098, 2002.

[73] Lawrence Hubert and Phipps Arabie. Comparing partitions. *Journal of classification*, 2(1):193–218, 1985.

[74] Madhura Ingalhalikar, Alex Smith, Drew Parker, Theodore D Satterthwaite, Mark A Elliott, Kosha Ruparel, Hakon Hakonarson, Raquel E Gur, Ruben C Gur, and Ragini Verma. Sex differences in the structural connectome of the human brain. *Proceedings of the National Academy of Sciences*, 111(2):823–828, 2014.

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

[76] Hemant Ishwaran and J Sunil Rao. Spike and slab variable selection: frequentist and Bayesian strategies. *Annals of Statistics*, 33(2):730–773, 2005.

[77] Saad Jbabdi, Stamatios N Sotiropoulos, Suzanne N Haber, David C Van Essen, and Timothy E Behrens. Measuring macroscopic brain connections in vivo. *Nature neuroscience*, 18(11):1546, 2015.

[78] Valen E Johnson and David Rossell. Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association*, 107(498):649–660, 2012.

[79] Timothy A Judge, Chad A Higgins, Carl J Thoresen, and Murray R Barrick. The big five personality traits, general mental ability, and career success across the life span. *Personnel psychology*, 52(3):621–652, 1999.

[80] Rex E Jung, Judith M Segall, H Jeremy Bockholt, Ranee A Flores, Shirley M Smith, Robert S Chavez, and Richard J Haier. Neuroanatomy of creativity. *Human Brain Mapping*, 31(3):398–409, 2010.

[81] G Kiar, K Gorgolewski, and D Kleissas. Example use case of sic with the ndmg pipeline (sic: ndmg). *GigaScience Database*, 2017.

[82] G Kiar, W Gray Roncal, D Mhembere, E Bridgeford, R Burns, and JT Vogelstein. ndmg: Neurodata's MRI graphs pipeline, 2016.

[83] Gregory Kiar, Krzysztof J Gorgolewski, Dean Kleissas, William Gray Roncal, Brian Litt, Brian Wandell, Russel A Poldrack, Martin Wiener, R Jacob Vogelstein, Randal Burns,

et al. Science in the cloud (sic): A use case in MRI connectomics. *Giga Science*, 6(5):1–10, 2017.

[84] Noona Kiuru, William J Burk, Brett Laursen, Katariina Salmela-Aro, and Jari-Erik Nurmi. 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, 2010.

[85] Achim Klenke. *Probability theory: A Comprehensive Course*. Springer Science & Business Media, 2013.

[86] Eric D Kolaczyk and Gábor Csárdi. *Statistical analysis of network data with R*, volume 65. Springer, 2014.

[87] Bryan Kolb and Brenda Milner. Performance of complex arm and facial movements after focal brain lesions. *Neuropsychologia*, 19(4):491–503, 1981.

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

[89] Minjung Kyung, Jeff Gill, Malay Ghosh, George Casella, et al. Penalized regression, standard errors, and bayesian lassos. *Bayesian Analysis*, 5(2):369–411, 2010.

[90] John W Lau and Peter J Green. Bayesian model-based clustering procedures. *Journal of Computational and Graphical Statistics*, 16(3):526–558, 2007.

[91] Bing Li, Min Kyung Kim, and Naomi Altman. On dimension folding of matrix-or array-valued statistical objects. *The Annals of Statistics*, pages 1094–1121, 2010.

[92] Hanning Li and Debdeep Pati. Variable selection using shrinkage priors. *Computational Statistics & Data Analysis*, 107:107–119, 2017.

[93] YaPeng Li, Yuanyuan Qin, Xi Chen, and Wei Li. Exploring the functional brain network of alzheimer's disease: based on the computational experiment. *PloS one*, 8(9):e73186, 2013.

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

[95] Xi Luo. High dimensional low rank and sparse covariance matrix estimation via convex minimization. *Arxiv preprint*, 2011.

[96] Dean Lusher, Johan Koskinen, and Garry Robins. *Exponential random graph models for social networks: Theory, methods, and applications*. Cambridge University Press, 2012.

[97] Enes Makalic and Daniel F Schmidt. A simple sampler for the horseshoe estimator. *IEEE Signal Processing Letters*, 23(1):179–182, 2015.

[98] Ryan Martin, Raymond Mess, Stephen G Walker, et al. Empirical bayes posterior concentration in sparse high-dimensional linear models. *Bernoulli*, 23(3):1822–1847, 2017.

[99] Laurie Miller and Brenda Milner. Cognitive risk-taking after frontal or temporal

lobectomy-II. The synthesis of phonemic and semantic information. *Neuropsychologia*, 23(3):371–379, 1985.

[100] S. Minhas, P.D. Hoff, and M.D. Ward. Influence networks in international relations. 2017.

[101] Peter Muller, Giovanni Parmigiani, and Kenneth Rice. Fdr and bayesian multiple comparisons rules. 2006.

[102] Katherine L Narr, Roger P Woods, Paul M Thompson, Philip Szeszko, Delbert Robinson, Teodora Dimtcheva, Mala Gurbani, Arthur W Toga, and Robert M Bilder. Relationships between iq and regional cortical gray matter thickness in healthy adults. *Cerebral cortex*, 17(9):2163–2171, 2006.

[103] Mark Newman. *Networks: An Introduction*. Oxford University Press, 2010.

[104] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[105] N M K Niezink and T A B Snijders. Co-evolution of social networks and continuous actor attributes. 2016.

[106] Krzysztof Nowicki and Tom A B Snijders. Estimation and prediction for stochastic block structures. *Journal of the American Statistical Association*, 96(455):1077–1087, 2001.

[107] Kim TE Olde Dubbelink, Arjan Hillebrand, Diederick Stoffers, Jan Berend Deijen, Jos WR Twisk, Cornelis J Stam, and Henk W Berendse. Disrupted brain network

topology in parkinson's disease: a longitudinal magnetoencephalography study. *Brain*, 137(1):197–207, 2013.

[108] Hae-Jeong Park and Karl Friston. Structural and functional brain networks: from connections to cognition. *Science*, 342(6158):1238411, 2013.

[109] Trevor Park and George Casella. The Bayesian lasso. *Journal of the American Statistical Association*, 103(482):681–686, 2008.

[110] Nicholas G Polson and James G Scott. Shrink globally, act locally: Sparse bayesian regularization and prediction. *Bayesian Statistics*, 9:501–538, 2010.

[111] Nicholas G Polson, James G Scott, and Jesse Windle. Bayesian inference for logistic models using pólya–gamma latent variables. *Journal of the American statistical Association*, 108(504):1339–1349, 2013.

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

[113] Jesús D Arroyo Relión, Daniel Kessler, Elizaveta Levina, and Stephan F Taylor. Network classification with applications to brain connectomics. *arXiv preprint arXiv:1701.08140*, 2017.

[114] Jonas Richiardi, Hamdi Eryilmaz, Sophie Schwartz, Patrik Vuilleumier, and Dimitri Van De Ville. Decoding brain states from fmri connectivity graphs. *Neuroimage*, 56(2):616–626, 2011.

[115] Garry Robins, Tom Snijders, Peng Wang, Mark Handcock, and Philippa Pattison. Recent developments in exponential random graph (p*) models for social networks. *Social Networks*, 29(2):192–215, 2007.

[116] Abel Rodriguez, David B Dunson, and Alan E Gelfand. The nested dirichlet process. *Journal of the American Statistical Association*, 103(483):1131–1154, 2008.

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

[118] Mikail Rubinov and Olaf Sporns. Complex network measures of brain connectivity: uses and interpretations. *Neuroimage*, 52(3):1059–1069, 2010.

[119] Jayaram Sethuraman. A constructive definition of dirichlet priors. *Statistica sinica*, pages 639–650, 1994.

[120] Cosma Rohilla Shalizi and Andrew C Thomas. Homophily and contagion are generically confounded in observational social network studies. *Sociological methods & research*, 40(2):211–239, 2011.

[121] Maksim G Sharaev, Viktoria V Zavyalova, Vadim L Ushakov, Sergey I Kartashov, and Boris M Velichkovsky. Effective connectivity within the default mode network: dynamic causal modeling of resting-state fmri data. *Frontiers in human neuroscience*, 10:14, 2016.

[122] Phillip R Shaver and Kelly A Brennan. Attachment styles and the” big five” personal-

ity traits: Their connections with each other and with romantic relationship outcomes. *Personality and Social Psychology Bulletin*, 18(5):536–545, 1992.

[123] Kennon M Sheldon, Richard M Ryan, Laird J Rawsthorne, and Barbara Ilardi. Trait self and true self: Cross-role variation in the big-five personality traits and its relations with psychological authenticity and subjective well-being. *Journal of personality and social psychology*, 73(6):1380, 1997.

[124] David A Shoham, Ross Hammond, Hazhir Rahmandad, Youfa Wang, and Peter Hovmand. Modeling social norms and social influence in obesity. *Current Epidemiology Reports*, 2(1):71–79, 2015.

[125] Robin Sibson. Studies in the robustness of multidimensional scaling: Procrustes statistics. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 234–238, 1978.

[126] Tom Snijders, Christian Steglich, and Michael Schweinberger. *Modeling the coevolution of networks and behavior*. https://s3.amazonaws.com/academia.edu.documents, 2007.

[127] Tom AB Snijders. The statistical evaluation of social network dynamics. *Sociological Methodology*, 31(1):361–395, 2001.

[128] Qifan Song and Faming Liang. Nearly optimal bayesian shrinkage for high dimensional regression. *arXiv preprint arXiv:1712.08964*, 2017.

[129] Ashwin Srinivasan, Stephen H Muggleton, Michael JE Sternberg, and Ross D King.

Theories for mutagenicity: A study in first-order and feature-based induction. *Artificial Intelligence*, 85(1-2):277–299, 1996.

[130] Cornelis J Stam. Modern network science of neurological disorders. *Nature Reviews Neuroscience*, 15(10):683, 2014.

[131] DT Stuss, P Ely, H Hugenholtz, MT Richard, S LaRochelle, CA Poirier, and I Bell. Subtle neuropsychological deficits in patients with good recovery after closed head injury. *Neurosurgery*, 17(1):41–47, 1985.

[132] Yee Whye Teh, Dilan Grür, and Zoubin Ghahramani. Stick-breaking construction for the indian buffet process. In *Artificial Intelligence and Statistics*, pages 556–563, 2007.

[133] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[134] Stéphanie L Van Der Pas, Bas JK Kleijn, Aad W Van Der Vaart, et al. The horseshoe estimator: Posterior concentration around nearly black vectors. *Electronic Journal of Statistics*, 8(2):2585–2618, 2014.

[135] S Vichy N Vishwanathan, Nicol N Schraudolph, Risi Kondor, and Karsten M Borgwardt. Graph kernels. *Journal of Machine Learning Research*, 11(Apr):1201–1242, 2010.

[136] Joshua T Vogelstein, William Gray Roncal, R Jacob Vogelstein, and Carey E Priebe. Graph classification using signal-subgraphs: Applications in statistical connectomics. *IEEE transactions on pattern analysis and machine intelligence*, 35(7):1539–1551, 2013.

[137] Lu Wang, Daniele Durante, Rex E Jung, and David B Dunson. Bayesian network–response regression. *Bioinformatics*, 33(12):1859–1866, 2017.

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

[139] Michael D Ward and Peter D Hoff. Persistent patterns of international commerce. *Journal of Peace Research*, 44(2):157–175, 2007.

[140] Stanley Wasserman and Katherine Faust. *Social Network Analysis: Methods and Applications*, volume 8. Cambridge University Press, 1994.

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

[142] Duncan J Watts and Peter Dodds. Threshold models of social influence. *The Oxford Handbook of Analytical Sociology*, pages 475–497, 2009.

[143] Ran Wei and Subhashis Ghosal. Contraction properties of shrinkage priors in logistic regression. *Preprint at http://www4. stat. ncsu. edu/˜ ghoshal/papers*, 2017.

[144] Eric P Xing, Wenjie Fu, Le Song, et al. A state-space mixed membership blockmodel for dynamic network tomography. *The Annals of Applied Statistics*, 4(2):535–566, 2010.

[145] Youngwoo Bryan Yoon, Won-Gyo Shin, Tae Young Lee, Ji-Won Hur, Kang Ik K Cho, William Seunghyun Sohn, Seung-Goo Kim, Kwang-Hyuk Lee, and Jun Soo Kwon. Brain

structural networks associated with intelligence and visuomotor ability. *Scientific reports*, 7(1):2177, 2017.

[146] Jie Zhang, Wei Cheng, ZhengGe Wang, ZhiQiang Zhang, WenLian Lu, GuangMing Lu, and Jianfeng Feng. Pattern classification of large-scale functional brain networks: identification of informative neuroimaging markers for epilepsy. *PloS one*, 7(5):e36733, 2012.

[147] Hua Zhou, Lexin Li, and Hongtu Zhu. Tensor regression with applications in neuroimaging data analysis. *Journal of the American Statistical Association*, 108(502):540–552, 2013.