2019

Bayesian Approaches For Modeling Variation

Gemma Elyse Moran
University of Pennsylvania, gem.e.moran@gmail.com

Bayesian Approaches For Modeling Variation

Abstract
A core focus of statistics is determining how much of the variation in data may be attributed to the signal of interest, and how much to noise. When the sources of variation are many and complex, a Bayesian approach to data analysis offers a number of advantages. In this thesis, we propose and implement new Bayesian methods for modeling variation in two general settings. The first setting is high-dimensional linear regression where the unknown error variance is also of interest. Here, we show that a commonly used class of conjugate shrinkage priors can lead to underestimation of the error variance. We then extend the Spike-and-Slab Lasso (SSL, Rockova and George, 2018) to the unknown variance case, using an alternative, independent prior framework. This extended procedure outperforms both the fixed variance approach and alternative penalized likelihood methods on both simulated and real data.

For the second setting, we move from univariate response data where the predictors are known, to multivariate response data in which potential predictors are unobserved. In this setting, we first consider the problem of biclustering, where a motivating example is to find subsets of genes which have similar expression in a subset of patients. For this task, we propose a new biclustering method called Spike-and-Slab Lasso Biclustering (SSLB). SSLB utilizes the SSL prior to find a doubly-sparse factorization of the data matrix via a fast EM algorithm. Applied to both a microarray dataset and a single-cell RNA-sequencing dataset, SSLB recovers biologically meaningful signal in the data.

The second problem we consider in this setting is nonlinear factor analysis. The goal here is to find low-dimensional, unobserved ``factors'' which drive the variation in the high-dimensional observed data in a potentially nonlinear fashion. For this purpose, we develop factor analysis BART (faBART), an MCMC algorithm which alternates sampling from the posterior of (a) the factors and (b) a functional approximation to the mapping from the factors to the data. The latter step utilizes Bayesian Additive Regression Trees (BART, Chipman et al., 2010). On a variety of simulation settings, we demonstrate that with only the observed data as the input, faBART is able to recover both the unobserved factors and the nonlinear mapping.

Degree Type
Dissertation

Degree Name
Doctor of Philosophy (PhD)

Graduate Group
Statistics

First Advisor
Edward I. George

Subject Categories
Statistics and Probability

BAYESIAN APPROACHES FOR MODELING VARIATION

Gemma E. Moran

A DISSERTATION

in

Statistics

For the Graduate Group in Managerial Science and Applied Economics

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2019

Supervisor of Dissertation

Edward I. George, Universal Furniture Professor, Professor of Statistics

Graduate Group Chairperson

Catherine M. Schrand, Celia Z. Moh Professor, Professor of Accounting

Dissertation Committee

Veronika Ročková, Assistant Professor of Econometrics and Statistics, Chicago Booth

Nancy R. Zhang, Professor of Statistics

Shane Jensen, Associate Professor of Statistics

BAYESIAN APPROACHES FOR MODELING VARIATION

Dedicated to my grandparents:

Mary and Bernard Curtin, & Zita and Geoff Moran

ACKNOWLEDGEMENT

First and foremost, I would like to thank my advisor, Ed George. Your incredible insight and statistical intuition have been instrumental in shaping both this thesis and my development as a statistician. Moreover, your boundless enthusiasm and encouragement always led me to come away from meetings with a renewed positive outlook and energy, especially when I was at my most stressed.

Thank you to my committee members: Veronika Ročková, Shane Jensen and Nancy Zhang. Veronika - your tremendous work ethic and intellect have been an inspiration and I have learnt so much from working with you over the course of my PhD. Shane - thank you for being a fantastic teacher, for your wealth of knowledge of all things Bayesian and for all your encouragement. Nancy - thank you for fostering my interest in both genomics and dimensionality reduction - many of my current research interests were inspired from attending your reading group.

Next, an enormous thank you to the Wharton Statistics Department. On my prospective PhD visit from Australia five years ago, I was struck by the warmth and collegiality of the department, a feature which has remained constant over my five years here. To the faculty that I have been fortunate enough to have as teachers, thank you for your dedication and helping me grow as a researcher and statistician. To the staff - thank you for all you do to keep the department running smoothly, and for all your generous help, from reserving classrooms for TA sessions, to navigating PhD forms and for always being there with kind words and/or a donut.

To our PhD cohort - Raiden Hasegawa, Bikram Karmakar, Justin Khim and Linjun Zhang - it has been a privilege to call you my classmates. I will always have fond memories of the first year office where we would alternate doing our probability homework with playing putt putt golf. To the Bayesian gang - Cecilia Balocchi and Sameer Deshpande - our reading group and discussions were always both intellectually stimulating and a lot of fun - I look

forward to future collaborations in the years to come! To the Statistics PhD students, both present and graduated - thank you for all the laughs and great conversations, both statistical and otherwise, over our daily lunches in the department, board game nights, drinks at Cav's and barbecues.

I feel so lucky to have so many amazing friends from all over the world. To my Philadelphia friends - you have made my five years here enriching personally as well as intellectually. Thank you especially to Lesley Meng, Cecilia Balocchi, Elica Dhundia McCarthy, Sameer Deshpande, Kathy Li and Daniela Schmitt for always being there for me, from the stressful times to the celebratory times, and everything in between. Finding balance during the PhD was so important - thanks to the erstwhile running club: Justin Khim, Kathy Li, Lesley Meng, Min Xu; the bowling crew: Justin Chiu, Colman Humphrey, Matt Olson; the bridge club: Eric Baxter, Ashley Baker, Ling Lin; and the vegan food gang: Edward Chang and Lesley Meng. To my Australian friends: true friendship is when it feels like just yesterday since you've last seen each other, even if it has been a year - thank you.

To Eric Baxter - thank you for always being able to make me laugh, supporting me, being there for me in tough times, and for all of our adventures, even when I forget to read contour maps and plan a hike up an (almost) vertical mountain slope.

Finally, thank you to my family. As the quote goes - you have given me roots and wings. Roots, to ground me and give me a sense of belonging and identity, and wings, to give me the courage to go out in the wider world. Thank you especially to my parents: Gabrielle and Paul, and Greg and Trish: even though I am on the other side of the world, it gives me such strength knowing you are always there for me, just a call away. Nick - thank you for being the best big brother. Margaret - thank you for being my "Philly Mum" - it has been so wonderful to have had the opportunity to connect here and I am so grateful for all your care, encouragement, and showing me this great city. Your equanimity, generosity and kindness of spirit are truly remarkable and something I aspire towards.

In the week before submitting this thesis, my uncle Simon tragically passed away. Si - thank you for being such a great uncle, from introducing me to my favorite sci-fi and fantasy novels as a kid, to always having a witty response or joke for every situation, and always encouraging my studies. You, Grandma and Bernard are much loved and missed.

ABSTRACT

BAYESIAN APPROACHES FOR MODELING VARIATION

Gemma E. Moran

Edward I. George

A core focus of statistics is determining how much of the variation in data may be attributed to the signal of interest, and how much to noise. When the sources of variation are many and complex, a Bayesian approach to data analysis offers a number of advantages. In this thesis, we propose and implement new Bayesian methods for modeling variation in two general settings. The first setting is high-dimensional linear regression where the unknown error variance is also of interest. Here, we show that a commonly used class of conjugate shrinkage priors can lead to underestimation of the error variance. We then extend the Spike-and-Slab Lasso (SSL, Ročková and George, 2018) to the unknown variance case, using an alternative, independent prior framework. This extended procedure outperforms both the fixed variance approach and alternative penalized likelihood methods on both simulated and real data.

For the second setting, we move from univariate response data where the predictors are known, to multivariate response data in which potential predictors are unobserved. In this setting, we first consider the problem of biclustering, where a motivating example is to find subsets of genes which have similar expression in a subset of patients. For this task, we propose a new biclustering method called Spike-and-Slab Lasso Biclustering (SSLB). SSLB utilizes the SSL prior to find a doubly-sparse factorization of the data matrix via a fast EM algorithm. Applied to both a microarray dataset and a single-cell RNA-sequencing dataset, SSLB recovers biologically meaningful signal in the data.

The second problem we consider in this setting is nonlinear factor analysis. The goal here is to find low-dimensional, unobserved "factors" which drive the variation in the high-

dimensional observed data in a potentially nonlinear fashion. For this purpose, we develop factor analysis BART (faBART), an MCMC algorithm which alternates sampling from the posterior of (a) the factors and (b) a functional approximation to the mapping from the factors to the data. The latter step utilizes Bayesian Additive Regression Trees (BART, Chipman et al., 2010). On a variety of simulation settings, we demonstrate that with only the observed data as the input, faBART is able to recover both the unobserved factors and the nonlinear mapping.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF ILLUSTRATIONS

CHAPTER 1 : Introduction

Statistics has been said to be the science of variation. A core problem of any statistical analysis is determining how much of the observed variation in the data can be attributed to the signal of interest, and how much to noise. Understanding and modeling these sources of variation is crucial for both inferring the size and significance of the signal, and predicting future realizations of the data.

In this thesis, we first consider the problem of high-dimensional linear regression where the unknown noise variance is also of interest. We then move from this univariate response setting where the predictors are known, to the multivariate response setting in which potential predictors are unobserved. Although this lack of predictors presents an even greater challenge, this multivariate setting also presents a tremendous opportunity to learn about the covariation of the responses. In this setting, the covariation itself is often a signal of interest; in gene expression data, for example, finding sets of responses which exhibit similar behavior (that is, covary) can be an indication that these responses are driven by the same underlying biological process.

To tackle the challenge of modeling variation in these settings, we adopt a Bayesian perspective. A Bayesian approach to modeling begins with specifying a data generating process, or model. Within this model, the Bayesian paradigm allows for the coherent inclusion of multiple sources of variation which give rise to the observed data. In treating the parameters of this model as themselves random, a Bayesian approach confers a number of advantages. Firstly, it provides uncertainty quantification for the parameters via their posterior distribution. Secondly, by treating the parameters as random instead of fixed, the parameters are able to adapt to the data at hand. Finally, Bayesian analyses allow for the "borrowing of strength" across multiple observations to ultimately yield parameter estimates which are less susceptible to noise.

In this thesis, we propose and deploy new Bayesian methods to solve specific problems in

both the univariate linear regression setting, and the multiple response setting where no predictors are observed.

In Chapter 2, we consider the problem of simultaneously estimating the regression coefficients and error variance in the high-dimensional Gaussian linear model. A common Bayesian approach to modeling the error variance is to use a conjugate shrinkage prior framework. Here, however, we show that these commonly used conjugate shrinkage priors can actually have detrimental consequences for error variance estimation. Such priors are often motivated by the invariance argument of Jeffreys (1961). Revisiting this work, however, we highlight a caveat that Jeffreys himself noticed; namely that biased estimators can result from inducing dependence between parameters *a priori*. In a similar way, we show that conjugate priors for linear regression, which induce prior dependence, can lead to such underestimation in the Bayesian high-dimensional regression setting. Following Jeffreys, we recommend as a remedy to treat regression coefficients and the error variance as independent *a priori*.

In the latter half of Chapter 2, we then extend the Spike-and-Slab Lasso of Ročková and George (2018) to the unknown variance case, using an independent prior framework. This extended procedure outperforms both the fixed variance approach and alternative penalized likelihood methods on simulated data. On the protein activity dataset of Clyde and Parmigiani (1998), the Spike-and-Slab Lasso with unknown variance achieves lower cross-validation error than alternative penalized likelihood methods, demonstrating the gains in predictive accuracy afforded by simultaneous error variance estimation.

In the next part of this thesis, we move to the multivariate setting, where for each individual, we observe many responses, or features. Unlike Chapter 2, however, we now do not observe any potential predictors for these responses.

In Chapter 3, we consider the problem of finding small sets of individuals which covary over only a small set of their features; these sets of both individuals and features are then

referred to as biclusters. In this way, biclustering methods differ from traditional clustering methods, which find groups of individuals that are similar over their *entire* set of features. Motivating applications for biclustering include genomics data, where the goal is to cluster patients or samples by their gene expression profiles; and recommender systems, which seek to group customers based on their product preferences. More precisely, biclusters of interest are often assumed to manifest as rank-1 submatrices of the data matrix. This submatrix detection problem can be viewed as a factor analysis problem in which both the factors and loadings are sparse.

We propose a new biclustering method called Spike-and-Slab Lasso Biclustering (SSLB). SSLB utilizes the Spike-and-Slab Lasso of Ročková and George (2018) to find a doubly-sparse factorization of the data matrix. SSLB also incorporates an Indian Buffet Process prior to automatically choose the number of biclusters. Many biclustering methods make assumptions about the size of the latent biclusters, either assuming that the biclusters are all of the same size, or that the biclusters are either very large or very small. In contrast, SSLB can adapt to find biclusters which have a continuum of sizes. SSLB is implemented via a fast Expectation-Maximization (EM) algorithm with a variational step. In a variety of simulation settings, SSLB outperforms other biclustering methods. We apply SSLB to both a microarray dataset and a single-cell RNA-sequencing dataset and highlight that SSLB can recover biologically meaningful signal in the data.

In Chapter 4, we again consider the unsupervised multivariate response setting with the goal of finding low-dimensional "factors" which drive the variation in the observed data. Unlike in Chapter 3, however, we now relax the assumption that the observed data is *linearly* related to the unobserved factors. This adds an additional layer of complexity to the problem: we need to both estimate the unobserved factors, and the mapping between the factors and observed data. To accomplish this task, we develop a Markov Chain Monte Carlo (MCMC) algorithm which alternates between sampling from the posterior of the factors and a functional approximation to the mapping. The latter step utilizes Bayesian

Additive Regression Trees (BART), introduced by Chipman et al. (2010). We refer to our method as Factor Analysis BART (faBART). On a variety of simulation settings, we demonstrate that with only the observed data as the input, faBART is able to recover both the unobserved factors and the nonlinear mapping. We then develop tempered faBART, a modification of faBART which includes a tempering step to allow the algorithm to more easily detect structure for data visualization. On two canonical datasets for visualization, we highlight that tempered faBART can find meaningful low-dimensional embeddings.

CHAPTER 2 : Variance Priors

2.1. Introduction

Consider the classical linear regression model

$$\mathbf{Y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \sim N_n(0, \sigma^2 \mathbf{I}_n) \tag{2.1}$$

where $\mathbf{Y} \in \mathbb{R}^n$ is a vector of responses, $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_p] \in \mathbb{R}^{n \times p}$ is a fixed regression matrix of p potential predictors, $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_p)^T \in \mathbb{R}^p$ is a vector of unknown regression coefficients and $\boldsymbol{\varepsilon} \in \mathbb{R}^n$ is the noise vector of independent normal random variables with σ^2 as their unknown common variance.

When $\boldsymbol{\beta}$ is sparse so that most of its elements are zero or negligible, finding the non-negligible elements of $\boldsymbol{\beta}$, the so-called variable selection problem, is of particular importance. Whilst this problem has been studied extensively from both frequentist and Bayesian perspectives, much less attention has been given to the simultaneous estimation of the error variance σ^2. Accurate estimates of σ^2 are important to discourage fitting the noise beyond the signal, thereby helping to mitigate overfitting of the data. Variance estimation is also essential in uncertainty quantification for inference and prediction.

In the frequentist literature, the question of estimating the error variance in our setting has begun to be addressed with papers including the scaled Lasso (Sun and Zhang, 2012) and the square-root Lasso (Belloni et al., 2014). Contrastingly, in the Bayesian literature, the error variance has been fairly straightforwardly estimated by including σ^2 in prior specifications. Despite this conceptual simplicity, the majority of theoretical guarantees for Bayesian procedures restrict attention to the case of known σ^2, as there is not a generally agreed upon prior specification when σ^2 is unknown. More specifically, priors on $\boldsymbol{\beta}$ and σ^2

Adapted from a research article:
Moran, G. E., Ročková, V. and George, E. I. (2019) "Variance Prior Forms for High-Dimensional Bayesian Variable Selection" *Bayesian Analysis (Accepted)*

are typically introduced in one of two ways: either via a conjugate prior framework or via an independence prior framework.

Conjugate priors have played a major role in regression analyses. The conjugate prior framework for (2.1) begins with specifying a prior on $\boldsymbol{\beta}$ that depends on σ^2 as follows:

$$\boldsymbol{\beta}|\sigma^2 \sim N(0, \sigma^2\mathbf{V}), \tag{2.2}$$

where $\mathbf{V}$ may be fixed or random. This prior (2.2) results in a Gaussian posterior for $\boldsymbol{\beta}$ and as such is conjugate. To complete the framework, σ^2 is assigned an inverse-gamma (or equivalently scaled-inverse-χ^2) prior. A common choice in this regard is the right-Haar prior for the location-scale group (Berger et al., 1998):

$$\pi(\sigma) \propto 1/\sigma. \tag{2.3}$$

Whilst the right-Haar prior is improper, it can be viewed as the limit of an inverse-gamma density. When combined with (2.2), the prior (2.3) results in an inverse-gamma posterior for σ^2 and as such it behaves as a conjugate prior. Prominent examples that utilize the above conjugate prior framework include:

- Bayesian ridge regression priors, with $\mathbf{V} = \tau^2\mathbf{I}$;
- Zellner's g-prior, with $\mathbf{V} = g(\mathbf{X}^T\mathbf{X})^{-1}$; and
- Gaussian global-local shrinkage priors, with $\mathbf{V} = \tau^2\Lambda$, for $\Lambda = \text{diag}\{\lambda_j\}_{j=1}^p$.

We note that the conjugate prior framework refers only to the prior characterization of $\boldsymbol{\beta}$ and σ^2, and allows for any prior specification on subsequent hyper-parameters such as g and τ^2 which do not appear in the likelihood.

A main reason for the popularity of the conjugate prior framework is that it often allows for marginalization over $\boldsymbol{\beta}$ and σ^2, resulting in closed form expressions for Bayes factors and

updates of posterior model probabilities. This allowed for analyses of the model selection consistency (Bayarri et al., 2012) as well as more computationally efficient MCMC algorithms (George and McCulloch, 1997). Despite these advantages, however, the conjugate prior framework is not innocuous for variance estimation, as we will show in this work.

Alternatively to the conjugate prior framework, one might treat $\boldsymbol{\beta}$ and σ^2 as independent *a priori*. The formulation corresponding to (2.2) for this independence prior framework is:

$$\begin{aligned} \boldsymbol{\beta} &\sim N(0, \mathbf{V}), \\ \pi(\sigma) &\propto 1/\sigma. \end{aligned} \tag{2.4}$$

Note that the prior characterization (2.4) does not yield a normal inverse-gamma posterior distribution on $(\boldsymbol{\beta}, \sigma^2)$ and as such is not conjugate.

In addition to the above prior frameworks, Bayesian methods for variable selection can be further categorized by the way they treat negligible predictors. Discrete component Bayesian methods for variable selection exclude negligible predictors from consideration, adaptively reducing the dimension of $\boldsymbol{\beta}$. Examples of such discrete component methods include spike-and-slab priors where the "spike" distribution is a point-mass at zero (Mitchell and Beauchamp, 1988). In contrast, continuous Bayesian methods for variable selection shrink, rather than exclude, negligible predictors and as such $\boldsymbol{\beta}$ remains p-dimensional (George and McCulloch, 1993; Polson and Scott, 2010; Ročková and George, 2014).

In this chapter, we show that for continuous Bayesian variable selection methods, the conjugate prior framework can result in underestimation of the error variance when: (i) the regression coefficients $\boldsymbol{\beta}$ are sparse; and (ii) p is of the same order as, or larger than n. Intuitively, conjugate priors implicitly add p "pseudo-observations" to the posterior which can distort inference for the error variance when the true number of non-zero $\boldsymbol{\beta}$ is much smaller than p. This is not the case for discrete component methods which adaptively reduce the size of $\boldsymbol{\beta}$. To avoid the underestimation problem in the continuous case, we recommend

the use of independent priors on $\boldsymbol{\beta}$ and σ^2. Further, we extend the Spike-and-Slab Lasso of Ročková and George (2018) to the unknown variance case with an independent prior formulation, and highlight the performance gains over the known variance case via a simulation study. On the protein activity dataset of Clyde and Parmigiani (1998), we demonstrate the benefit of simultaneous variance estimation for both variable selection and prediction.

It is important to note the difference in the scope of this work with previous work on variance priors, including Gelman (2004); Bayarri et al. (2012); Liang et al. (2008). Here, we are focused on the estimation of the error variance, σ^2. In contrast, the aforementioned works are concerned with the choice of priors for hyper-parameters which do not appear in the likelihood, i.e. the g in the g-prior, and τ^2 and λ_j^2 for global-local priors. We recognize the importance of the choice of these priors for Bayesian variable selection; however, the focus of this chapter is the prior choice for the error variance in conjunction with variable selection.

We also note that our discussion considers only Gaussian related prior forms for the regression coefficients. Despite this seemingly limited scope, we note that the majority of priors used in Bayesian variable selection can be cast as a scale-mixture of Gaussians (Polson and Scott, 2010), and that popular frequentist procedures such as the Lasso and variants thereof also fall under this framework.

The chapter is structured as follows. In Section 2, we discuss invariance arguments for conjugate priors and draw connections with Jeffreys priors. We then highlight situations where we ought to depart from Jeffreys priors; namely, in multivariate situations. In Section 3, we take Bayesian ridge regression as an example to highlight why conjugate priors can be a poor choice. In Section 4, we draw connections between Bayesian regression and concurrent developments with variance estimation in the penalized likelihood literature. In Section 5, we examine the mechanisms of the Gaussian global-local shrinkage framework and illustrate why they can be incompatible with the conjugate prior structure. In Section 6, we consider the Spike-and-Slab Lasso of Ročková and George (2018) and highlight

how the conjugate prior yields poor estimates of the error variance. We then extend the procedure to include the unknown variance case using an independent prior structure and demonstrate via simulation studies how this leads to performance gains over not only the known variance case, but a variety of other variable selection procedures. In Section 7, we apply the Spike-and-Slab Lasso with unknown variance to the protein activity dataset of Clyde and Parmigiani (1998), highlighting the improved predictive performance afforded by simultaneous variance estimation. We conclude with a discussion in Section 8.

2.2. Invariance Criteria

A common argument used in favor of the conjugate prior for Bayesian linear regression is that it is invariant to scale transformations of the response (Bayarri et al., 2012). That is, the regression coefficients depend *a priori* on σ^2 in a "scale-free way" through

$$\pi(\boldsymbol{\beta}|\sigma^2) = \frac{1}{\sigma^p} h(\boldsymbol{\beta}/\sigma), \tag{2.5}$$

for some proper density function $h(x)$. This means that the units of measurement used for the response do not affect the resultant estimates; for example, if $\mathbf{Y}$ is scaled by a factor of c, one would expect that the estimates for the regression coefficients, $\boldsymbol{\beta}$, and error variance, σ^2, should also be scaled by c.

A more general principle of invariance was proposed by Jeffreys (1961) in his seminal work, *The Theory of Probability*, a reference which is also sometimes given for the conjugate prior. In this section, we examine the original invariance argument of Jeffreys (1961) and highlight a caveat with this principle that the author himself noted; namely that it should be avoided in multivariate situations. We then draw connections between this suboptimal multivariate behavior and the conjugate prior framework, ultimately arguing similarly to Jeffreys that we should treat the mean and variance parameters as independently *a priori*.

2.2.1. Jeffreys Priors

For a parameter α, the Jeffreys prior is

$$\pi(\alpha) \propto |I(\alpha)|^{1/2}, \tag{2.6}$$

where $I(\alpha)$ is the Fisher information matrix. The main motivation given by Jeffreys (1961) for these priors was that they are invariant for all nonsingular transformations of the parameters. This property appeals to intuition regarding objectivity; ideally, the prior information we decide to include should not depend upon the choice of the parameterization, which itself is arbitrary.

Despite this intuitively appealing property, the following problem with this principle was spotted in the original work of Jeffreys (1961) and later re-emphasized by Robert et al. (2009) in their revisit of the work. Consider the normal means model

$$Y_i \sim N(\mu_i, \sigma^2), \quad i = 1, \ldots, n$$

where the n-dimensional mean is denoted by $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)$. If we treat the parameters $\boldsymbol{\mu}$ and σ independently, the Jeffreys prior is $\pi(\boldsymbol{\mu}, \sigma) \propto 1/\sigma$. However, if the parameters are considered jointly, the Jeffreys prior is $\pi(\boldsymbol{\mu}, \sigma) \propto 1/\sigma^{n+1}$. In effect, by considering the parameters jointly as opposed to independently, we are implicitly including additional "pseudo-observations" of σ^2 and consequently distorting our estimates of the error variance.

This "pseudo-observation" interpretation can be seen explicitly in the conjugate form of the Jeffreys prior for a Gaussian likelihood. The joint Jeffreys prior $\pi(\boldsymbol{\mu}, \sigma) \propto 1/\sigma^{n+1}$ is an improper inverse-gamma prior with shape parameter, $n/2$, and scale parameter zero. As the prior is conjugate, the posterior distribution for the variance is also inverse-gamma:

$$\pi(\sigma^2|\mathbf{Y}, \boldsymbol{\mu}) \sim IG\left(\frac{n}{2} + \frac{n}{2},\ 0 + \frac{\sum_{i=1}^{n}(Y_i - \mu_i)^2}{2}\right) \tag{2.7}$$

where the first term of both the shape and scale parameters in (2.7) are the prior hyper-parameters. Thus, the dependent Jeffreys prior can be thought of as encoding knowledge of σ^2 from a previous experiment where there were n observations which yielded a sample variance of zero. This results in the prior concentrating around zero for large n and will severely distort posterior estimates of σ^2. As we shall see later, this *dependent Jeffreys prior* for the parameters is in some cases akin to the conjugate prior framework in (2.2).

This prior dependence between the parameters is explicitly repudiated by Jeffreys (1961) who states (with notation changed to match ours): "in the usual situation in an estimation problem, μ and σ^2 are each capable of any value over a considerable range, and neither gives any appreciable information about the other. We should then take: $\pi(\mu, \sigma) = \pi(\mu)\pi(\sigma)$." That is, Jeffreys' remedy is to treat the parameters independently *a priori*, a recommendation which we also adopt. In addition, Jeffreys points out that a key problem with the joint Jeffreys prior is that it does not have the same reduction of degrees of freedom required by the introduction of additional nuisance parameters. We shall examine this phenomenon in more detail in Section 2.3 where we will discuss the consequences of using dependent Jefferys priors and other conjugate formulations in Bayesian linear regression.

We note a possible exception to this independence argument which is found later in *The Theory of Probability* where Jeffreys argues that for simple normal testing, the prior on μ under the alternative hypothesis should depend on σ^2. However, it is important to note that this recommendation is for the situation where μ is one-dimensional and so the underestimation phenomenon observed in (2.7) is not a problem. Given Jeffreys' earlier concerns regarding multivariate situations, it is unlikely he intended this dependence to generalize for higher dimensional μ.

2.3. Bayesian Regression

2.3.1. Prior Considerations

Consider again the classical linear regression model in (2.1). For a non-informative prior, it is common to use $\pi(\boldsymbol{\beta}, \sigma^2) \propto 1/\sigma^2$ (see, for example, Gelman et al., 2014). Similarly to our earlier discussion, this prior choice corresponds to multiplying the independent, Jeffreys priors for $\boldsymbol{\beta}$ and σ^2. In contrast, the joint Jeffreys prior would be $\pi(\boldsymbol{\beta}, \sigma^2) \propto 1/\sigma^{p+2}$. Let us now examine the estimates resulting from the former, independent Jeffreys prior. In this case, we have the following marginal posterior mean estimate for the error variance:

$$\mathbb{E}[\sigma^2|\mathbf{Y}] = \frac{\|\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2}{n - p - 2} \tag{2.8}$$

where $\widehat{\boldsymbol{\beta}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{Y}$ is the usual least squares estimator. We observe that the degrees of freedom adjustment, $n - p - 2$, naturally appears in the denominator.[1] This degrees of freedom adjustment does not occur with the joint Jeffreys prior where the marginal posterior mean is given by:

$$\mathbb{E}[\sigma^2|\mathbf{Y}] = \frac{\|\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2}{n - 2}. \tag{2.9}$$

For large p, this estimator will severely underestimate the error variance. Avoiding this, it is commonly accepted that the independent Jeffreys prior $\pi(\boldsymbol{\beta}, \sigma^2) \propto 1/\sigma^2$ should be the default non-informative prior in this setting.

There is no such clarity, however, in the use of conjugate priors for Bayesian linear regression. To add to this discourse, we show that these conjugate priors can suffer the same problem as the dependent Jeffreys priors and recommend, similarly to Jeffreys, that independent priors should be used instead. We make this point with the following example. A common

[1] Note that had we treated β_1 as an intercept and integrated it out with respect to a uniform prior, this term would be the usual $n - p - 1$.

conjugate prior choice for Bayesian linear regression is

$$\boldsymbol{\beta}|\sigma^2, \tau^2 \sim N_p(0, \sigma^2\tau^2\mathbf{I}). \tag{2.10}$$

For simplicity of exposition, in this section we consider the parameter τ^2 to be fixed, which corresponds to Bayesian ridge regression. In later sections we will consider the global-local shrinkage framework where τ^2 is assigned a prior.

With an additional non-informative prior $\pi(\sigma^2) \propto 1/\sigma^2$, we then have the joint prior

$$\pi(\boldsymbol{\beta}|\sigma^2)\pi(\sigma^2) = \pi(\boldsymbol{\beta}, \sigma^2) \propto \frac{1}{\sigma^{p+2}} \exp\left\{-\frac{1}{2\sigma^2\tau^2}\|\boldsymbol{\beta}\|^2\right\}. \tag{2.11}$$

Note again the σ^{p+2} in the denominator, similarly to the joint Jeffreys prior.

Instead of considering how $\boldsymbol{\beta}$ depends on σ^2 *a priori* as in (2.10), it is illuminating to consider the reverse: how this prior induces dependence of σ^2 on $\boldsymbol{\beta}$. From (2.11), the implicit conditional prior on σ^2 is given by

$$\sigma^2|\boldsymbol{\beta} \sim IG\left(\frac{p}{2}, \frac{\|\boldsymbol{\beta}\|^2}{2\tau^2}\right). \tag{2.12}$$

The mean of this inverse-gamma prior is approximately $\frac{1}{p}\|\boldsymbol{\beta}\|^2/\tau^2$. Heuristically, this term is of order $O(q/p)$, where q is the number of non-zero $\boldsymbol{\beta}$. When $\boldsymbol{\beta}$ is sparse and bounded with $q \ll p$, (2.12) will then transmit downward biasing information from $\boldsymbol{\beta}$ to σ^2. This intuition is formalized in Proposition 1, which shows that the implicit conditional prior on σ^2 concentrates around zero in regions where $\boldsymbol{\beta}$ is sparse.

Proposition 1. *Suppose* $\|\boldsymbol{\beta}\|_0 = q$ *and* $\max_j \beta_j^2 = K$ *for some constant* $K \in \mathbb{R}$. *Denote the true variance as* σ_0^2. *Then*

$$P\left(\sigma^2/\sigma_0^2 \geq \varepsilon \mid \boldsymbol{\beta}\right) \leq \frac{q}{p-2}\frac{K}{\tau^2}\frac{1}{\varepsilon\sigma_0^2}. \tag{2.13}$$

Proof. Proposition 1 follows from Markov's inequality and the bound $\|\boldsymbol{\beta}\|^2 \leq qK$. □

Proposition 1 implies that we can choose $0 < \varepsilon < 1$ such that as $q/p \to 0$, the prior places decreasing mass on values of σ^2 greater than $\varepsilon\sigma_0^2$. Thus, in regions of bounded sparse regression coefficients, the conjugate Gaussian prior can result in poor estimation of the true variance.

Further, from a more philosophical perspective, it is troubling that the error variance depends on the regression coefficients *a priori*, given that the noise is generally assumed to be independent of the signal and in particular the regression coefficients.

In the next section, we conduct a simulation study for the simple case of Bayesian ridge regression and show empirically how this implicit prior on σ^2 can distort estimates of the error variance.

2.3.2. The Failure of a Conjugate Prior

As an illustrative example, we take $n = 100$ and $p = 90$ and compare the least squares estimates of $\boldsymbol{\beta}$ and σ^2 to Bayesian ridge regression estimates with (i) the conjugate formulation with (2.10) and (ii) the independent prior formulation with

$$\pi(\boldsymbol{\beta}) \sim N_p(0, \tau^2\mathbf{I}). \tag{2.14}$$

For both Bayesian ridge regression procedures we use the non-informative error variance prior: $\pi(\sigma^2) \propto 1/\sigma^2$. The predictors $\mathbf{X}_i$, $i = 1, \ldots, p$ are generated as independent standard normal random variables. The true $\boldsymbol{\beta}_0$ is set to be sparse with only six non-zero elements; the non-zero coefficients are set to $\{-2.5, -2, -1.5, 1.5, 2, 2.5\}$. The response $\mathbf{Y}$ is generated according to (2.1) with the true variance being $\sigma^2 = 3$. We take $\tau = 10$ as known and highlight that this weakly informative choice leads to poor variance estimates in the conjugate prior framework. Whilst an empirical or fully Bayes approach for estimating τ^2 may be preferable for high-dimensional regression, it is troubling that the conjugate prior yields poor results for a simple example where $n > p$ and in which least squares and the independent prior formulation perform well.

The conjugate prior formulation allows for the exact expressions for the marginal posterior means of $\boldsymbol{\beta}$ and σ^2:

$$\mathbb{E}[\boldsymbol{\beta}|\mathbf{Y}] = [\mathbf{X}^T\mathbf{X} + \tau^{-2}\mathbf{I}]^{-1}\mathbf{X}^T\mathbf{Y} \tag{2.15}$$

$$\mathbb{E}[\sigma^2|\mathbf{Y}] = \frac{\mathbf{Y}^T[\mathbf{I} - \mathbf{H}_\tau]\mathbf{Y}}{n-2} \tag{2.16}$$

where $\mathbf{H}_\tau = \mathbf{X}[\mathbf{X}^T\mathbf{X} + \tau^{-2}\mathbf{I}]^{-1}\mathbf{X}^T$. Similarly to (2.9), the above marginal posterior mean for σ^2 does not incorporate a degrees of freedom adjustment and so we expect this estimator to underestimate the true error variance.

It is illuminating to observe the underestimation problem when considering the conditional posterior mean of σ^2, instead of the marginal:

$$\mathbb{E}[\sigma^2|\mathbf{Y}, \boldsymbol{\beta}] = \frac{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \|\boldsymbol{\beta}\|^2/\tau^2}{n+p-2}. \tag{2.17}$$

The additional p in the denominator here leads to severe underestimation of σ^2 when $\boldsymbol{\beta}$ is sparse and bounded as in Proposition 1 and p is of the same order as, or larger than, n, as discussed in the previous section. We note in passing that a value of τ^2 close to $\|\boldsymbol{\beta}\|^2/p\sigma^2$, which may be obtainable with an empirical or fully Bayes approach, would avoid this variance underestimation problem, as can be seen from (2.17).

This is in contrast to the conditional posterior mean for σ^2 using the independent prior formulation (2.4), which we also consider. This estimator is given by:

$$\mathbb{E}[\sigma^2|\mathbf{Y}, \boldsymbol{\beta}] = \frac{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2}{n-2}. \tag{2.18}$$

Here we do not observe a degrees of freedom adjustment because (2.18) is the *conditional* posterior mean, not the marginal. Earlier in (2.8) we considered the marginal posterior mean for the independent Jeffreys' prior which led to the $n-p-2$ in the denominator. For the marginal posterior means of $\boldsymbol{\beta}$ and σ^2, the independent prior formulation does not yield

closed form expressions. To compute these, we use a Gibbs sampler, the details of which may be found in Section 2.9.1 of the Appendix.

When τ^2 is large, the estimate of $\boldsymbol{\beta}$ for both the conjugate and independent formulations are almost exactly the least-squares estimate, $\widehat{\boldsymbol{\beta}} = [\mathbf{X}^T\mathbf{X}]^{-1}\mathbf{X}^T\mathbf{Y}$. However, the estimates of the variance σ^2 differ substantially.

In Figure 1, we display a boxplot of the estimates of σ^2 for (i) Least Squares, (ii) Conjugate Bayesian ridge regression, (iii) Zellner's prior:

$$\boldsymbol{\beta}|\sigma^2 \sim N(0, \sigma^2\tau^2[\mathbf{X}^T\mathbf{X}]^{-1}), \tag{2.19}$$

and (iv) Independent Bayesian ridge regression over 100 replications. Here, the estimates from least squares and the independent ridge are reasonably distributed around the truth. In sharp contrast, the estimates from the conjugate ridge and Zellner's prior consistently underestimate the error variance with medians of $\widehat{\sigma}^2 = 0.27$ and 0.55, respectively. This poor performance is a result of the bias induced by adding p "pseudo-observations" of σ^2 as discussed in Section 2.3.1, which also occurs for the Zellner prior.

In the above simulation study, we considered the posterior mean of σ^2 over many replications of the data. This allowed us to assess the variability of these point estimates in a frequentist sense. For a Bayesian perspective, we can also consider the entire marginal posterior distribution for σ^2. For the conjugate prior formulation, this posterior is given by:

$$\sigma^2|\mathbf{Y} \sim IG\left(\frac{n}{2}, \frac{\mathbf{Y}^T[\mathbf{I} - \mathbf{H}_\tau]\mathbf{Y}}{2}\right). \tag{2.20}$$

The above distribution (2.20) is tightly concentrated about the posterior mode, given by $\mathbf{Y}^T[\mathbf{I} - \mathbf{H}_\tau]\mathbf{Y}/(n+2)$, which suffers from the same underestimation phenomenon as the posterior mean (2.16). Thus, consideration of the posterior distribution of σ^2 will yield similar conclusions to our consideration of the posterior mean.

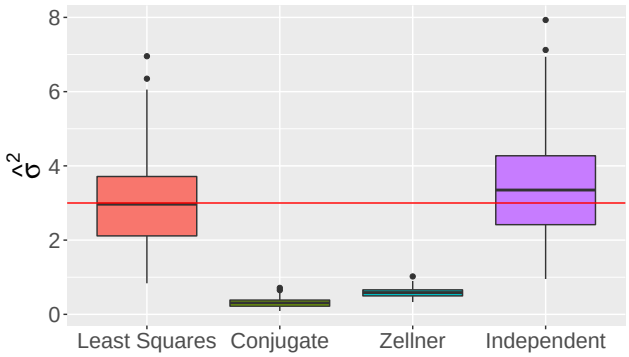


Figure 1: Estimated $\widehat{\sigma}^2$ for each procedure over 100 repetitions. The true $\sigma^2 = 3$ is the red horizontal line.

This phenomenon of underestimating σ^2 can also be seen in EMVS (Ročková and George, 2014), which can be viewed as iterative Bayesian ridge regression with an adaptive penalty term for each regression coefficient β_j instead of the same τ^2 above. EMVS also uses a conjugate prior formulation in which $\boldsymbol{\beta}$ depends on σ^2 *a priori* similarly to (2.10). As in the above ridge regression example, with this prior EMVS yields good estimates for $\boldsymbol{\beta}$, but severely underestimates σ^2. This occurs in the Section 4 example of Ročková and George (2014) with $n = 100$ and $p = 1000$. There, conditionally on the modal estimate of β, the associated modal estimate of σ^2 is 0.0014, a severe underestimate of the true variance $\sigma^2 = 3$. Fortunately, EMVS can be easily modified to use the independent prior specification, as now has been done in the publicly available `EMVS` R package (Ročková and Moran, 2018). It is interesting to note that the SSVS procedure of George and McCulloch (1993) used the nonconjugate independence prior formulation in lieu of the conjugate prior formulation for the continuous spike-and-slab setup.

A natural question to ask is: how does the poor estimate of the variance in the conjugate case affect the estimated regression coefficients? Insight is obtained by comparing (2.15) to the conditional posterior mean of $\boldsymbol{\beta}$ in the independent case, given by:

$$\mathbb{E}[\boldsymbol{\beta}|\sigma^2, \mathbf{Y}] = \left[\mathbf{X}^T\mathbf{X} + \frac{\sigma^2}{\tau^2}\mathbf{I}\right]^{-1}\mathbf{X}^T\mathbf{Y}. \tag{2.21}$$

In (2.15), the Gaussian prior structure allows for σ^2 to be factorized out so that the estimate of $\boldsymbol{\beta}$ does not depend on the variance. This lack of dependence on the variance is troubling, however, as we want to select fewer variables when the error variance is large making the signal-to-noise ratio low. This is in contrast to (2.21) where when σ^2 is large relative to τ^2, the signal-to-noise ratio is low and so the posterior estimate for $\boldsymbol{\beta}$ will be close to zero, correctly reflecting the relative lack of information. This does not occur for the posterior mean of $\boldsymbol{\beta}$ in the conjugate case.

Although the posterior mean of $\boldsymbol{\beta}$ in the conjugate prior formulation does not depend on σ^2, the posterior variance of $\boldsymbol{\beta}$ does depend on the error variance. Specifically, the posterior variance of $\boldsymbol{\beta}$ is given by:

$$\mathbb{E}[\boldsymbol{\beta}|\mathbf{Y}, \sigma^2] = \sigma^2[\mathbf{X}^T\mathbf{X} + \tau^{-2}\mathbf{I}]^{-1}. \tag{2.22}$$

Consequently, underestimation of σ^2 will result in too narrow credible intervals for $\boldsymbol{\beta}$. Further, underestimation of the error variance σ^2 will also result in too narrow prediction intervals for future responses.

2.3.3. What About a Prior Degrees of Freedom Adjustment?

At this point, one may wonder: if the problem seems to be the extra σ^p in the denominator, why not use the prior $\pi(\sigma^2) \propto \sigma^{p-4}$ instead of the right-Haar prior $\pi(\sigma^2) \propto \sigma^{-2}$ that is commonly used? This "p-sigma" prior then results in the joint prior:

$$\pi(\boldsymbol{\beta}|\sigma^2)\pi(\sigma^2) \propto \frac{1}{(\sigma^2)^2} \exp\left\{-\frac{1}{2\sigma^2\tau^2}\|\boldsymbol{\beta}\|^2\right\}, \tag{2.23}$$

which yields the implicit conditional prior on σ^2:

$$\sigma^2|\boldsymbol{\beta} \sim IG\left(1, \frac{\|\boldsymbol{\beta}\|^2}{2\tau^2}\right). \tag{2.24}$$

For the simulation setup in Section 2.3.2, this alternative conjugate prior would in fact remedy the variance estimates of the conjugate formulation (2.10). However, the p-sigma prior can actually lead to *overestimation* of the error variance, as opposed to the underestimation observed in Section 2.3.1. Heuristically, the mean of the prior (2.24) is now of order $O(q)$, where q is the number of non-zero $\boldsymbol{\beta}$. As many posterior concentration results require $q \to \infty$, albeit at a much slower rate than p (see, for example, van der Pas et al., 2016), this is particularly troublesome.

This overestimation can be further seen from the concentration of the prior captured in Proposition 2 below. As we will discuss in Section 2.4, a similar phenomenon also occurs for a penalized likelihood procedure that implicitly uses a p-sigma prior.

Proposition 2. *Suppose* $\|\boldsymbol{\beta}\|_0 = q$ *and* $\min_{j,\beta_j \neq 0} \beta_j^2 = K$ *for some constant* $K \in \mathbb{R}$. *Denote the true variance as* σ_0^2. *Then*

$$P(\sigma^2 \geq \delta\sigma_0^2 \mid \boldsymbol{\beta}) \geq 1 - \exp\left(-\frac{qK}{2\delta\sigma_0^2\tau^2}\right). \tag{2.25}$$

Proof. We have:

$$\begin{aligned} P(\sigma^2 \geq \delta\sigma_0^2 \mid \boldsymbol{\beta}) &= \int_{\delta\sigma_0^2}^{\infty} \frac{\|\boldsymbol{\beta}\|^2}{2\tau^2}\frac{1}{u^2}\exp\left(-\frac{\|\boldsymbol{\beta}\|^2}{2\tau^2}\frac{1}{u}\right)du \\ &\geq 1 - \exp\left(-\frac{qK}{2\delta\sigma_0^2\tau^2}\right). \square \end{aligned}$$

Proposition 2 implies that we can choose arbitrary $\delta > 1$ such that as $q \to \infty$, the p-sigma prior places increasing mass on values of σ^2 greater than $\delta\sigma_0^2$. Another concern regarding the p-sigma prior is more philosophical. As p gets larger, the p-sigma prior puts increasing mass on larger and larger values of σ^2, which does not seem justifiable.

For these reasons, we prefer the independent prior forms for the regression coefficients and error variance. We are also of the opinion that the simplicity of the independent prior is in its favor.

2.4. Connections with Penalized Likelihood Methods

Here we pause briefly to examine connections between Bayesian methods and developments in estimating the error variance in the penalized regression literature. Such connections can be drawn as penalized likelihood methods are implicitly Bayesian; the penalty functions can be interpreted as priors on the regression coefficients so these procedures also in effect yield MAP estimates.

One of the first papers to consider the unknown error variance case for the Lasso was Städler et al. (2010), who suggested the following penalized loss function for introducing unknown variance into the frequentist Lasso framework:

$$L_{pen}(\boldsymbol{\beta}, \sigma^2) = \frac{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2}{2\sigma^2} + \frac{\lambda}{\sigma}\|\boldsymbol{\beta}\|_1 + n \log \sigma. \tag{2.26}$$

Optimizing this objective function is in fact equivalent to MAP estimation for the following Bayesian model with the p-sigma prior discussed in Section 2.3.2:

$$\begin{aligned} \mathbf{Y} &\sim N(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}) \\ \pi(\boldsymbol{\beta}|\sigma^2) &\propto \frac{1}{\sigma^p}\prod_{j=1}^{p} e^{-\lambda|\beta_j|/\sigma} \\ \pi(\sigma^2) &\propto \sigma^p. \end{aligned} \tag{2.27}$$

Interestingly, Sun and Zhang (2010) proved that the resulting estimator for the error variance *overestimates* the noise level unless $\lambda\|\boldsymbol{\beta}^*\|_1/\sigma^* = o(1)$, where $\boldsymbol{\beta}^*$ and σ^* are the true values of the regression coefficients and error variance, respectively. However, this condition requires q, the true number of non-zero $\boldsymbol{\beta}$, to be of the following order (details in Section 2.9.2 of the Appendix).

$$q = o\left(\sqrt{n/\log p}\right). \tag{2.28}$$

That is, the true dimension q cannot at the same time increase at the required rate for

posterior contraction *and* result in consistent estimates for the error variance. Note also the connection to Proposition 2: there, the prior mass on σ^2 will concentrate on values greater than the true variance unless $\|\boldsymbol{\beta}\|^2/\tau^2 = o(1)$.

To resolve this issue of overestimating the error variance, Sun and Zhang (2012) proposed as an alternative the "scaled Lasso", an algorithm which minimizes the following penalized joint loss function via coordinate descent:

$$L_\lambda(\boldsymbol{\beta}, \sigma) = \frac{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2}{2\sigma} + \frac{n\sigma}{2} + \lambda \sum_{j=1}^{p} |\beta_j|. \tag{2.29}$$

This loss function is a penalized version of Huber's concomitant loss function, and so may be viewed as performing robust high-dimensional regression. It is also equivalent to the "square-root Lasso" of Belloni et al. (2014). Minimization of the loss function (2.29) can be viewed as MAP estimation for the Bayesian model (with a slight modification):

$$\begin{aligned} \mathbf{Y} &\sim N(\mathbf{X}\boldsymbol{\beta}, \sigma\mathbf{I}) \\ \pi(\boldsymbol{\beta}) &\propto \prod_{j=1}^{p} \frac{\lambda}{2} e^{-\lambda|\beta_j|} \\ \sigma &\sim \text{Gamma}(n+1, n/2). \end{aligned} \tag{2.30}$$

Note that to interpret the scaled Lasso as a Bayesian procedure, σ, rather than σ^2, plays the role of the variance in (2.30). Sun and Zhang (2012) essentially then re-interpret σ as the standard deviation again after optimization of (2.29). This re-interpretation can be thought of as an "unbiasing" step for the error variance. It is a little worrisome, however, that the implicit prior on the error variance is very informative: as $n \to \infty$, this Gamma prior concentrates around $\sigma = 2$.

Sun and Zhang (2012) proved that the scaled Lasso estimate $\widehat{\sigma}(\mathbf{X}, \mathbf{Y})$ is consistent for the "oracle" estimator

$$\sigma^* = \frac{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}^*\|}{\sqrt{n}}, \tag{2.31}$$

where $\boldsymbol{\beta}^*$ are the true regression coefficients, for the value of $\lambda_0 \propto \sqrt{(2/n)\log p}$. This estimator (2.31) is called the oracle because it treats the true regression coefficients as if they were known. The term $\|\mathbf{Y}-\mathbf{X}\boldsymbol{\beta}^*\|^2$ is then simply the sum of normal random variables, of which we calculate the variance as $\sum_{i=1}^n \varepsilon_i^2/n$.

2.5. Global-Local Shrinkage

In this section, we examine how the use of a conjugate prior affects the machinery of the Gaussian global-local shrinkage paradigm. The general structure for this class of priors is given by:

$$\begin{aligned} &\beta_j \sim N(0, \tau^2\lambda_j^2), \quad \lambda_j^2 \sim \pi(\lambda_j^2), \quad j = 1, \ldots, p \\ &\tau^2 \sim \pi(\tau^2) \end{aligned} \tag{2.32}$$

where τ^2 is the "global" variance and λ_j^2 is the "local" variance. Note that taking τ^2 to be the same as the error variance σ^2 would result in a conjugate prior in this setting. This is exactly what was done in the original formulation of the Bayesian lasso by Park and Casella (2008), which can be recast in the Gaussian global-local shrinkage framework as follows (notation changed slightly for consistency):

$$\begin{aligned} &\mathbf{Y}|\boldsymbol{\beta}, \sigma^2 \sim N_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}_n) \\ &\beta_j|\sigma^2, \lambda_j^2 \sim N(0, \sigma^2\lambda_j^2), \quad \pi(\lambda_j^2) = \frac{u^2}{2}e^{-u^2\lambda_j^2/2}, \quad j = 1, \ldots, p \\ &\pi(\sigma^2) \propto \sigma^{-2}. \end{aligned} \tag{2.33}$$

In the conjugate formulation (2.33), σ^2 plays the dual role of representing the error variance as well as acting as the global shrinkage parameter. This is problematic in light of the mechanics of global-local shrinkage priors. Specifically, Polson and Scott (2010) recommend the following requirements for the global and local variances in (2.32): $\pi(\tau^2)$ should have substantial mass near zero to shrink all the regression coefficients so that the vast majority

are negligible; and $\pi(\lambda_j^2)$ should have heavy tails so that it can be quite large, allowing for a few large coefficients to "escape" the heavy shrinkage of the global variance.

This heuristic is formalized in much of the shrinkage estimation theory. For the normal means problem where $\mathbf{X} = \mathbf{I}_n$ and $\boldsymbol{\beta} \in \mathbb{R}^n$, van der Pas et al. (2016) prove that the following conditions result in the posterior recovering nonzero means with the optimal rate: (i) $\pi(\lambda_j^2)$ should be a uniformly regular varying function which does not depend on n; and (ii) $\tau^2 = \frac{q}{n}\log(n/q)$, where q is number of non-zero β_j.

The uniformly regular varying property in (i) intuitively preserves the "flatness" of the prior even under transformations of the parameters, unlike traditional "non-informative" priors (Bhadra et al., 2016). In preserving these heavy tails, such priors for λ_j^2 allow for a few large coefficients to be estimated. The condition (ii) encourages τ^2 to tend to zero which would be a concerning property if it were also the error variance. These results suggest we cannot identify the error variance with the global variance parameter on the regression coefficients as in (2.33): it cannot simultaneously both shrink all the regression coefficients and be a good estimate of the residual variance. Finally, we note that Hans (2009) also considered the independent case for the Bayesian lasso in which the error variance is not identified with the global variance.

An alternative conjugate formulation for Gaussian global-local shrinkage priors is to instead include three variance terms in the prior for β_j: the error variance, σ^2, the global variance, τ^2, and the local variance, λ_j^2. For example, Carvalho et al. (2010) give the conjugate form of the horseshoe prior:

$$\begin{aligned} \beta_j|\sigma^2,\tau^2,\lambda_j^2 \sim N(0,\sigma^2\tau^2\lambda_j^2), \quad \lambda_j^2 \sim \pi(\lambda_j^2), \quad j=1,\ldots,p \\ \tau^2 \sim \pi(\tau^2), \\ \pi(\sigma^2) \propto \sigma^{-2}. \end{aligned} \tag{2.34}$$

This prior formulation (2.34) remedies the aforementioned issue in the Bayesian lasso as it

separates the roles of the error variance and global variance. However, this prior structure can still be problematic for error variance estimation.

Consider the conditional posterior mean of σ^2 for the model (2.34):

$$\mathbb{E}[\sigma^2|\mathbf{Y}, \boldsymbol{\beta}, \tau^2, \lambda_j^2] = \frac{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^p \beta_j^2/\lambda_j^2\tau^2}{n+p-2}. \tag{2.35}$$

Proposition 3 highlights that, given the true regression coefficients, the conditional posterior mean of σ^2 underestimates the oracle variance (2.31) when $\boldsymbol{\beta}$ is sparse.

Proposition 3. *Consider the global-local prior formulation given in* (2.34)*. Denote the true vector of regression coefficients by* $\boldsymbol{\beta}^*$ *where* $\|\boldsymbol{\beta}^*\|_0 = q$*. Suppose* $\max_j \beta_j^{*2} = M_1$ *for some constant* $M_1 \in \mathbb{R}$*. Denote the oracle estimator for* σ *given in* (2.31) *by* σ^* *and suppose* $\sigma^* = O(1)$*. Suppose also that for* $j \in \{1, \ldots, p\}$ *with* $\beta_j \neq 0$*, we have* $\tau^2\lambda_j^2 > M_2$ *for some* $M_2 \in \mathbb{R}$*. Then*

$$\mathbb{E}[\sigma^2|\mathbf{Y}, \boldsymbol{\beta}^*, \tau^2, \lambda_j^2] \leq \frac{n\sigma^{*2}}{n+p-2} + \frac{q}{n+p-2}\frac{M_1}{M_2}. \tag{2.36}$$

In particular, as $p/n \to \infty$ *and* $q/p \to 0$*, we have*

$$\mathbb{E}[\sigma^2|\mathbf{Y}, \boldsymbol{\beta}^*, \tau^2, \lambda_j^2] = o(1). \tag{2.37}$$

Given the mechanics of global-local shrinkage priors, the assumption in Proposition 3 that the term $\tau^2\lambda_j^2$ is bounded from below for non-zero β_j is not unreasonable. This is because for large β_j, the local variance λ_j^2 must be large enough to counter the extreme shrinkage effect of τ^2. Indeed, the prior for λ_j^2 must have "heavy enough" tails to enable this phenomenon.

We should note that Proposition 3 illustrates the poor performance of the posterior mean (2.35) given the true regression coefficients $\boldsymbol{\beta}^*$, whereas the horseshoe procedure does not actually threshold the negligible β_j to zero in the posterior mean of $\boldsymbol{\beta}$. For these small β_j, the term $\tau^2\lambda_j^2$ may be very small and potentially counteract the underestimation phenomenon.

However, it is still troubling to use an estimator for the error variance that does not behave as the oracle estimator when the true regression coefficients are known. This is in contrast to the independent prior formulation where the conditional posterior mean of σ^2 is simply:

$$\mathbb{E}[\sigma^2|\mathbf{Y}, \boldsymbol{\beta}] = \frac{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2}{n-2}. \tag{2.38}$$

Note also that the problem of underestimation of σ^2 is exacerbated for modal estimation under the prior (2.34). This is because modal estimators often threshold small coefficients to zero and so the term $\sum_{j=1}^{p} \beta_j^2/\lambda_j^2\tau^2$ becomes negligible as in Proposition 3. As MAP estimation using global-local shrinkage priors is becoming more common (see, for example, Bhadra et al., 2017), we caution against the use of these conjugate prior forms.

A different argument for using conjugate priors with the horseshoe is given by Piironen and Vehtari (2017). They advocate for the model (2.34), arguing that it leads to a prior on the effective number of non-zero coefficients which does not depend on σ^2 and n. However, this quantity is derived from the posterior of $\boldsymbol{\beta}$ and so does not take into account the uncertainty inherent in the variable selection process. As a thought experiment: suppose that we know the error variance, σ^2, and number of observations, n. If the error variance is too large and the number of observations are too few, we would not expect to be able to say much about $\boldsymbol{\beta}$ at all, and this intuition should be reflected in the effective number of non-zero coefficients. This point is similar to our discussion at the end of Section 2.3.2 regarding estimation of $\boldsymbol{\beta}$.

As before, we recommend independent priors on both the error variance and regression coefficients to both prevent distortion of the global-local shrinkage mechanism and to obtain better estimates of the error variance.

2.6. Spike-and-Slab Lasso with Unknown Variance

2.6.1. Spike-and-Slab Lasso

We now turn to the Spike-and-Slab Lasso (SSL, Ročková and George, 2018) and consider how to incorporate the unknown variance case. The SSL places a mixture prior on the regression coefficients $\boldsymbol{\beta}$, where each β_j is assumed *a priori* to be drawn from either a Laplacian "spike" concentrated around zero (and hence be considered negligible), or a diffuse Laplacian "slab" (and hence may be large). Thus the hierarchical prior over $\boldsymbol{\beta}$ and the latent indicator variables $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_p)$ is given by

$$\begin{aligned} \pi(\boldsymbol{\beta}|\boldsymbol{\gamma}) &\sim \prod_{j=1}^{p} \left[\gamma_j \psi_1(\beta_j) + (1-\gamma_j)\psi_0(\beta_j)\right], \\ \pi(\boldsymbol{\gamma}|\theta) &= \prod_{j=1}^{p} \theta^{\gamma_j}(1-\theta)^{1-\gamma_j} \quad \text{and} \quad \theta \sim \text{Beta}(a,b), \end{aligned} \tag{2.39}$$

where $\psi_1(\beta) = \frac{\lambda_1}{2} e^{-|\beta|\lambda_1}$ is the slab distribution and $\psi_0(\beta) = \frac{\lambda_0}{2} e^{-|\beta|\lambda_0}$ is the spike ($\lambda_1 \ll \lambda_0$), and we have used the common exchangeable beta-binomial prior for the latent indicators.

Ročková and George (2018) recast this hierarchical model into a penalized likelihood framework, allowing for the use of existing efficient algorithms for modal estimation while retaining the adaptivity inherent in the Bayesian formulation. The regression coefficients $\boldsymbol{\beta}$ are then estimated by

$$\widehat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^p}{\arg\max} \left\{ -\frac{1}{2}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + pen(\boldsymbol{\beta}) \right\} \tag{2.40}$$

where

$$pen(\boldsymbol{\beta}) = \log\left[\frac{\pi(\boldsymbol{\beta})}{\pi(\mathbf{0}_p)}\right], \quad \pi(\boldsymbol{\beta}) = \int_0^1 \prod_{j=1}^{p} [\theta\psi_1(\beta_j) + (1-\theta)\psi_0(\beta_j)] d\pi(\theta). \tag{2.41}$$

Ročková and George (2018) note a number of advantages in using a mixture of Laplace densities in (2.39), instead of the usual mixture of Gaussians as has been standard in the Bayesian variable selection literature. First, the Laplacian spike serves to automatically threshold modal estimates of β_j to zero when β_j is small, much like the Lasso. However, unlike the Lasso, the slab distribution in the prior serves to stabilize the larger coefficients so they are not downward biased. Additionally, the heavier Laplacian tails of the slab distribution yields optimal posterior concentration rates (Ročková, 2018).

Although the use of the spike-and-slab prior is typically associated with "two-group" Bayesian variable selection methods, the Spike-and-Slab Lasso can also be interpreted as a "one-group" global-local shrinkage method as the spike density is continuous. As such, the use of a conjugate prior for the error variance here will result in underestimation, similarly to the results for global-local shrinkage priors in Section 2.5. This is especially the case as the SSL procedure finds the modes of the posterior, automatically thresholding negligible regression coefficients to zero. In the next section, we provide further details on why this underestimation phenomenon occurs for the SSL with a conjugate prior formulation. Afterwards, we introduce the SSL with unknown variance which avoids this underestimation problem by instead utilizing an independent prior framework.

2.6.2. The Failure of a Conjugate Prior

This conjugate prior formulation for the Spike-and-Slab Lasso is given by:

$$\pi(\boldsymbol{\beta}|\boldsymbol{\gamma}, \sigma^2) \sim \prod_{j=1}^{p} \left(\gamma_j \frac{\lambda_1}{2\sigma} e^{-|\beta_j|\lambda_1/\sigma} + (1-\gamma_j) \frac{\lambda_0}{2\sigma} e^{-|\beta_j|\lambda_0/\sigma} \right) \tag{2.42}$$

$$\boldsymbol{\gamma}|\theta \sim \prod_{j=1}^{p} \theta^{\gamma_j}(1-\theta)^{1-\gamma_j}, \quad \theta \sim \text{Beta}(a, b) \tag{2.43}$$

$$p(\sigma^2) \propto \sigma^{-2}. \tag{2.44}$$

We find the posterior modes of our parameters using the EM algorithm, the details of which

can be found in Section 2.9.3 of the Appendix. At the $(k+1)$th iteration, the EM update for the error variance is:

$$\sigma^{(k+1)} = \frac{Q + \sqrt{Q^2 + 4(\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}^{(k)}\|^2)(n+p+2)}}{2(n+p+2)} \tag{2.45}$$

with

$$Q = \sum_{i=1}^{p} |\beta_j^{(k)}| \lambda^*(\beta_j^{(k)}/\sigma^{(k)}; \theta^{(k)}), \tag{2.46}$$

$$\lambda^*(\beta;\theta) = \lambda_1 p^*(\beta;\theta) + \lambda_0(1 - p^*(\beta;\theta)), \tag{2.47}$$

$$p^*(\beta;\theta) = \left[1 + \frac{\lambda_0}{\lambda_1}\left(\frac{1-\theta}{\theta}\right)\exp\{-|\beta|(\lambda_0 - \lambda_1)\}\right]^{-1}, \tag{2.48}$$

where $\boldsymbol{\beta}^{(k)}, \sigma^{(k)}, \theta^{(k)}$ are the parameter values after the kth iteration.

Let us take a closer look at the update (2.45). Following the line of reasoning in Sun and Zhang (2010), an expert with oracle knowledge of the true regression coefficients $\boldsymbol{\beta}^*$ would estimate the noise level by the oracle estimator:

$$\sigma^{*2} = \frac{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}^*\|}{n}. \tag{2.49}$$

However, the maximum *a posteriori* estimate of σ at the true values of $\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*$ is given by

$$\widehat{\sigma}_{MAP} = \tau + \sqrt{\tau^2 + \frac{(\sigma^*)^2}{1 + p/n + 2/n}} \tag{2.50}$$

where $\tau = \lambda_1 \|\boldsymbol{\beta}^*\|_1/[2(n+p+2)]$. Here we see that if $n \to \infty$ with p fixed, we have $\widehat{\sigma}_{MAP} \to \sigma^*$. If, however, we have $p/n \to \infty$ and $q/p \to 0$, where the underlying sparsity is $q = \|\boldsymbol{\beta}^*\|_0$, we have $\widehat{\sigma}_{MAP} \to 0$. Thus, similarly to our previous examples in Sections 2.3 and 2.5, we will severely underestimate the error variance. As in these examples, the remedy is to use the independent prior on σ^2 and $\boldsymbol{\beta}$.

2.6.3. Spike-and-Slab Lasso with Unknown Variance

We now introduce the Spike-and-Slab Lasso with unknown variance, which considers the regression coefficients and error variance to be *a priori* independent. The hierarchical model is

$$\pi(\boldsymbol{\beta}|\boldsymbol{\gamma}) \sim \prod_{j=1}^{p} [\gamma_j \psi_1(\beta_j) + (1-\gamma_j)\psi_0(\beta_j)] \tag{2.51}$$

$$\boldsymbol{\gamma}|\theta \sim \prod_{j=1}^{p} \theta^{\gamma_j}(1-\theta)^{1-\gamma_j}, \quad \theta \sim \text{Beta}(a,b) \tag{2.52}$$

$$\pi(\sigma^2) \propto \sigma^{-2}. \tag{2.53}$$

The log posterior, up to an additive constant, is given by

$$L(\boldsymbol{\beta}, \sigma^2) = -\frac{1}{2\sigma^2}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 - (n+2)\log\sigma + \sum_{j=1}^{p} pen(\beta_j|\theta_j) \tag{2.54}$$

where, for $j = 1, \ldots, p$,

$$pen(\beta_j|\theta_j) = -\lambda_1|\beta_j| + \log[p^*(0;\theta_j)/p^*(\beta_j;\theta_j)], \tag{2.55}$$

$$\text{with} \quad p^*(\beta;\theta) = \frac{\theta\psi_1(\beta)}{\theta\psi_1(\beta) + (1-\theta)\psi_0(\beta)} \quad \text{and} \quad \theta_j = \mathbb{E}[\theta|\boldsymbol{\beta}_{\setminus j}]. \tag{2.56}$$

For large p, Ročková and George (2018) note that the conditional expectation $\mathbb{E}[\theta|\boldsymbol{\beta}_{\setminus j}]$ is very similar to $\mathbb{E}[\theta|\boldsymbol{\beta}]$ and so for practical purposes we treat them as equal and denote $\theta_\beta = \mathbb{E}[\theta|\boldsymbol{\beta}]$.

To find the modes of (2.54), we pursue a similar coordinate ascent strategy to Ročková and George (2018), cycling through updates for each β_j and σ^2 while updating the conditional expectation θ_β. This conditional expectation does not have an analytical expression;

however, Ročková and George (2018) note that it can be approximated by

$$\theta_\beta \approx \frac{a + \|\boldsymbol{\beta}\|_0}{a + b + p}. \tag{2.57}$$

We now outline the estimation strategy for $\boldsymbol{\beta}$. As noted in Lemma 3.1 of Ročková and George (2018), there is a simple expression for the derivative of the SSL penalty:

$$\frac{\partial pen(\beta_j|\theta_\beta)}{\partial|\beta_j|} \equiv -\lambda^*(\beta_j;\theta_\beta) \tag{2.58}$$

where

$$\lambda^*(\beta_j;\theta_\beta) = \lambda_1 p^*(\beta_j;\theta_\beta) + \lambda_0[1 - p^*(\beta_j;\theta_\beta)]. \tag{2.59}$$

Using the above expression, the Karush-Kuhn-Tucker (KKT) conditions yield the following necessary condition for the global mode $\widehat{\boldsymbol{\beta}}$:

$$\widehat{\beta}_j = \frac{1}{n}\left[|z_j| - \sigma^2\lambda^*(\widehat{\beta}_j;\theta_\beta)\right]_+ sign(z_j), \quad j = 1,\ldots,p \tag{2.60}$$

where $z_j = \mathbf{X}_j^T(\mathbf{Y} - \sum_{k\neq j}^p \widehat{\beta}_k \cdot \mathbf{X}_k)$ and we assume that the design matrix $\mathbf{X}$ has been centered and standardized to have norm $\sqrt{n}$. The condition (2.60) is very close to the familiar soft-thresholding operator for the Lasso, except that the penalty term $\lambda^*(\beta_j;\theta)$ differs for each coordinate. Similarly to other non-convex methods, this enables *selective shrinkage* of the coefficients, mitigating the bias issues associated with the Lasso. As a non-convex method, however, (2.60) is not a sufficient condition for the global mode. This is particularly problematic when the posterior landscape is highly multimodal, a consequence of $p \gg n$ and large λ_0. To eliminate many of these suboptimal local modes from consideration, Ročková and George (2018) develop a more refined characterization of the global mode. This characterization follows the arguments of Zhang and Zhang (2012) and can easily be modified for the unknown variance case of the SSL, detailed in Proposition 4.

Proposition 4. *The global mode* $\widehat{\boldsymbol{\beta}}$ *satisfies*

$$\widehat{\beta}_j = \begin{cases} 0 & \text{when } |z_j| \leq \Delta \\ \frac{1}{n}[|z_j| - \sigma^2\lambda^*(\widehat{\beta}_j;\theta_\beta)]_+ sign(z_j) & \text{when } |z_j| > \Delta \end{cases} \tag{2.61}$$

where

$$\Delta \equiv \inf_{t>0}[nt/2 - \sigma^2 pen(t|\theta_\beta)/t]. \tag{2.62}$$

Unfortunately, computing (2.62) can be difficult. Instead, we seek an approximation to the threshold Δ. A useful upper bound is $\Delta \leq \sigma^2\lambda^*(0;\theta_\beta)$ (Zhang and Zhang, 2012). However, when λ_0 gets large, this bound is too loose and can be improved. The improved bounds are given in Proposition 5, the analogue of Proposition 3.2 of Ročková and George (2018) for the unknown variance case. Before stating the result, the following function is useful to simplify exposition:

$$g(x;\theta) = [\lambda^*(x;\theta) - \lambda_1]^2 + \frac{2n}{\sigma^2}\log[p^*(x;\theta)]. \tag{2.63}$$

Proposition 5. *When* $\sigma(\lambda_0 - \lambda_1) > 2\sqrt{n}$ *and* $g(0;\theta_\beta) > 0$ *the threshold* Δ *is bounded by*

$$\Delta^L < \Delta < \Delta^U,$$

where

$$\Delta^L = \sqrt{2n\sigma^2\log[1/p^*(0;\theta_\beta)] - \sigma^4 d_j} + \sigma^2\lambda_1, \tag{2.64}$$

$$\Delta^U = \sqrt{2n\sigma^2\log[1/p^*(0;\theta_\beta)]} + \sigma^2\lambda_1 \tag{2.65}$$

and

$$0 < d_j < \frac{2n}{\sigma^2} - \left(\frac{n}{\sigma^2(\lambda_0 - \lambda_1)} - \frac{\sqrt{2n}}{\sigma}\right)^2.$$

Thus, when λ_0 is large and consequently $d_j \to 0$, the lower bound on the threshold ap-

proaches the upper bound, yielding the approximation $\Delta \approx \Delta^U$. We additionally note the central role that the error variance plays in the thresholds in Proposition 5. As σ^2 increases, the thresholds also increase, making it more difficult for regression coefficients to be selected. This is exactly what we want when the signal to noise ratio is small.

Bringing this all together, we incorporate this refined characterization of the global mode into the update for the coefficients via the generalized thresholding operator of Mazumder et al. (2011):

$$\widetilde{S}(z, \lambda, \Delta) = \frac{1}{n}(|z| - \lambda)_+ sign(z)\mathbb{I}(|z| > \Delta). \tag{2.66}$$

The coordinate-wise update is then

$$\widehat{\beta}_j \leftarrow \widetilde{S}(z_j, \widehat{\sigma}^2\lambda^*(\widehat{\beta}_j; \widehat{\theta}_\beta), \Delta) \tag{2.67}$$

where

$$\Delta = \begin{cases} \sqrt{2n\widehat{\sigma}^2 \log[1/p^*(0; \widehat{\theta}_\beta)]} + \widehat{\sigma}^2\lambda_1 & \text{if } g(0; \widehat{\theta}_\beta) > 0, \\ \widehat{\sigma}^2\lambda^*(0; \widehat{\theta}_\beta) & \text{otherwise.} \end{cases} \tag{2.68}$$

The conditional expectation θ_β is updated according to (2.57).

Finally, given the most recent update of the coefficient vector $\widehat{\boldsymbol{\beta}}$, the update for the error variance σ^2 is simply:

$$\widehat{\sigma}^2 \leftarrow \frac{\|\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2}{n + 2}. \tag{2.69}$$

Note that this update for σ^2 is a *conditional* mode, not a marginal mode, and so it does not underestimate the error variance in the same way as (2.16). Indeed, conditional on the true regression coefficients, (2.69) is essentially the oracle estimator (2.31). However, although we retain the update (2.69) *during* optimization in order to iterate between the modes of

$\boldsymbol{\beta}$ and σ^2, after the algorithm has converged, our final estimator of σ^2 is obtained as

$$\widehat{\sigma}^2_{adj} = \frac{\|\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2}{n - \widehat{q}}, \tag{2.70}$$

where $\widehat{q} = \|\widehat{\boldsymbol{\beta}}\|_0$. Note that (2.70) incorporates an appropriate degrees of freedom adjustment to account for the fact that $\widehat{\boldsymbol{\beta}}$ is an estimate of the unknown true $\boldsymbol{\beta}$.

In principle, both σ^2 and the conditional expectation θ_β should be updated after each β_j, $j = 1, \ldots, p$. In practice, however, there will be little change after one coordinate update and so both σ^2 and θ_β can be updated after M coordinates are updated, where M is the update frequency. The default implementation updates σ^2 and θ_β after every $M = 10$ coordinate updates.

2.6.4. Implementation

In the SSL with fixed variance, Ročková and George (2018) propose a "dynamic posterior exploration" strategy whereby the slab parameter λ_1 is held fixed and the spike parameter λ_0 is gradually increased to approximate the ideal point mass prior. Holding the slab parameter fixed serves to stabilize the non-zero coefficients, unlike the Lasso which applies an equal level of shrinkage to all regression coefficients. Meanwhile, gradually increasing λ_0 over a "ladder" of values serves to progressively threshold negligible coefficients. More practically, the dynamic strategy aids in mode detection: when $(\lambda_1 - \lambda_0)^2 \leq 4$, the objective is convex (Ročková and George, 2018). In fact, when $\lambda_0 = \lambda_1$, it is equivalent to the Lasso. As λ_0 is increased, the posterior landscape becomes multimodal, but using the solution from the previous value of λ_0 as a "warm start" allows the procedure to more easily find modes. Thus, progressively increasing λ_0 acts as an annealing strategy.

When σ^2 is treated as unknown, the successive warm start strategy of Ročková and George (2018) will require additional intervention. This is because the objective (2.54) is *always* non-convex when σ^2 is unknown, unlike the fixed case where it is convex when $(\lambda_1 - \lambda_0)^2 \leq 4$. In particular, for small $\lambda_0 \approx \lambda_1$ there may be many negligible but non-zero β_j included in

the model. When $p > n$, this severe overfitting can result in all the variation in $\mathbf{Y}$ being explained by the model, forcing the estimate of the error variance, $\widehat{\sigma}^2$ to a mode at zero. If this suboptimal solution is propagated for larger values of λ_0, the optimization routine will remain "stuck" in that part of the posterior landscape. As an implementation strategy to avoid this absorbing state, we keep the estimate of σ^2 fixed at an initial value until λ_0 reaches a value at which the algorithm converges in less than 100 iterations. We then reinitialize $\boldsymbol{\beta}$ and σ^2 and begin to simultaneously update σ^2 for the next largest λ_0 value in the ladder. The intuition behind this strategy is that we first find a solution to a convex problem (with σ^2 fixed) and then use this solution as a warm start for the non-convex problem (with σ^2 unknown). A related two-step strategy for non-convex optimization has also been proven successful for robust M-estimation (Loh, 2017).

For initialization, we follow Ročková and George (2018) and initialize the regression coefficients, $\boldsymbol{\beta}$, at zero and $\theta_0 = 0.5$. For the error variance, we devised an initialization strategy that is motivated by the prior for σ^2 used in Chipman et al. (2010). Those authors used a scaled-inverse-χ^2 prior for the error variance with degrees of freedom $\nu = 3$ and scale parameter chosen such that the sample variance of $\mathbf{Y}$ corresponds to the $90th$ quantile of the prior. This is a natural choice as the variance of $\mathbf{Y}$ is the maximum possible value for the error variance. We set the initial value of σ^2 to be the mode of this scaled-inverse-χ^2 distribution, a strategy which we have found to be effective in practice.

The entire implementation strategy is summarized in Algorithm 1.

2.6.5. Scaled Spike-and-Slab Lasso

An alternative approach for extending the SSL for unknown variance is to follow the scaled Lasso framework of Sun and Zhang (2012). In their original scaled Lasso paper, Sun and Zhang (2012) note that their loss function can be used with many penalized likelihood procedures, including the MCP and the SCAD penalties. Here, we develop the *scaled Spike-and-Slab Lasso*. The loss function for the scaled SSL is the same as that of the scaled

Lasso but with a different penalty:

$$L(\boldsymbol{\beta}, \sigma^2) = -\frac{1}{2\sigma}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 - \frac{n\sigma}{2} + \sum_{j=1}^{p} pen(\beta_j|\theta_\beta) \tag{2.71}$$

where $pen(\beta_j|\theta_\beta)$ is as defined in (2.55) and again we use the approximation (2.57) for the conditional expectation θ_β. In using this loss function, we are of course departing from the Bayesian paradigm and simply considering this procedure as a penalized likelihood method with a spike-and-slab inspired penalty.

The algorithm to find the modes of (2.71) is very similar to Algorithm 1, the only difference being we replace all σ^2 terms in the updates (2.67) and (2.68) with σ. This is because the refined thresholds for the coefficients are derived using the KKT conditions where the only difference between the two procedures is σ vs. σ^2.

The update for σ^2 is only very slightly different from the SSL with unknown variance:

$$\widehat{\sigma}^2 \leftarrow \frac{\|\mathbf{Y} - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2}{n}. \tag{2.72}$$

How do we expect the scaled Spike-and-Slab Lasso to compare to the Spike-and-Slab Lasso with unknown variance? The threshold levels Δ for the scaled SSL will be smaller after replacing σ^2 with σ. This may potentially result in more false positives being included in the scaled SSL model. In terms of variance estimation, the updates for σ^2 are effectively the same; the only differences we should expect are those arising from a more saturated estimate for $\boldsymbol{\beta}$. These hypotheses are examined in the simulation study in the next session.

2.6.6. Simulation Study

We now compare the Spike-and-Slab Lasso with unknown variance with several penalized likelihood methods, including the original Spike-and-Slab Lasso with fixed variance of Ročková and George (2018) as well as the scaled Spike-and-Slab Lasso outlined in the previous section. We investigate both the efficacy of the SSL with unknown variance and

the benefits of simultaneously estimating the regression coefficients $\boldsymbol{\beta}$ and error variance σ^2 in variable selection. We do not consider the SSL with the p-sigma prior from Section 3.3 as the objective is similar to Städler et al. (2010) (albeit with an adaptive penalty) and so we would expect similar overestimation of σ^2 as proved by Sun and Zhang (2012). We consider three different simulation settings.

For the first simulation setting, we consider the same simulation setting of Ročková and George (2018) with $n = 100$ and $p = 1000$ but use an error variance of $\sigma^2 = 3$ instead of $\sigma^2 = 1$. The data matrix $\mathbf{X}$ is generated from a multivariate Gaussian distribution with mean $\mathbf{0}_p$ and a block-diagonal covariance matrix $\boldsymbol{\Sigma} = \text{bdiag}(\widetilde{\Sigma}, \ldots, \widetilde{\Sigma})$ where $\widetilde{\Sigma} = \{\widetilde{\sigma}\}_{i,j=1}^{50}$ with $\widetilde{\sigma}_{ij} = 0.9$ if $i \neq j$ and $\widetilde{\sigma}_{ii} = 1$. The true vector $\boldsymbol{\beta}_0$ is constructed by assigning regression coefficients $\{-2.5, -2, -1.5, 1.5, 2, 2.5\}$ to $q = 6$ entries located at $\{1, 51, 101, 151, 201, 251\}$ and setting to zero the remaining coefficients. Hence, there are 20 independent blocks of 50 highly correlated predictors where the first 6 blocks each contain only one active predictor. The response was generated as in (2.1) with error variance $\sigma^2 = 3$.

We compared the Spike-and-Slab Lasso with unknown variance to the fixed variance Spike-and-Slab Lasso with two settings: (i) $\sigma^2 = 1$, and (ii) $\sigma^2 = 3$, the true variance. The prior settings for θ were $a = 1, b = p$. The slab parameter was set to $\lambda_1 = 1$. For the spike parameter, we used a ladder $\lambda_0 \in I = \{1, 2, \ldots, 100\}$.

Additional methods compared were the scaled SSL from Section 2.6.5, the Lasso (Friedman et al., 2010), the scaled Lasso (Sun and Zhang, 2012), the Adaptive Lasso (Zou, 2006), SCAD (Fan and Li, 2001), and MCP (Zhang, 2010).

The analysis was repeated 100 times with new covariates and responses generated each time. For each, the metrics recorded were: the Hamming distance (HAM) between the support of the estimated $\boldsymbol{\beta}$ and the true $\boldsymbol{\beta}_0$; the prediction error (PE), defined as

$$\text{PE} = \|\mathbf{X}\boldsymbol{\beta}_0 - \mathbf{X}\widehat{\boldsymbol{\beta}}\|^2; \tag{2.73}$$

the number of false negatives (FN); the number of false positives (FP); the number of true positives (TP); Matthew's Correlation Coefficient (MCC), defined as:

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}; \tag{2.74}$$

the percentage of times the method found the correct model (COR); and the time in seconds (TIME). The average of these metrics for each method over the 100 repetitions are displayed in Table 1.

We can see that the Spike-and-Slab Lasso with the variance fixed and equal to the truth ($\sigma^2 = 3$) performs the best in terms of the Hamming distance, prediction error, and MCC. Encouragingly, the Spike-and-Slab Lasso with unknown variance performs almost as well as the "oracle" version where the true variance is known. The SSL with unknown variance in turn performs better than a naive implementation of the SSL with fixed variance ($\sigma^2 = 1$). We note that the prediction error for the latter implementation is higher than the Adaptive Lasso and SCAD; however, these frequentist methods use cross-validation to choose their regularization parameter and so are optimizing for prediction to the possible detriment of other metrics; the SSL ($\sigma^2 = 1$) still has fewer false positives and a higher MCC. However, both the SSL ($\sigma^2 = 3$) and unknown σ^2 have smaller prediction error than the rest of the methods, including those which use cross-validation, which highlights the predictive gains afforded by variance estimation.

Following from the discussion in Section 2.6.5, we can see that the scaled SSL indeed finds more false positives than the SSL with unknown variance. This is a result of the smaller thresholds in estimating the regression coefficients. We can see that the scaled Lasso significantly reduces the number of false positives found as compared to the Lasso; however, the issues with the Lasso penalty remain.

Figure 2 shows the variance estimates over the 100 repetitions for the SSL with unknown variance, the scaled SSL and the scaled Lasso. For the SSL with unknown σ^2, these are the

	HAM	PE	MCC	TP	FP	FN	COR	TIME
SSL (fixed $\sigma^2 = 3$)	**1.1 (0.1)**	**39.6 (3.7)**	**0.91 (0.01)**	**5.5 (0.1)**	**0.5 (0.1)**	**0.5 (0.1)**	**58**	**0.03 (0.00)**
SSL (unknown σ^2)	1.2 (0.2)	43.4 (3.9)	0.90 (0.01)	5.4 (0.1)	0.6 (0.1)	0.6 (0.1)	55	0.04 (0.00)
Scaled SSL	2.0 (0.2)	65.8 (5.0)	0.84 (0.01)	5.2 (0.1)	1.2 (0.1)	0.8 (0.1)	32	0.07 (0.00)
SSL (fixed $\sigma^2 = 1$)	4.5 (0.3)	114.9 (5.3)	0.69 (0.02)	4.8 (0.1)	3.3 (0.2)	1.2 (0.1)	5	0.17 (0.01)
MCP**	7.0 (0.4)	186.1 (7.0)	0.48 (0.02)	3.1 (0.1)	4.1 (0.3)	2.9 (0.1)	1	0.32 (0.00)
Adaptive LASSO	8.1 (0.5)	92.0 (4.1)	0.60 (0.02)	4.8 (0.1)	6.9 (0.5)	1.2 (0.1)	1	5.36 (0.11)
SCAD	11.2 (0.6)	124.4 (6.2)	0.47 (0.02)	4.0 (0.1)	9.2 (0.5)	2.0 (0.1)	0	0.39 (0.01)
MCP*	11.5 (0.4)	181.4 (6.3)	0.35 (0.02)	2.8 (0.1)	8.3 (0.3)	3.2 (0.1)	0	0.32 (0.00)
Scaled LASSO	16.1 (0.4)	302.4 (9.6)	0.42 (0.01)	4.5 (0.1)	14.6 (0.4)	1.5 (0.1)	0	0.51 (0.01)
LASSO	30.9 (0.6)	111.0 (2.5)	0.36 (0.01)	5.4 (0.1)	30.3 (0.6)	0.6 (0.1)	0	0.40 (0.01)

Table 1: Average metrics over 100 repetitions for each of the procedures, ordered by increasing Hamming distance. Standard errors are shown in parentheses. *`ncvreg` implementation using cross-validation over a one-dimensional grid with a default value of the second tuning parameter γ. **hard thresholding tuning with $\gamma = 1.0001$ and cross-validation over λ.

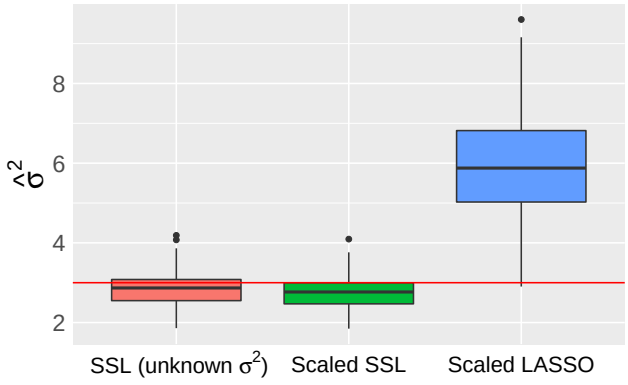


Figure 2: Estimated $\widehat{\sigma}^2_{adj}$ over 100 repetitions. The true variance $\sigma^2 = 3$ is the red horizontal line.

estimates (2.70). For the scaled SSL and the scaled Lasso variance estimates, we also applied a degrees of freedom correction similarly to (2.70) using the number of non-zero coefficients found by each method. The variance estimates from the SSL (unknown σ^2) have a median of 2.87 and standard error 0.04. Meanwhile, the scaled SSL slightly underestimates the variance with a median of 2.76 and standard error 0.04, as expected from the larger number of false positives observed in Table 1. Finally, the scaled Lasso highly inflates the variance with a median of 5.88 and standard error 0.14.

2.7. Protein Activity Data

We now apply the Spike-and-Slab Lasso with unknown variance to the protein activity data set from Clyde and Parmigiani (1998). Following those authors, we code the categorical variables by indicator variables and consider all main effects, two-way interactions and quadratic terms for the continuous variables. This results in a linear model with $p = 88$ potential predictors. The sample size is $n = 96$. We assess the performance of the Spike-and-Slab Lasso with unknown variance in both variable selection and prediction.

2.7.1. Variable Selection

As an approximation to the "truth", we use the Bayesian adaptive sampling algorithm (BAS, Clyde et al., 2011), which has previously been applied successfully to this dataset.

BAS gives posterior inclusion probabilities (PIP) for each of the p potential predictors from which we determined the median probability model (MPM: predictors with PIP $>$ 0.5). The median probability model found by BAS consisted of $q = 7$ predictors: (i) `con:detT`: the interaction of protein concentration and detergent T, (ii) `detT`: detergent T, (iii) `bufTRS:detN`: the interaction of buffer TRS and detergent N, (iv) `con`: protein concentration, (v) `bufPO4:temp`: the interaction of buffer P04 and temperature, (vi) `detN`: detergent N, and (vii) `detN:temp`: the interaction of detergent N and temperature.

For the SSL with unknown variance, we used the same settings as the simulation study with $\lambda_1 = 1$ and $\lambda_0 \in \{1, 2, \ldots, n\}$. The procedure found a model with $\widehat{q} = 6$ predictors, including four of the MPM: `con, detN, bufTRS:detN, con:detT`. Additionally, instead of `detT`, the SSL with unknown variance found the interaction of pH with detergent T (`pH:detT`). The correlation between `detT` and `pH:detT` is 0.988, rendering the two predictors essentially exchangeable. Thus, 5 out of the 6 predictors found by the SSL with unknown variance matched with the benchmark MPM.

For the SSL with known variance, we fixed $\sigma^2 = 0.24$. This is the mean of the scaled-inverse-χ^2 distribution induced by the variance of the response, as detailed in Section 2.6.4. For the protein data, the variance of the response is 0.41 and so fixing $\sigma^2 = 1$ overestimates the variance, resulting in no signal being found. The SSL with this fixed variance found $\widehat{q} = 2$ predictors: one of the MPM (`detT`) and one not in the MPM but having a correlation of 0.735 with `detN`.

Here, we can see the benefit of simultaneously estimating the error variance; the estimate from SSL with unknown variance was $\widehat{\sigma}^2 = 0.167$, resulting in a less sparse solution.

2.7.2. Predictive Performance

We now compare the predictive performance of the SSL with unknown variance with the penalized regression methods from the simulation study in Section 2.6.6. We additionally considered both the SSL with fixed variance (set to $\sigma^2 = 0.24$ as in the previous section)

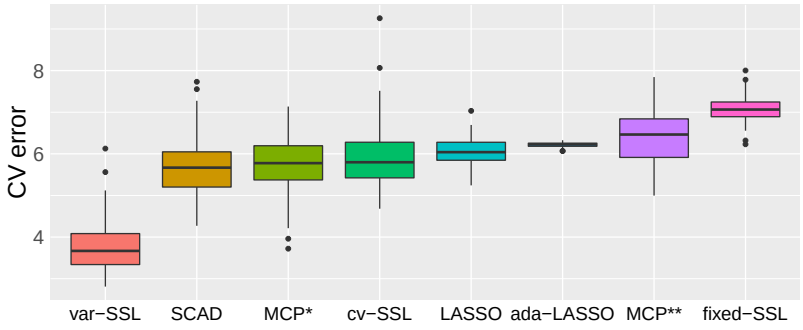


Figure 3: Boxplots of 8-fold cross-validation error over 100 replications for each of the methods (from left to right): 1. SSL (unknown σ^2). 2. SCAD. 3. MCP (`ncvreg`). 4. cv-SSL (fixed σ^2 with cross-validation). 5. LASSO. 6. Adaptive LASSO. 7. MCP ($\gamma = 1.0001$). 8. SSL (fixed σ^2).

and a cross-validated version of fixed variance SSL (cv-SSL): this procedure chooses the values of λ_1 and λ_0 that result in the smallest cross-validation error.

To assess out-of-sample predictive performance, we calculated the 8-fold cross-validation (CV) error of each of the methods as follows. We split the data into $K = 8$ sets, denoting each set by S_k, $k = 1, \ldots, K$. Then, the 8-fold cross-validation error is:

$$CV = \frac{1}{K} \sum_{k=1}^{K} \sum_{i \in S_k} \left[y_i - \mathbf{x}_i \widehat{\boldsymbol{\beta}}_{\backslash k} \right]^2 \tag{2.75}$$

where $\widehat{\boldsymbol{\beta}}_{\backslash k}$ is the estimated regression coefficient using the data in S_k^C. We repeated this procedure for 100 different splits of the data; the resulting cross-validation errors are displayed in Figure 3. We do not display the results from the scaled Lasso in Figure 3 as there were a number of outliers: the cross-validation error for the scaled Lasso was greater than 25 in 20% of the replications.

The SSL with unknown variance has the smallest cross-validation error, highlighting the gains in predictive performance that can be achieved by simultaneously estimating the error variance and regression coefficients. This result is also very encouraging given that all the other methods (except for the fixed variance SSL) use cross-validation to choose

their regularization parameters. Using cross-validation in this way in some sense accounts for and implicitly estimates the error variance. However, in pre-specifying the values the regularization parameter can take, such methods essentially limit the possible values of the error variance. In contrast, the SSL with unknown variance allows for the error variance to be unknown and so can obtain improved estimates of the noise and, consequently, improved out-of-sample predictive performance.

2.8. Conclusion

In this chapter, we have shown that conjugate continuous priors for Bayesian variable selection can lead to underestimation of the error variance when (i) $\boldsymbol{\beta}$ is sparse; and (ii) when p is of the same order as, or larger than, n. This is because such priors implicitly add p "pseudo-observations" of σ^2 which shift prior mass on σ^2 towards zero. Conjugate priors for linear regression are often motivated by the invariance principle of Jeffreys (1961). Revisiting this work however, we highlighted that Jeffreys' himself cautioned against applying his invariance principle in multivariate problems. Following Jeffreys, we recommended priors which treat the regression coefficients and error variance as independent.

We then proceeded to extend the Spike-and-Slab Lasso of Ročková and George (2018) to the unknown variance case, using an independent prior for the variance. We showed that this procedure for the Spike-and-Slab Lasso with unknown variance performs almost as well empirically as the SSL where the true variance is known. We additionally compared the Spike-and-Slab Lasso with unknown variance to a popular frequentist method to estimate the variance in high dimensional regression: the scaled Lasso. In simulation studies, the SSL with unknown variance performed much better than the scaled Lasso and additionally outperformed the "scaled Spike-and-Slab Lasso", a variant of the latter procedure but with the Spike-and-Slab Lasso penalty. On a protein activity dataset, the SSL with unknown variance performed well for both variable selection and prediction. In particular, the SSL with unknown variance exhibited smaller cross-validation error than other penalized likelihood procedures which choose their regularization parameters based on cross-validation.

This highlights the predictive benefit of simultaneous variance estimation. The unknown variance implementation of the SSL is provided in the publicly available R package `SSLASSO` (Ročková and Moran, 2017). Code to reproduce the results in this chapter is also available at `https://github.com/gemma-e-moran/variance-priors`.

2.9. Appendix

2.9.1. Gibbs Sampler for Bayesian Ridge Regression

Here, we present the details of the Gibbs sampler used to obtain posterior estimates for the independent Bayesian ridge regression model in Section 2.3.2. The model is:

$$\mathbf{Y} \sim N_n(\mathbf{X}\boldsymbol{\beta}, \sigma^2\mathbf{I}) \tag{2.76}$$

$$\boldsymbol{\beta} \sim N_p(0, \tau^2\mathbf{I}) \tag{2.77}$$

$$\pi(\sigma) \propto 1/\sigma. \tag{2.78}$$

The full conditional distributions of the parameters $\boldsymbol{\beta}$ and σ^2 are:

$$\boldsymbol{\beta}|\mathbf{Y}, \sigma^2 \sim N_p(\sigma^{-2}\mathbf{V}\mathbf{X}^T\mathbf{Y}, \mathbf{V}) \tag{2.79}$$

$$\sigma^2|\mathbf{Y}, \boldsymbol{\beta} \sim IG(n/2, \|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2/2) \tag{2.80}$$

where $\mathbf{V} = \left[\sigma^{-2}\mathbf{X}^T\mathbf{X} + \tau^{-2}\mathbf{I}_p\right]^{-1}$. The Gibbs sampling algorithm alternates sampling from (2.79) and (2.80). After burn-in, the posterior mean estimates are the means of the samples.

2.9.2. Connections with Penalized Likelihood Methods

To show (4.3) in Section 2.4, consider the objective function (4.1) proposed by Städler et al. (2010) to simultaneously estimate the regression coefficients and error variance in the Lasso. Denote the estimator of the error variance from this objective function by $\widehat{\sigma}^2$. Let $\boldsymbol{\beta}^*$ and σ^* denote the true regression coefficients and error variance, respectively. Sun and Zhang

(2010) proved that $\widehat{\sigma}^2$ will overestimate the true variance unless

$$\lambda \|\boldsymbol{\beta}^*\|_1 / \sigma^* = o(1). \tag{2.81}$$

Suppose now the true dimension of $\boldsymbol{\beta}^*$ is q and that $\max_j |\beta_j^*| = K_1$ for some constant $K_1 \in \mathbb{R}$. Suppose also that the true variance is bounded: $K_2 < \sigma^* < K_3$ for constants $K_2, K_3 \in \mathbb{R}$. Let $\lambda = A\sqrt{(2/n)\log p}$ for some constant $A > 1$ (the universal threshold). Then,

$$\lambda \|\boldsymbol{\beta}^*\|_1 / \sigma^* \leq A\sqrt{(2/n)\log p}\frac{K_1}{K_2} q$$

Hence, from (2.81), the estimator $\widehat{\sigma}^2$ will overestimate the true variance unless

$$q = o\left(\sqrt{\frac{n}{\log p}}\right). \tag{2.82}$$

2.9.3. EM Algorithm for SSL with Conjugate Prior Formulation

Here, we provide the details of the EM algorithm for the Spike-and-Slab Lasso with a conjugate prior formulation in Section 2.6.2. The model is given by:

$$\pi(\boldsymbol{\beta}|\boldsymbol{\gamma}, \sigma^2) \sim \prod_{j=1}^{p} \left(\gamma_j \frac{\lambda_1}{2\sigma} e^{-|\beta_j|\lambda_1/\sigma} + (1-\gamma_j)\frac{\lambda_0}{2\sigma} e^{-|\beta_j|\lambda_0/\sigma} \right) \tag{2.83}$$

$$\boldsymbol{\gamma}|\theta \sim \prod_{j=1}^{p} \theta^{\gamma_j}(1-\theta)^{1-\gamma_j}, \quad \theta \sim \text{Beta}(a, b) \tag{2.84}$$

$$p(\sigma^2) \propto \sigma^{-2}. \tag{2.85}$$

Then, the "complete" data log posterior is given by

$$\begin{aligned}\log \pi(\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma, \theta | \mathbf{Y}) = & -\frac{1}{2\sigma^2}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 - (n+2)\log\sigma \\ & + \sum_{j=1}^{p} \log\left(\gamma_j \frac{\lambda_1}{2\sigma} e^{-|\beta_j|\lambda_1/\sigma} + (1-\gamma_j)\frac{\lambda_0}{2\sigma} e^{-|\beta_j|\lambda_0/\sigma}\right) \\ & + \sum_{j=1}^{p} \log\left(\frac{\theta}{1-\theta}\right)\gamma_j + (a-1)\log(\theta) \\ & + (p+b-1)\log(1-\theta) + C \end{aligned} \tag{2.86}$$

The EM algorithm then proceeds as follows: treat $\boldsymbol{\gamma}$ as unknown and iteratively maximize

$$E[\log \pi(\boldsymbol{\beta}, \boldsymbol{\gamma}, \sigma, \theta | \mathbf{Y}) | \boldsymbol{\beta}^{(k)}, \sigma^{(k)}, \theta^{(k)}] \tag{2.87}$$

where $\boldsymbol{\beta}^{(k)}, \sigma^{(k)}, \theta^{(k)}$ are the parameter values after the kth iteration.

At the $(k+1)$th iteration, these EM updates are then given by:

$$\boldsymbol{\beta}^{(k+1)} = \arg\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2\sigma^{(k)}}\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \sum_{j=1}^{p} |\beta_j| \lambda^*(\beta_j^{(k)}/\sigma^{(k)}; \theta^{(k)}) \right\} \tag{2.88}$$

$$\theta^{(k+1)} = \frac{\sum_{j=1}^{p} p^*(\beta_j^{(k)}/\sigma^{(k)}; \theta^{(k)}) + a - 1}{a + b + p - 2} \tag{2.89}$$

$$\sigma^{(k+1)} = \frac{Q + \sqrt{Q^2 + 4(\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}^{(k)}\|^2)(n+p+2)}}{2(n+p+2)} \tag{2.90}$$

Algorithm 1 Spike-and-Slab Lasso with unknown variance

Input: grid of increasing λ_0 values $I = \{\lambda_0^1, \ldots, \lambda_0^L\}$, update frequency M

Initialize: $\boldsymbol{\beta}^* = \mathbf{0}_p$, σ^{*2}, $\theta^* = 0.5$

For $l = 1, \ldots, L$:

1. Set $k_l = 0$
2. Initialize: $\boldsymbol{\beta}_l^{(k_l)} = \boldsymbol{\beta}^*$, $\theta_l^{(k_l)} = \theta^*$, $\sigma_l^{(k_l)2} = \sigma^{*2}$
3. While $\mathsf{diff} > \varepsilon$
   (i) Increment k_l
   (ii) For $s = 1, \ldots, \lfloor p/M \rfloor$:
      i. Update
$$\Delta \leftarrow \begin{cases} \sqrt{2n\sigma_l^{(k_l)2} \log\left[1/p^*(0;\theta_l^{(k_l)})\right]} + \sigma_l^{(k_l)2}\lambda_1 & \text{if } g\left(0;\theta_l^{(k_l)}\right) > 0 \\ \sigma_l^{(k_l)2}\lambda^*(0;\theta_l^{(k_l)}) & \text{otherwise} \end{cases}$$
      ii. For $j = 1, \ldots, M$: update
$$\beta_{l(s-1)M+j}^{(k_l)} \leftarrow \widetilde{S}\left(z_j, \sigma_l^{(k_l-1)2}\lambda^*(\beta_{l(s-1)M+j}^{(k_l-1)}; \theta_l^{(k_l-1)}), \Delta\right)$$
      iii. Update
$$\theta_l^{(k_l)} \leftarrow \frac{a + \|\boldsymbol{\beta}_l^{(k_l)}\|_0}{a + b + p}$$
      iv. If $k_{l-1} < 100$:
         A. Update
$$\sigma_l^{(k_l)2} \leftarrow \frac{\|\mathbf{Y} - \mathbf{X}\boldsymbol{\beta}_l^{(k_l)}\|^2}{n + 2}$$
      v. $\mathsf{diff} = \|\boldsymbol{\beta}_l^{(k_l)} - \boldsymbol{\beta}_l^{(k_l-1)}\|_2$
4. Assign $\boldsymbol{\beta}^* = \boldsymbol{\beta}_l^{(k_l)}$, $\sigma^{*2} = \sigma_l^{(k_l)2}$, $\theta^* = \theta_l^{(k_l)}$

CHAPTER 3 : Spike-and-Slab Lasso Biclustering

3.1. Introduction

Biclustering has emerged as a popular tool for simultaneously grouping samples and their associated features. Standard clustering methods typically group the samples based on their entire set of features; however, this may be problematic in large datasets where many of the features are not expected to play a role in distinguishing the groups. For example, in gene expression data it is expected that only a small fraction of genes are differentially expressed across groups of interest. If samples are required to be similar over all genes to belong to the same cluster, such groups may be missed. Biclustering methods mitigate this problem by finding subsets of samples that are similar on only a subset of the features. In this way, biclustering methods perform variable selection for clustering. Along with gene expression data (Cheng and Church, 2000), biclustering methods have also been applied to recommender systems, which seek to group consumers based on their ratings of different products (De Castro et al., 2007; Zhu et al., 2016); neuroscience (Fan et al., 2010); and agriculture (Mucherino et al., 2009). Biclustering also yields more interpretable results than clustering; by finding features that are associated with group membership, biclustering methods have identified novel biological modules (Xiong et al., 2013).

The observed data is the matrix of samples by features, denoted by

$$\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N]^T \in \mathbb{R}^{N \times G},$$

where Y_{ij} is the measurement of feature j in sample i for $i = 1, \ldots, N$, and $j = 1, \ldots, G$. The goal is to find submatrices of the data matrix (up to permutation of rows and columns) for which the elements Y_{ij} are "similar". The row and column indices of such a submatrix are then referred to as a "bicluster". In the literature, different notions of similarity have been used to define such submatrices of interest. Generally speaking, these notions of similarity can be grouped into four categories, as outlined by Madeira and Oliveira (2004).

The first category assumes that biclusters manifest as submatrices of constant values; specifically, $\mathbf{Y}$ is assumed to have the following structure:

$$Y_{ij} = \sum_{k=1}^{K} \beta_k I(i, j \in \text{bicluster } k) + \varepsilon_{ij}, \quad i = 1, \ldots, N, \ j = 1, \ldots, G, \tag{3.1}$$

where β_k is the constant value of bicluster k and ε_{ij} is additive noise. Methods which fall into this category include that of Hartigan (1972), the first paper to consider simultaneous clustering of rows and columns. Later, the method Large Average Submatrices (LAM, Shabalin et al., 2009) extended this notion to allow for such constant submatrices to overlap.

The second category extends the constant submatrix assumption to accommodate additive row and column bicluster-specific effects. That is, methods in this category assume the data matrix may be decomposed as:

$$Y_{ij} = \mu + \sum_{k=1}^{K} [x_{ik} + \beta_{jk}] + \varepsilon_{ij}, \quad i = 1, \ldots, N, \ j = 1, \ldots, G, \tag{3.2}$$

where μ is the main effect, x_{ik} is the sample effect for bicluster k, β_{jk} is the feature effect for bicluster k and ε_{ij} is the noise. Here x_{ik} and β_{jk} are non-zero only if sample i and feature j are in the bicluster k. This ANOVA-style decomposition was first utilized by Cheng and Church (2000). Lazzeroni and Owen (2002) also used this definition of a bicluster, naming it the plaid model for the patterns in the data matrix that result from this assumption.

The third category assumes multiplicative row and columns effects, instead of additive. That is, the data matrix is assumed to have the following structure

$$Y_{ij} = \sum_{k=1}^{K} x_{ik}\beta_{jk} + \varepsilon_{ij}, \quad i = 1, \ldots, N, \ j = 1, \ldots, G, \tag{3.3}$$

where the definitions are the same as in (3.2). This assumption allows for biclusters to be found in which samples and features exhibit similar behavior, not just similar values. Specifically, within a bicluster, samples are assumed to be correlated across the features.

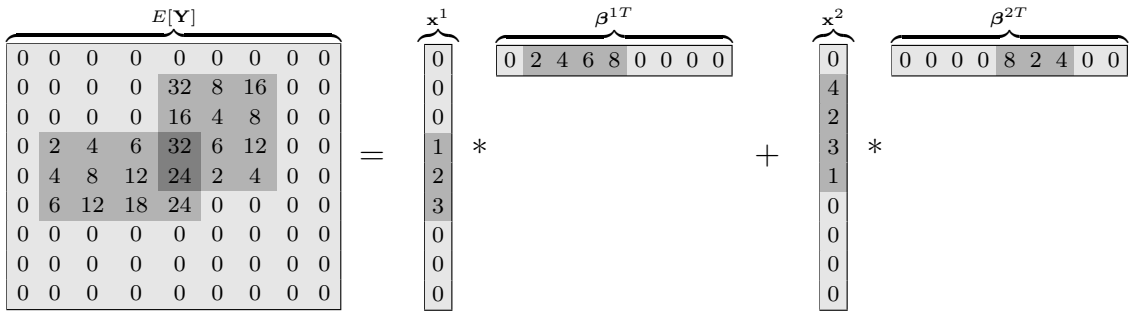


Figure 4: Mean of a data matrix with two biclusters: $E[\mathbf{Y}] = \mathbf{x}^1\boldsymbol{\beta}^{1T} + \mathbf{x}^2\boldsymbol{\beta}^{2T}$.

The model (3.3) corresponds to identifying rank-1 submatrices in the data matrix, up to permutation of rows and columns (see Figure 4). One approach for finding rank-1 structures in the observed data matrix is to use doubly-sparse factor analysis (Hochreiter et al., 2010; Gao et al., 2016). Other methods use Pearson's correlation coefficient as a similarity score and then find rows and columns that have scores above a specified threshold (Bozdağ et al., 2009; Bhattacharya and De, 2009; Bhattacharya and Cui, 2017). Rangan et al. (2018) use a novel "loop-counting" method to find rank-1 submatrices in the data matrix.

Methods in the fourth category do not assume a model for the data matrix but instead search for patterns in the data matrix. Such patterns may be viewed as generalizations of the additive or multiplicative assumptions. For example, the Iterative Signature Algorithm (ISA, Bergmann et al., 2003) finds submatrices in which all rows and all columns are above a certain threshold. Ben-Dor et al. (2003) generalize the multiplicative assumption (3.3) to find subsets of features which have the same order on a subset of samples, which can be thought of as a slightly more flexible correlation structure.

In addition to how they define biclusters, methods can also be classified according to other criteria, including: the types of algorithms they utilize to find such biclusters; the assumptions they make regarding the noise distribution; and whether features and samples are allowed to belong to more than one bicluster, to name a few. For more detailed reviews of biclustering methods, see Madeira and Oliveira (2004); Prelić et al. (2006); Bozdağ et al. (2010); Eren et al. (2012); Padilha and Campello (2017).

3.1.1. Our Approach: Spike-and-Slab Lasso Biclustering

In this chapter, we introduce a new approach for Bayesian biclustering called Spike-and-Slab Lasso Biclustering (SSLB). Our method assumes that biclusters manifest as rank-1 submatrices of the data matrix, $\mathbf{Y}$. This assumption corresponds to a factor analysis model where both the factors and the loadings are sparse. That is, we assume that $\mathbf{Y}$ has the following structure:

$$\mathbf{Y} = \sum_{k=1}^{K} \mathbf{x}^k \boldsymbol{\beta}^{kT} + \mathbf{E}, \tag{3.4}$$

where $\mathbf{X} = [\mathbf{x}^1, \dots \mathbf{x}^K] \in \mathbb{R}^{N \times K}$ is the factor matrix, $\mathbf{B} = [\boldsymbol{\beta}^1, \dots, \boldsymbol{\beta}^K] \in \mathbb{R}^{G \times K}$ is the loadings matrix and $\mathbf{E} = [\boldsymbol{\varepsilon}_1, \dots, \boldsymbol{\varepsilon}_N]^T \in \mathbb{R}^{N \times G}$ is a matrix of Gaussian noise with $\boldsymbol{\varepsilon}_i \overset{ind}{\sim} N_G(\mathbf{0}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} = \text{diag}\{\sigma_j^2\}_{j=1}^G$ for $i = 1, \dots, N$. We allow for the number of biclusters, K to be unknown. We use the convention that the superscript $\mathbf{x}^k$ refers to the kth column of $\mathbf{X}$, and the subscript $\mathbf{x}_i$ to refers to the ith row of $\mathbf{X}$.

In assuming a multiplicative model for the biclusters, our method falls into the third category outlined in the previous section. We prefer this definition of a bicluster for a number of reasons. Firstly, it is interpretable. Using gene expression data as an example: the genes (i.e. features) in a bicluster may be expressed at different levels to drive a biological process. This expression pattern in turn may be weaker or stronger in different samples, as determined by the sample-specific multiplicative effect. Secondly, there are many applications in which features and samples have been shown to be well approximated by such multiplicative effect models (Hochreiter et al., 2010). Thirdly, the definition allows for the specification of the model (3.4), allowing for systematic analysis of the noise variance and, in possible future work, coherent inclusion of prior information regarding the features or the samples.

In (3.4), x_{ik} is non-zero if sample i belongs to bicluster k and β_{jk} is non-zero if feature j belongs to bicluster k. As such, the problem of finding the biclusters in this framework can be viewed as a variable selection problem: identifying biclusters corresponds to finding the

support of $\mathbf{x}^k$ and $\boldsymbol{\beta}^k$. To address this problem of variable selection, we adopt a Bayesian framework and place sparsity-inducing Spike-and-Slab Lasso priors (Ročková and George, 2018) on each of the columns of the factor matrix, $\mathbf{X}$, and and the loadings matrix, $\mathbf{B}$. The Spike-and-Slab Lasso was introduced by Ročková and George (2018) for variable selection in linear regression and has subsequently been used in multivariate regression (Deshpande et al., 2017) and sparse factor analysis (Ročková and George, 2016). A difference here from Ročková and George (2016) is that we induce sparsity in both the factor matrix and the loadings matrix, instead of only the loadings matrix. A benefit of the Spike-and-Slab Lasso is that it can adapt to the underlying levels of sparsity (or lack thereof) in the data. As we will show, this allows the method to find biclusters of a range of different sizes.

To determine the number of biclusters, K, we use a Bayesian nonparametric strategy. Specifically, we use an Indian Buffet Process prior (IBP, Griffiths and Ghahramani, 2005) on the "size" of each bicluster, which ensures that each new bicluster is smaller than the previous one. We also allow for the IBP prior to be extended to a Pitman-Yor IBP (Teh et al., 2007), which drives the size of consecutive biclusters to decrease as a power law. This extension may be appropriate in applications where one expects a larger number of biclusters of a smaller size.

We develop a fast, deterministic EM algorithm with a variational step to find the modal estimates of $\mathbf{X}$ and $\mathbf{B}$. Biclustering is in general NP-hard (Peeters, 2003). The Spike-and-Slab Lasso prior ameliorates such computational difficulties as it uses a continuous relaxation of bicluster membership. This continuous relaxation corresponds to assigning a probability, instead of a binary indicator, to whether each sample or feature is in a particular bicluster.

We note that the factorization (3.4) is similar to the singular value decomposition (SVD) of $\mathbf{Y}$. However, the SVD assumption forces the columns of $\mathbf{X}$ and $\mathbf{B}$ to be orthogonal, a requirement which is relaxed here, as is done in factor analysis more generally. A benefit of not requiring orthogonality is that it allows for biclusters to overlap, enabling samples and

features to belong to more than one bicluster. Further, samples and features do not have to belong to any biclusters.

A potential issue with not requiring orthogonality is that the model (3.4) is not identifiable up to rotation: that is, $\mathbf{XB}^T = (\mathbf{PX})(\mathbf{PB})^T$ for any rotation matrix $\mathbf{P}$. However, by seeking a sparse factorization of $\mathbf{Y}$, the space of rotation matrices is restricted, placing a "soft constraint" on identifiability.

3.1.2. Related Work

There have been a number of biclustering methods which utilize the same factor analysis model as (3.4) and adopt sparsity-inducing priors for the factor and loading matrices. The first method to do so was Factor Analysis for Bicluster Acquisition (FABIA, Hochreiter et al., 2010) who placed single Laplace priors on both $\mathbf{x}^k$ and $\boldsymbol{\beta}^k$. However, the posterior resulting from Laplacian priors does not place enough mass on sparse solutions in variable selection problems (Castillo et al., 2015). This is because such a single Laplace prior has one variance parameter and so cannot both shrink negligible values to zero and maintain the larger signal. As a result, the estimates of $\mathbf{X}$ and $\mathbf{B}$ from FABIA are not sparse; the authors recommend a heuristic thresholding rule to then determine bicluster membership. In contrast, the Spike-and-Slab Lasso performs selective shrinkage on the latent variables; indeed the Spike-and-Slab Lasso concentrates at the optimal rate for sparse models (Ročková, 2018). Further, our method gives an indicator of bicluster membership, precluding the need for an arbitrary thresholding strategy. Finally, FABIA does not automatically select the number of biclusters.

Gao et al. (2016) also begin with the model (3.4) for their method BicMix. They allow for the components $\mathbf{x}^k$ and $\boldsymbol{\beta}^k$ to be either sparse, or dense to account for potential confounders. To achieve strong regularization on the sparse components, the authors utilize a three parameter beta distribution (Armagan et al., 2011), a generalization of the horseshoe prior (Carvalho et al., 2010). Whilst this dichotomous framework may be appropriate

for applications such as genomics, in other applications it may be more appropriate to allow for a continuum of sparsity levels. Such a continuum is achieved in our model as the Spike-and-Slab Lasso prior is indexed by a continuous parameter which controls the proportion of non-zero values in each bicluster. Further, the Spike-and-Slab Lasso automatically thresholds negligible values to zero; such thresholding does not occur automatically for the horseshoe prior and generalizations thereof. Gao et al. (2016) also allow for the number of biclusters, K, to be unknown by starting with an overestimate of K, imposing strong regularization on $\mathbf{X}$ and $\mathbf{B}$, and then removing zero columns. This strategy is similar to to our Bayesian nonparametric strategy; the difference is that the IBP prior which we utilize increases the strength of the regularization of $\mathbf{X}$ and $\mathbf{B}$ as a function of the column number k, as opposed to BicMix which applies the same regularization to each column.

Recently, Denitto et al. (2017) proposed the similarly named method "Spike and Slab Biclustering". Despite this likeness, there are a number of differences between our methods. Firstly, Denitto et al. (2017) utilize Gaussians distributions for their spike and slab priors, whereas we use Laplacian priors. In Bayesian variable selection, the slab distribution requires tails at least as heavy as the Laplace for optimal posterior concentration (Castillo and van der Vaart, 2012). In addition, Denitto et al. (2017) do not allow for the number of biclusters to be unknown.

Xu et al. (2013) proposed a Bayesian biclustering method for count data. To choose the number of biclusters, they implemented a Bayesian non-parametric strategy with Polya-urn priors on the clusters. However, unlike SSLB, their method does not allow for biclusters to overlap. Additionally, they utilize an Markov Chain Monte Carlo (MCMC) strategy to obtain posterior estimates whilst SSLB is implemented by a deterministic optimization algorithm.

3.2. Model

We now introduce Spike-and-Slab Lasso Biclustering (SSLB). We adopt the factor analysis model in (3.4). We first place an inverse gamma prior on the elements of the covariance matrix, $\mathbf{\Sigma}$:

$$\sigma_j^2 \sim IG\left(\frac{\eta}{2}, \frac{\eta\xi}{2}\right). \tag{3.5}$$

To allow for uncertainty in the number of biclusters, K, we initialize the factor and loading matrices with an overestimate, K^*. The IBP prior discourages biclusters with negligible signal from entering consideration, and so the estimated factor and loading matrices will contain columns of all zeroes, provided K^* is a true overestimate. After removing these zero columns, the number of remaining columns is the estimated number of biclusters.

We also restrict $\mathbf{X}$ and $\mathbf{B}$ to be matrices with at least two non-zero entries per column (Fruehwirth-Schnatter and Lopes, 2018; Ročková and George, 2016). This is because a singleton column in either $\mathbf{X}$ or $\mathbf{B}$ will be unidentifiable with regard to the noise matrix $\mathbf{\Sigma}$ in the marginal covariance of $\mathbf{Y}$ (after marginalizing over either $\mathbf{B}$ or $\mathbf{X}$, respectively).

3.2.1. Hierarchical structure for loadings $\mathbf{B}$

For each column $\boldsymbol{\beta}^k$, we have a Spike-and-Slab Lasso prior. That is, each β_{jk} is drawn *a priori* from either a Laplacian "spike" parameterized by λ_0 and is consequently negligible, or a Laplacian "slab" parameterized by λ_1 and thus can be large:

$$\pi(\beta_{jk}|\gamma_{jk}, \lambda_0, \lambda_1) = (1-\gamma_{jk})\psi(\beta_{jk}|\lambda_0) + \gamma_{jk}\psi(\beta_{jk}|\lambda_1), \quad 1 \le j \le G,\ 1 \le k \le K^*, \tag{3.6}$$

where the Laplace density is denoted by $\psi(\beta|\lambda) = \frac{\lambda}{2}e^{-\lambda|\beta|}$ and γ_{jk} is a binary indicator variable. Here, $\gamma_{jk} = 1$ if feature j is active in bicluster k, and $\gamma_{jk} = 0$ if feature j has a negligible contribution to bicluster k. We allow for uncertainty in bicluster membership by

using the common Beta-Bernoulli prior for the latent indicators:

$$\begin{aligned} \gamma_{jk}|\theta_k &\sim \text{Bernoulli}(\theta_k), \\ \theta_k &\sim \text{Beta}(a, b). \end{aligned} \tag{3.7}$$

It is important to emphasize here the "sparsity-indexing" parameter θ_k. Due to the Beta-Bernoulli prior, it has a natural interpretation as the percentage of non-zero elements in the column $\boldsymbol{\beta}^k$. By allowing θ_k to vary continuously, the method can adapt to differing levels of sparsity in each of the different columns of $\mathbf{B} = [\boldsymbol{\beta}^1, \ldots, \boldsymbol{\beta}^K]$.

Here, we can use a finite approximation to the IBP by setting the hyperparameters of the Beta prior in (3.7) to: $a \propto 1/K^*$, $b = 1$. This ensures that in the limit as $K^* \to \infty$, this prior is the IBP. While this is the default choice for these hyperparameters, we note that they can be easily tailored to the problem at hand. For instance, a choice of $a = 1/G$, $b = 1/G$ will result in the prior mass concentrating around $\theta = 0$ and $\theta = 1$, which may be preferred when both very dense and very sparse biclusters are expected.

3.2.2. Hierarchical structure for factors $\mathbf{X}$

To find biclusters, we also want sparsity in the columns of $\mathbf{X}$. To this end, we place a Spike-and-Slab Lasso prior on each x_{ik}. However, we require an alternate formulation of the Spike-and-Slab Lasso prior to Section 3.2.1 for the x_{ik} in order to yield a tractable EM algorithm. This is accomplished by introducing auxiliary variables $\{\tau_{ik}\}_{i,k=1}^{N,K^*}$ for the variance of each $\mathbf{x}_{ik}$:

$$x_{ik}|\tau_{ik} \sim N\left(0, \tau_{ik}\right) \quad 1 \leq i \leq N,\ 1 \leq k \leq K^*. \tag{3.8}$$

Then, the τ_{ik} are each assigned a mixture of exponentials prior, where τ_{ik} is drawn *a priori* from either an exponential "spike" parameterized by $\widetilde{\lambda}_0^2$ and consequently is small, or from

an exponential "slab" parameterized by $\widetilde{\lambda}_1^2$ and hence can be large:

$$\pi(\tau_{ik}|\widetilde{\gamma}_{ik}) = \widetilde{\gamma}_{ik}\frac{\widetilde{\lambda}_1^2}{2}e^{-\widetilde{\lambda}_1^2\tau_{ik}/2} + (1-\widetilde{\gamma}_{ik})\frac{\widetilde{\lambda}_0^2}{2}e^{-\widetilde{\lambda}_0^2\tau_{ik}/2} \tag{3.9}$$

where $\widetilde{\gamma}_{ik}$ is a binary indicator variable. This augmentation strategy uses the fact that the Laplace distribution can be represented as a scale mixture of a normal with an exponential mixing density; marginalizing over the τ_{ik} yields the usual Spike-and-Slab Lasso prior in (3.6).

We place independent Bernoulli priors on each of the $\widetilde{\gamma}_{ik}$ binary indicators. Similarly as before, $\widetilde{\gamma}_{ik} = 1$ if sample i is active in bicluster k, and $\widetilde{\gamma}_{ik} = 0$ if sample i has a negligible contribution to bicluster k. The Bernoulli priors are parameterized by the "sparsity indexing" parameters $\widetilde{\theta}_k$. Instead of placing a Beta prior on the $\widetilde{\theta}_k$ as for the hierarchical model for the loadings $\mathbf{B}$, we use an Indian Buffet Process prior with an optional Pitman-Yor extension. This is achieved using the stick-breaking construction of Teh et al. (2007):

$$\begin{aligned}
\widetilde{\gamma}_{ik} &\sim \text{Bernoulli}(\widetilde{\theta}_{(k)}), \\
\widetilde{\theta}_{(k)} &= \prod_{l=1}^{k}\nu_{(l)}, \\
\nu_{(k)} &\sim \text{Beta}(\widetilde{\alpha}+kd, 1-d), \qquad \text{where } d \in [0,1), \quad \widetilde{\alpha} > -d.
\end{aligned} \tag{3.10}$$

When $d = 0$, the above formulation is the usual IBP prior. When $0 < d < 1$, the ordered sparsity weights, $\widetilde{\theta}_{(k)}$, decrease in expectation as a $O(k^{-1/d})$ power-law (Teh et al., 2007). This may be useful in applications where there are expected to be more, but smaller, biclusters.

We note that we only utilize this stick-breaking formulation of the IBP prior for the sparsity weights for the factors, $\mathbf{X}$, and not the loadings, $\mathbf{B}$. This is because this formulation requires ordering the columns of $\mathbf{X}$ from most dense to least dense. There is no reason to assume that the bicluster with the largest number of samples (i.e. non-zero x_{ik}) would also have the

largest number of features (i.e. non-zero β_{jk}). That is, the most dense column of $\mathbf{X}$ should not be forced to line up with the most dense column of $\mathbf{B}$, which would be the case if we used a similar stick-breaking construction for the priors of $\mathbf{B}$.

In the simulation studies in Section 3.3, we will also consider the finite approximation to the IBP for comparison. Similarly as for the loadings $\mathbf{B}$, this formulation has a Beta prior on the sparsity weights, $\widetilde{\theta}_k \sim \text{Beta}(\widetilde{a}, \widetilde{b})$ with $\widetilde{a} \propto 1/K^*$ and $\widetilde{b} = 1$.

Finally, we use the notation $\mathbf{T} = \{\tau_{ik}\}_{i,k=1}^{N,K^*} \in \mathbb{R}^{N \times K^*}$, $\widetilde{\mathbf{\Gamma}} = \{\widetilde{\gamma}_{ik}\}_{i,k=1}^{N,K^*}$ and $\mathbf{D}_i = \text{diag}\{\tau_{i1}^{-1}, \ldots, \tau_{iK^*}^{-1}\}$.

3.2.3. Implementation

We develop an EM algorithm with a variational step to quickly target modes of the posterior. In the E-Step, we compute the expectation of the factors $\mathbf{X}$ and factor indicators $\widetilde{\mathbf{\Gamma}}$, conditional on the data and current values of the rest of the parameters. This step is rendered tractable by the augmentation strategy outlined in Section 3.2.2. In the M-Step, we marginalize over the loading indicators, $\mathbf{\Gamma}$, and use a coordinate ascent strategy to find the modes of $\mathbf{B}$ (Ročková and George, 2018). For this algorithm, we also use the variance updates detailed by Moran et al. (2018). To maximize the parameters of the IBP prior, we implement a variational step with closed form updates inspired by Doshi et al. (2009). Further details of the algorithm are given in Section 3.7.1 of the Appendix.

We adopt a dynamic posterior exploration strategy for finding the modes of $\mathbf{B}$ (Ročková and George, 2018). Specifically, we hold the slab parameters λ_1 fixed and then gradually increase the spike parameter λ_0 along a "ladder" of values, propagating the solutions forward as "warm starts" for the next largest spike values in the ladder. As outlined by Ročková and George (2018), holding the slab parameter fixed serves to stabilize the large coefficients; this is in contrast to the Lasso, which shrinks the larger coefficients along with the small. Meanwhile, gradually increasing λ_0 over a ladder of values progressively thresholds negligible coefficients to zero.

For the factor matrix, $\mathbf{X}$, we modify this strategy slightly. As we are calculating the conditional mean of $\mathbf{X}$, values of x_{ik} that were previously zero may re-enter the bicluster for very large $\widetilde{\lambda}_0$. This phenomenon is illustrated in the following simple example: suppose the true value is $x_{ik} = 0.005$. Then, the contribution of sample i is essentially negligible and so x_{ik} should reasonably "belong" to the spike. However, if spike parameter is $\widetilde{\lambda}_0 = 200$, it is actually unlikely that x_{ik} was drawn from the spike distribution; this is because this $\widetilde{\lambda}_0$ corresponds to an extremely small spike variance of 5×10^{-5}. This phenomenon is an example of Lindley's paradox. Whilst this phenomenon occurs for both $\mathbf{B}$ and $\mathbf{X}$, we estimate the *mode* of $\mathbf{B}$ which does have this problem, unlike the mean. For estimation of the mean of $\mathbf{X}$, we implement a stopping rule for $\widetilde{\lambda}_0$. We have found that an effective data-driven strategy is to "freeze" $\widetilde{\lambda}_0$ at the value at which $\mathbf{X}$ is the most sparse, whilst continuing to increase λ_0 (the spike parameter for $\mathbf{B}$). To conclude the discussion on the dynamic posterior exploration strategy, we note that we increase λ_0 and $\widetilde{\lambda}_0$ concurrently (up until the point where $\widetilde{\lambda}_0$ is fixed).

We also implement a re-scaling step for the columns of $\mathbf{X}$ and $\mathbf{B}$. Whilst sparsity-inducing priors mitigate to some extent the identifiability problems of the likelihood in regard to rotation, the scale of the columns of the factor and loadings matrices remains unidentifiable. That is, $\mathbf{x}^k \boldsymbol{\beta}^{kT}$ is equivalent to $(c_k^{-1}\mathbf{x}^k)(c_k\boldsymbol{\beta}^k)^T$ for any constant $c_k \in \mathbb{R}$. The focus of biclustering, however, is to find the non-zero elements of these matrices; it is the covarying subsets that are of interest, and not their magnitude. As the scale is not of particular interest, we re-scale $\mathbf{X}$ and $\mathbf{B}$ at each step of the EM algorithm to ensure that the corresponding columns have the same norm. That is, for each $k = 1, \ldots, K$, we set

$$c_k \leftarrow \sqrt{\frac{\|\mathbf{x}^k\|_1}{\|\boldsymbol{\beta}^k\|_1}}, \quad \mathbf{x}^k \leftarrow \frac{1}{c_k}\mathbf{x}^k, \quad \boldsymbol{\beta}^k \leftarrow c_k\boldsymbol{\beta}^k. \tag{3.11}$$

The re-scaling step is also important to ensure that the default choices of regularization parameters $\lambda_0, \widetilde{\lambda}_0$ are appropriate; if $\mathbf{X}$ and $\mathbf{B}$ have vastly different scales, then one matrix may be over-thresholded whilst the other is under-thresholded.

A benefit of SSLB is that the binary variables γ_{jk} and $\widetilde{\gamma}_{ik}$ indicate whether feature j and sample i, respectively, are active in bicluster k. To find the modes of $\mathbf{B}$, we marginalize over the $\{\gamma_{jk}\}_{j,k=1}^{G,K^*}$ and so bicluster membership is determined simply from the support of $\mathbf{B}$. As the Spike-and-Slab Lasso prior automatically thresholds small values to zero, no further thresholding is required for $\mathbf{B}$. For the factors $\mathbf{X}$, we instead use the posterior mean of $\widetilde{\gamma}_{ik}$ (calculated in the E-Step) to determine bicluster membership. Specifically, we implement the following thresholding rule after convergence of the SSLB algorithm:

$$\widehat{x}_{ik} = \begin{cases} \widehat{x}_{ik} & \text{if } E[\widetilde{\gamma}_{ik}|\mathbf{Y}, \mathbf{T}^*, \widetilde{\boldsymbol{\theta}}^*] > 0.5, \quad 1 \leq i \leq N,\ 1 \leq k \leq K^* \\ 0 & \text{if } E[\widetilde{\gamma}_{ik}|\mathbf{Y}, \mathbf{T}^*, \widetilde{\boldsymbol{\theta}}^*] \leq 0.5, \end{cases} \tag{3.12}$$

where $\mathbf{T}^*$ and $\widetilde{\boldsymbol{\theta}}^*$ are the solutions obtained after convergence of the EM algorithm. That is, if the posterior probability of x_{ik} belonging to the "spike" is greater than 0.5, it is thresholded to zero.

The complexity of the SSLB algorithm is $O(NK^{*3}+GK^*)$, assuming that the initial number of biclusters, K^*, is less than both the number of samples, N, and the number of features, G. The first term comes from the E-Step for $\mathbf{X}$, where the $K^* \times K^*$ matrix $\mathbf{V}^i$ needs to be inverted for $i = 1, \ldots, N$. The second term comes from the M-Step for $\mathbf{B}$, where the coordinate ascent algorithm has complexity K^* and is applied to each of the G rows. However, the E-Step and M-Step are trivially parallelizable across the samples and features, respectively. Such a parallelization would yield an improved complexity of $O(K^{*3})$.

3.2.4. Connection to PX-EM

We pause for a moment to discuss the connections between our model for $\mathbf{X}$ (3.8) and parameter-expansion (PX-EM) methods for factor analysis (Liu et al., 1998; Ročková and George, 2016). The usual factor analysis framework generally takes a standard normal prior for the factors $\mathbf{x}_i \sim N(0, \mathbf{I})$. The PX-EM strategy is to add an auxiliary variance parameter to this prior to aid in navigating the posterior. For example, Ročková and George (2016)

take $\mathbf{x}_i \sim N(0, \mathbf{A})$ with $\pi(\mathbf{A}) \propto 1$ to yield an algorithm which rotates to regions of the posterior where $\mathbf{B}$ is sparse. In (3.8), we also place a more structured prior on the factors to help guide the search for doubly-sparse factorizations of $\mathbf{Y}$. A major difference between our approaches, however, is that PX-EM strategies implement the E-Step with respect to the standard normal prior (Ročková and George, 2016). Our E-Step is with respect to the augmented prior (3.8) as we want to retain the prior sparsity constraint for $\mathbf{X}$.

3.2.5. Default Settings

The default hyper-parameters settings are as follows. For both the loadings and the factors, $\mathbf{B}$ and $\mathbf{X}$, the slab parameters are set to $\lambda_1, \widetilde{\lambda}_1 = 1$ and the increasing ladder of spike parameters are set to $\lambda_0, \widetilde{\lambda}_0 \in \{1, 5, 10, 50, 100, 500, 1000, 10000, 100000, 1000000, 10000000\}$. Note, however, that $\widetilde{\lambda}_0$ is halted at a data-driven value as described earlier in this section.

To determine the hyper-parameters of the variances, $\{\sigma_j\}_{j=1}^G$, we use an informal empirical Bayes strategy, motivated by Chipman et al. (2010). We denote the sample variances of the columns of the data matrix, $\mathbf{Y}$, by $\{s_j^2\}_{j=1}^G$. Then, the intuition for our strategy is as follows: if we assume that most biclusters are sparse, then small values of the s_j^2 are essentially "pure noise" and contain no signal. Hence, the prior for the error variances should be centered around a small value of s_j^2. In addition, we recommend using a small value of the degrees of freedom parameter, η, to allow for larger prior uncertainty. As a default, we take $\eta = 3$. More specifically, we calculate the 5% quantile of the s_j^2 and find the value of ξ such that this 5% quantile is the median of the prior distribution.

We initialize the parameters of SSLB as follows. Each entry of $\mathbf{B}$ is generated independently from a standard normal distribution. The entries of $\mathbf{T}$, the matrix of auxiliary variance parameters, are set to 100, representing an initial relatively non-informative prior on $\mathbf{X}$. The sparsity weights, θ_k, are initialized at 0.5. The IBP parameters, $\boldsymbol{\nu}$, are generated independently from a Beta$(1, 1)$ distribution and then ordered from largest to smallest.

For the initialization of K, we recommend $K^* = 50$. If SSLB does not remove any biclusters,

we recommend running SSLB again with a larger initial K until SSLB finds fewer biclusters than the initial number.

3.3. Simulation Studies

In this section, we compare the performance of SSLB to the methods of BicMix and FABIA (outlined in Section 4.3) in two simulation settings. Similarly to Gao et al. (2016), the simulation studies we present illustrate the performance of our method on settings with different levels of sparsity in the biclusters. Specifically, the first simulation study considers matrices with only sparse biclusters while the second simulation study considers both sparse and dense biclusters.

3.3.1. Simulation 1

We first consider a simulated example with $N = 300$, $G = 1000$ and $K = 15$ biclusters. The data was simulated using settings very similar to the FABIA paper (Hochreiter et al., 2010). Specifically, the data matrix $\mathbf{Y}$ was generated as $\mathbf{XB}^T + \mathbf{E}$, where each entry of the noise matrix $\mathbf{E}$ is sampled from an independent standard normal distribution. For each column $\mathbf{x}^k$, we draw the number of samples in bicluster k uniformly from $\{5, \ldots, 20\}$. The indices of these elements were randomly selected and then assigned a value from $N(\pm 2, 1)$, with the sign of the mean chosen randomly. The elements of $\mathbf{x}^k$ not in the bicluster had values drawn from $N(0, 0.2^2)$. The columns $\boldsymbol{\beta}^k$ were generated similarly, except the number of elements each bicluster was drawn from $\{10, \ldots, 50\}$. We allow biclusters to share at most five samples and at least fifteen features. For both SSLB and BicMix, we set the initial overestimate of the number of biclusters to be $K^* = 30$. For FABIA, we set the number of biclusters to the truth, $K = 15$.

For each of the methods, we recorded the following metrics: (i) relevance and recovery (Prelić et al., 2006); and (ii) consensus (Hochreiter et al., 2010) (see Section 3.7.2 of the Appendix for precise definitions). Relevance measures how similar on average the biclusters found by a method are to the true biclusters (where similarity is defined by the Jaccard

index). Recovery instead measures how similar the true biclusters are to the found biclusters on average. However, if many duplicated biclusters are found by a method, this will not be reflected in either the relevance or recovery scores. To provide a meaningful metric in such circumstances, Hochreiter et al. (2010) developed the consensus score. The consensus score is similar to the recovery score, but penalizes overestimation of the true number of biclusters.

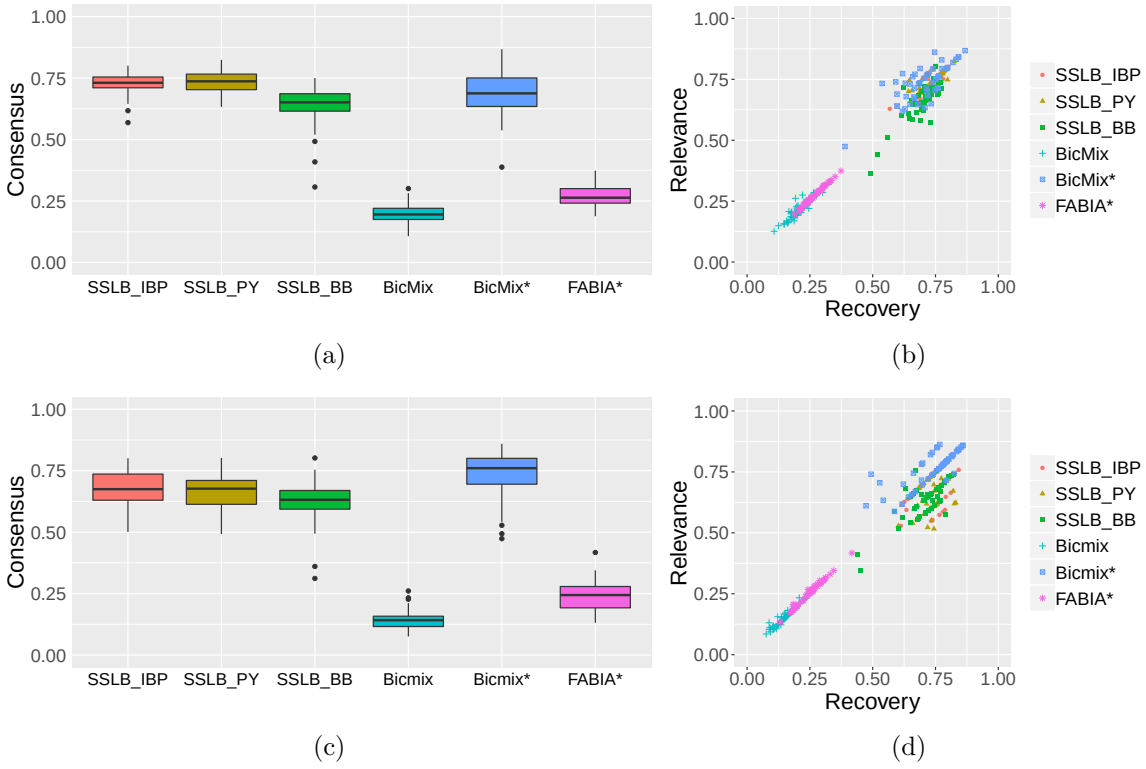
In this simulation study, we compared three implementations of Spike-and-Slab Lasso Biclustering: (i) SSLB with the Pitman-Yor extension where $\widetilde{\alpha} = 1$ and $d = 0.5$ (SSLB-PY); (ii) SSLB with the stick-breaking IBP prior for the factors where $\widetilde{\alpha} = 1$ (SSLB-IBP), and (iii) SSLB with the finite approximation to the IBP prior (i.e. Beta-Binomial) for the factors where $\widetilde{a} = 1/K^*$ and $\widetilde{b} = 1$ (SSLB-BB). For each implementation, we used the default settings as outlined in Section 3.2.5. For the loadings matrix, $\mathbf{B}$, we set the Beta-Binomial hyperparameters to be $a = 1/K^*$, $b = 1$.

BicMix[1] was implemented using the default parameters. Following Gao et al. (2016), we thresholded values less than 10^{-10}. We also considered the "best-thresholded" solution of BicMix (referred to as BicMix*); this is the thresholded solution of BicMix that attains the highest consensus score over a grid of 100 threshold values, equally spaced in $[0.1, 5]$. FABIA was implemented using the `fabia` R package (Hochreiter et al., 2010) with the default parameters and recommended post-processing thresholding step. We additionally consider the "best thresholded" solution for FABIA (referred to as FABIA*); this is obtained similarly to BicMix*.

For 50 realizations of the simulated data, we ran each method and calculated their consensus score (Figure 5a), and relevance and recovery scores (Figure 5b). All implementations of SSLB have higher consensus, relevance and recovery scores than the other methods. We display only FABIA*, as even the best-thresholded solution of FABIA is not competitive with SSLB in this simulation study, even when FABIA is initialized with the true number of

[1]Code obtained from `beehive.cs.princetone.edu/software`

Figure 5: (a) Boxplots of the consensus scores for Simulation 1. (b) Relevance versus recovery scores for Simulation 1. (c) Boxplots of the consensus scores for Simulation 2. (d) Relevance versus recovery scores for Simulation 2. `BicMix*` and `FABIA*` refer to the best-thresholded solutions of BicMix and FABIA, respectively.



biclusters. The lower scores of BicMix are due to small values not being thresholded exactly to zero by the three-parameter beta prior. This can be seen as the best-thresholded version, BicMix*, achieves consensus scores with a slightly lower median than the nonparametric implementations of SSLB, albeit with a higher variance in scores. We emphasize, however, that BicMix* requires oracular knowledge of the true bicluster structure, which is of course not known in practice. In contrast, SSLB achieves high scores on all metrics without the need for a post-processing thresholding step.

Table 2 displays the estimated number of biclusters, $\widehat{K}$, from SSLB and BicMix. Both the IBP and Pitman-Yor implementations of SSLB are centered at the truth. We can see empirically the benefit of using the stick-breaking construction for the IBP prior here; the SSLB-BB formulation with the finite IBP approximation slightly overestimates the

true number of biclusters. Meanwhile, BicMix slightly underestimates the true number of biclusters.

	$\widehat{K}$	
Method	Simulation 1	Simulation 2
Truth	*15*	*9*
SSLB (IBP)	15.0 (0.09)	9.9 (0.14)
SSL (PY)	15.0 (0.09)	10.1 (0.15)
SSLB (BB)	16.4 (0.24)	10.3 (0.14)
BicMix	14.5 (0.18)	8.7 (0.10)
BicMix*	14.5 (0.18)	8.7 (0.10)

Table 2: Mean estimated number of biclusters, K, over 50 replications. Standard errors are shown in parentheses. BicMix* refers to the "best-thresholded" solution of BicMix.

3.3.2. Simulation 2

We now assess how well SSLB can find both sparse and dense biclusters with a simulation study inspired by that of Gao et al. (2016). We again take $N = 300, G = 1000$ and $K = 15$. For both the factor and loading matrices, five columns are dense and ten columns are sparse. The sparse columns (corresponding to sparse biclusters) are generated as Simulation 1. The dense columns (corresponding to dense biclusters) are generated as independent $N(0, 2^2)$. We allow for one dense column in $\mathbf{X}$ to correspond to a sparse column in $\mathbf{B}$ and vice versa; this results in $K = 9$ biclusters which are sparse in both $\mathbf{X}$ and $\mathbf{B}$.

The goal for this simulation study is to recover the sparse biclusters while removing the effect of the dense biclusters, which are acting as confounders. As such, we calculate the recovery, relevance and consensus scores for the sparse biclusters found by each of the methods only. For SSLB, we determine a "sparse" bicluster to be one where both columns $\mathbf{x}^k$ and $\boldsymbol{\beta}^k$ have less than 50% of values being non-zero. BicMix provides a binary indicator for whether $\mathbf{x}^k$ and $\boldsymbol{\beta}^k$ are sparse or dense; we kept BicMix biclusters for which both $\mathbf{x}^k$ and $\boldsymbol{\beta}^k$ were sparse. Before running the FABIA algorithm, we removed the first six principal components of $\mathbf{Y}$. Without this adjustment, FABIA performs extremely poorly as it is unable to adapt

to differing levels of sparsity in the biclusters. In a similar simulation study, Gao et al. (2016) also considered this adjusted version of FABIA for a more fair comparison. As we already removed the dense biclusters, we then considered all biclusters found by FABIA as "sparse" for the purposes of computing the recovery, relevance and consensus scores.

For 50 replications of the data, we calculated the consensus scores for each method (Figure 5c) and the relevance and recovery scores (Figure 5d). Here, the best-thresholded version of BicMix (BicMix*) has slightly higher consensus scores than the SSLB implementations. Again, we emphasize that BicMix* is a thresholded solution which uses knowledge of the true bicluster membership. The slightly lower consensus scores of SSLB are a result of SSLB overestimating the number of biclusters by one (Table 2). However, this additional bicluster that SSLB finds is not spurious; it is the bicluster where the true $\mathbf{x}^k$ is sparse and $\boldsymbol{\beta}^k$ is dense. In SSLB, the estimated $\boldsymbol{\beta}^k$ is not completely dense and so is included in the count.

3.4. Breast Cancer Microarray Dataset

We now assess the performance of SSLB on a benchmark gene expression microarray dataset. The dataset[2] consists of the expression levels $G = 24,158$ genes from the breast cancer tumors of $N = 337$ patients with stage I or II breast cancer (Van De Vijver et al., 2002; Van't Veer et al., 2002). Gao et al. (2016) also used this dataset to illustrate the performance of their biclustering method, BicMix. We followed a similar data processing pipeline to Gao et al. (2016) (details in Section 3.7.3). However, unlike Gao et al. (2016), we did not project the quantiles of the expression levels to a standard normal. We chose not to do so to assess the ability of SSLB to capture biological signal in the presence of possible confounders. Removing of unwanted variation via matrix factorization (specifically, via singular value decomposition) has been shown to be an effective technique by previous authors (Leek and Storey, 2007), albeit not in the context of biclustering.

[2]Data sourced from R package `breastCancerNKI` (Schroeder et al., 2011)

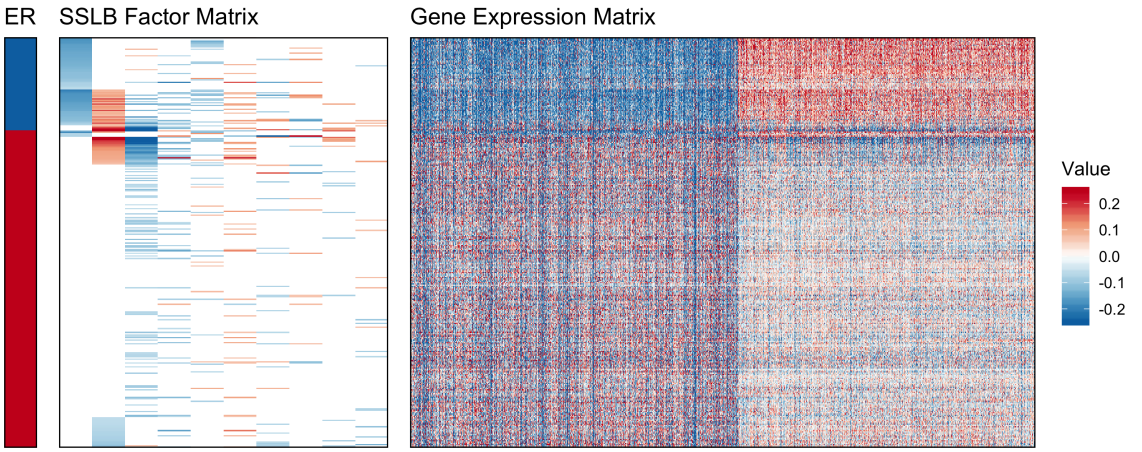


Figure 6: Left: Clinical Estrogen Receptor (ER) status (Blue = ER-, Red = ER+). Middle: SSLB factor matrix where each row corresponds to a patient and each column corresponds to a bicluster. A patient belongs to a bicluster if they have a non-zero value in that column. Rows are ordered by clinical ER status; within ER status, rows are ordered by factor values in biclusters 1 and 2. Only the first 10 biclusters (ordered by size) are shown for improved visualization; full factor matrix is displayed in Section 3.7.4. Right: normalized submatrix of gene expression values where rows correspond to all samples (ordered by ER status) and columns correspond to genes in Bicluster 1 (re-ordered according to their loadings in Bicluster 1). Expression values with magnitude greater than 0.25 have had magnitude set to 0.25 for improved visualization.

We ran SSLB-IBP with the initial number of biclusters set to $K^* = 50$. We set the Beta-Binomial hyperparameters to $a = 1/(GK^*)$ and $b = 1$, and the IBP hyperparameter to $\widetilde{\alpha} = 1/N$. For the remaining parameters, we use the default settings outlined in Section 3.2.5. SSLB-IBP found $\widehat{K} = 30$ biclusters (Figure 6).

3.4.1. SSLB identifies subtypes of breast cancer

Breast cancers can be broadly grouped into subtypes based on the expression levels of two genes: ESR1, which encodes an estrogen receptor (ER), and ERBB2, which encodes the human epidermal growth factor receptor 2 (HER2) (Horta and Campello, 2014). A patient is deemed ER-positive (-negative) if they have relatively high (low) expression levels of ESR1. HER2 status is similarly defined by the expression of ERBB2. The expression levels of these genes determine four subtypes of breast cancer: (i) ER+/HER2+, (ii) ER+/HER2-, (iii) ER-/HER2+ and (iv) ER-/HER2-. These subtypes have been shown to be valuable prog-

nostic indicators and are used to determine the treatment protocol for patients (Horta and Campello, 2014). The clinical ER status of patients (determined by immunohistochemical staining, not gene expression levels) was provided with the dataset and so can provide a measure of validation for the biclusters that SSLB found. The HER2 status of patients was not recorded, however.

SSLB found four biclusters with significantly different means in the factors between the clinically ER-negative and ER-positive patients[3]. The patients with negative factors in SSLB bicluster 1 are almost all patients whose clinical status was recorded as ER-negative (Figure 6). We then investigated the genes in this bicluster and found ESR1, the gene encoding an estrogen receptor, was down-regulated for these patients. There are five patients with clinical ER-positive status who were in the ER-negative bicluster found by SSLB. However, the down-regulation of the ESR1 gene in this patients suggests that the original clinical characterization was a misclassification. In the original paper analyzing this data, Van De Vijver et al. (2002) also found five patients had a discrepancy between their clinical ER-status and gene expression determined ER status, concluding that the latter classification was correct.

The gene ERBB2 is present in SSLB biclusters 1 and 2. In both biclusters, ERBB2 is up-regulated for patients with positive factors and down-regulated for patients with negative factors. For patients with negative bicluster 1 and zero bicluster 2 factors, ESR1 and ERBB2 are both down-regulated, indicating ER-/HER2- status. Meanwhile, patients with negative bicluster 1 and positive bicluster 2 factors are likely ER-/HER2+. Turning to the ER-positive patients (with zero bicluster 1 values), those with positive bicluster 2 values are potentially ER+/HER2+. Finally, ER-positive patients with negative bicluster 2 factors are likely ER+/HER2-. We note that a number of patients are in neither bicluster 1 or 2; we hypothesize that these patients are also ER+/HER2- as this is the most common breast cancer subtype (Onitilo et al., 2009). The proportions of patients in each subtype found by SSLB matches fairly well with reported subtype proportions in the literature (Table 3).

[3]Biclusters 1, 2, 5 and 22 had p-values, 6.1×10^{-50}, 2.2×10^{-9}, 1.0×10^{-5} and 7.2×10^{-6}, respectively, from a Wilcoxon rank-sum test with Bonferroni significance level $0.01/\widehat{K}$

	ER+/HER2+	ER+/HER2-	ER-/HER2+	ER-/HER2-
Onitilo et al. (2009)	10.2%	68.9%	7.5%	13.4%
SSLB	7.7%	70.3%	8.9%	13.1%

Table 3: Proportion of breast cancer patients in each of the subtypes determined by ER and HER2 status from (i) the study of Onitilo et al. (2009); and (ii) SSLB.

After determining these groups, we then investigated whether genes known to play a role in these subtypes were present in the biclusters. In particular, genes considered to be indicators (or markers) of ER+ status are KRT8, GATA-3, XBP-1, FOXA1 and ADH1B (Zhang et al., 2014). Four of these five marker genes were down-regulated in bicluster 1, and consequently were relatively over-expressed for the ER+ patients (p-value 0.002, Fisher's exact test). The gene GRB7 is located adjacent to the ERBB2 (HER2) gene and as such is often co-expressed with ERBB2; we indeed found that GRB7 was up-regulated in bicluster 2 (as well as down-regulated for the HER2- patients in bicluster 1).

3.4.2. Gene Ontology Enrichment Analysis

We next conducted gene ontology enrichment analysis on the genes found by SSLB using the R package `clusterProfiler` (Yu et al., 2012). This software conducts an overrepresentation test to determine whether genes which coordinate the same biological process are significantly co-occuring. If a subset of genes is found to be overrepresented in a set, the set is said to be "enriched" for the biological process in which those genes are active. With a false discovery rate (FDR) threshold of 0.05, we found that the genes which were up-regulated in SSLB bicluster 1 (corresponding to the ER-negative patients) were enriched for 124 biological processes. Many of these were related to cell proliferation, including the G1/S transition of mitotic cell cycle. As cancer is fundamentally the un-regulated growth of cells, such proliferation signatures are commonly found in tumor samples (Whitfield et al., 2006). Another biological process for which the ER-negative bicluster is enriched is: response to leukemia inhibitory factor. Leukemia inhibitory factor has actually been shown to stimulate cell proliferation in breast cancer (Kellokumpu-Lehtinen et al., 1996). An en-

richment map summarizing the most statistically significant processes is displayed in Figure 12a (Section 3.7.4 of the Appendix).

The genes up-regulated in the HER2+ patients in SSLB bicluster 2 were enriched for 495 biological processes (again with FDR threshold of 0.05). The enrichment map summarizing these processes is displayed in Figure 12b (Section 3.7.4 of the Appendix). In particular, these genes were enriched for the Wnt signaling pathway, the over-expression of which has been implicated in the development of cancer (Zhan et al., 2017). Further, stem cell proliferation was enriched in this bicluster; stem cells have been implicated as possible originators of tumors, and may in some cases potentially drive tumorigenesis (Reya et al., 2001).

Overall, 86.6% of the biclusters found by SSLB were enriched for biological processes. Further investigation of the remaining biclusters and their potential clinical utility may be interesting future work.

3.4.3. Comparison with BicMix and FABIA

We ran BicMix on this data using the default settings; however, BicMix found zero biclusters. This is in contrast to the results of (Gao et al., 2016) on this dataset: the difference here is because we did not use quantile normalization, unlike Gao et al. (2016) who projected the quantiles of the gene expression levels to the standard normal distribution.

We ran FABIA on this dataset using the default settings for two different bicluster initializations: (i) $K = 10$ and (ii) $K = 50$, as FABIA does not automatically select the number of biclusters (Figure 7). In the $K = 10$ setting, FABIA found five biclusters that had a significantly different mean between ER+ and ER- patients (p-values 3.9×10^{-24}, 2.5×10^{-12}, 9.2×10^{-10}, 4.8×10^{-8}, 1.7×10^{-7} from Wilcoxon rank-sum test with Bonferroni significance level 0.01/10). Unlike SSLB, however, FABIA does not find a bicluster with almost exclusively ER-negative patients.

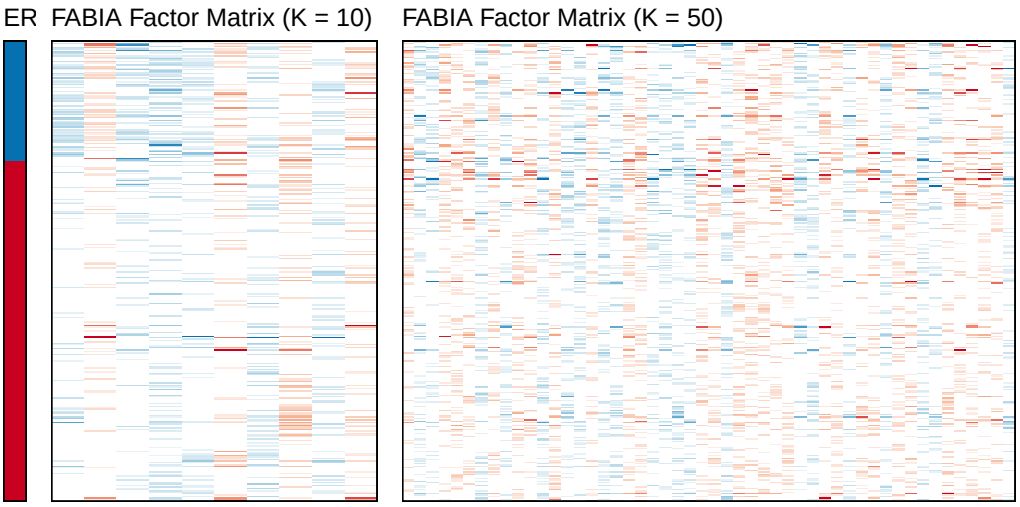


Figure 7: Left: Clinical Estrogen Receptor (ER) status (Blue = ER-, Red = ER+). Middle: FABIA factor matrix (initial $K^* = 10$) with rows ordered by clinical ER status. Right: FABIA factor matrix (initial $K^* = 50$) with rows ordered by clinical ER status.

In the $K = 50$ setting, FABIA found three biclusters that had a significantly different mean between ER+ and ER- patients (p-values 2.3×10^{-5}, 5.7×10^{-5}, 2.0×10^{-4} from Wilcoxon rank-sum test with Bonferroni significance level 0.01/50). We can see that with a larger number of initial biclusters, the ER signal is diluted across multiple biclusters. As a result, the conclusions of FABIA seem to be highly dependent on the initial number of biclusters. Further, for this larger value of K, FABIA also does not find a bicluster consisting of almost exclusively ER-negative patients. In contrast, SSLB was initialized with 50 biclusters and then determined $\widehat{K} = 30$ biclusters were sufficient, and found a bicluster consisting of almost all ER-negative patients (apart from the five patients whose clinical measurement was most likely misclassified).

3.5. Mouse Cortex and Hippocampus scRNA-seq Dataset

For our second application, we assess the performance of SSLB on the data of Zeisel et al. (2015) (hereafter referred to as Z15). Z15 used single-cell RNA-sequencing (scRNA-seq) to obtain counts of RNA molecules in 3005 cells from the mouse somatosensory cortex and hippocampal CA1 region. The goal of the study was to characterize the different cell types in

mouse brains by using the cell-specific RNA expression levels, or transcription profiles. For this purpose, Z15 developed a biclustering algorithm called BackSPIN which identified nine major types of cells in the mouse brain based on their transcription profiles: (i) interneurons; (ii) S1 pyramidal neurons; (iii) CA1 pyramidal neurons; (iv) oligodendrocytes; (v) microglia cells; (vi) endothelial cells; (vii) astrocytes; (viii) ependymal cells; and (ix) mural cells. By repeatedly applying BackSPIN on these biclusters, Z15 found a further 47 subclasses of cells. Here, we apply SSLB to the same dataset. A benefit of SSLB is that it can find classes and subclasses simultaneously without having to iteratively re-apply the method.

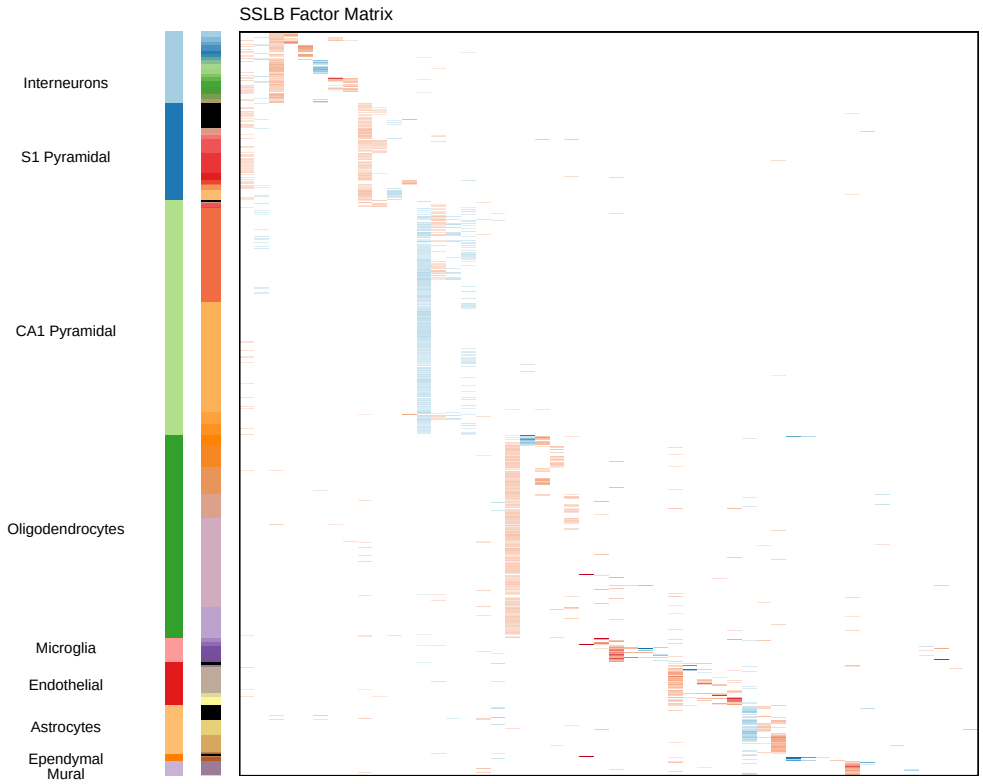
The scRNA-seq dataset made available by Z15 consists of RNA molecule counts for 19,972 genes in 3005 individual cells[4]. Following these authors, we (i) removed genes with less than 25 molecules in total over all cells; (ii) removed genes that were not correlated with more than 5 other genes; and (iii) retained the top 5000 most biologically variable genes. Further details of these processing steps are given in Section 3.7.5 of the Appendix. Although more sophisticated methods for removing technical variability in scRNA-seq data have been developed in recent years (for example, Huang et al., 2018), we follow the steps of Z15 to enable a direct comparison of our biclustering results.

After processing the data, the subset we used for biclustering is a matrix containing the RNA counts of $G = 5000$ genes in $N = 3005$ individual cells. We note that as a matrix of counts, this data is perhaps best modeled by a Poisson distribution, instead of assuming normally distributed residuals as in SSLB. However, Poisson-distributed data with a large rate parameter is approximately normal. As we are considering the most variable genes (with high RNA molecule counts), such a normal approximation is not too unreasonable. Despite this, there are still a high proportion of zero entries in the matrix and so this application may be seen as a test of the robustness of SSLB to model misspecification. We ran SSLB-IBP with the initial number of biclusters set to $K^* = 100$. We set the Beta-Binomial hyperparameters to $a = 1/(GK^*)$ and $b = 1$, and the IBP hyperparameter to

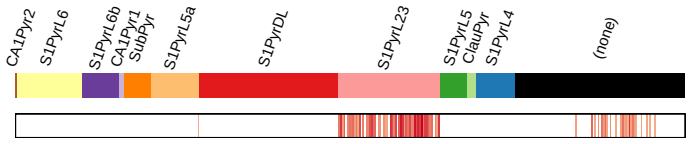
[4] `http://linnarssonlab.org/cortex`

$\widetilde{\alpha} = 1/N$. For the remaining parameters, we use the default settings outlined in Section 3.2.5. SSLB returned $\widehat{K} = 95$ biclusters.

Figure 8: Zeisel dataset: SSLB results



(a) Left: Cell types found by Z15. Middle: Cell subtypes found by Z15. The rows colored black were not assigned a subtype by Z15. Right: SSLB factor matrix with rows ordered to correspond to the Z15 cell types. Each row corresponds to a cell and each column corresponds to a bicluster. A cell belongs to a bicluster if they have a non-zero value in the bicluster (column). Factor values have been capped for improved visualization.



(b) "Zoom in" on S1 Pyramidal cells with subtypes annotated by Z15. Top: subtypes of S1 Pyramidal cells. Bottom: Column 10 of the SSLB factor matrix, corresponding to the cells in bicluster 10. SSLB groups a subset of the uncategorized "(none)" cells as of the S1PyrL23 subtype. (Colors have been modified from Figure 8a for improved visualization.)

3.5.1. SSLB recovers major cells types

SSLB recovered the nine major cell classes identified by Z15, finding a specific bicluster for each class except for the microglia class, which SSLB split into two biclusters (Figure 8a). For each class, Z15 also identified one or two potential marker genes; that is, a gene that is almost exclusively expressed in that cell class. Encouragingly, the SSLB biclusters corresponding to the major cell classes all contained the associated marker gene for that cell class. More specifically:

- The interneuron gene marker *Pnoc* was found in three SSLB biclusters, one corresponding to the major interneuron cell class and the others to subclasses of interneurons.

- The S1 pyramidal neuron marker genes *Gm*11549 and *Tbr*1 were present in two biclusters, one corresponding to the major S1 pyramidal neuron cell class and the other to a subclass of S1 pyramidal neurons. *Tbr*1 was also found in a bicluster containing cells from four different cell types, a potential false positive.

- The CA1 pyramidal neuron marker *Spink*8 was found in three biclusters. Two of these biclusters corresponded to the major CA1 pyramidal neuron cell class and a subclass of CA1 pyramidal neurons, respectively. The third bicluster contained CA1 pyramidal, S1 pyramidal and interneuron cells, suggesting that *Spink*8 may not necessarily be an exclusive marker for CA1 pyramidal neurons.

- The oligodendrocyte marker *Hapln*2 was active in three SSLB biclusters, all corresponding to either the major oligodendrocyte cell class or a subclass of oligodendrocytes. Interestingly, one of these biclusters contained 17 cells, all oligodendrocytes, but did not correspond to one of the Z15 identified subclasses; as such, this bicluster may correspond to a yet-to-be classified subtype of oligodendrocytes. Figure 15 shows the biological processes that are enriched in this bicluster, which can be broadly grouped into two categories: (i) processes related to oligodendrocyte-specific functions, includ-

ing myelination, and (ii) cell metabolic processes.

- The endothelial cell marker *Ly6c1* was found in four SSLB biclusters, two corresponding to the major endothelial group or a subclass. The other two biclusters were mostly all endothelial cells, but contained some astrocytes and microglia cells also.

- The mural cell marker *Acta2* was active in three SSLB biclusters. One bicluster corresponded to the main mural bicluster and another to a bicluster with almost all mural cells. The third bicluster contained mostly endothelial cells, with a few oligodendrocyte, microglia, astrocyte and mural cells, indicating that either *Acta2* is not exclusively expressed in mural cells, or a potential false positive of SSLB.

In addition to the nine main cell types, SSLB found two biclusters (biclusters 1 and 2) which contained many interneurons, S1 pyramidal neurons and CA1 pyramidal neurons. This is unsurprising as these cell types are all subsets of neurons, and so we would expect them to have more similar expression profiles than the other (non-neuronal) brain cells. We conducted gene ontology enrichment analysis on the genes SSLB found in these biclusters. With an FDR threshold of 0.05, bicluster 1 was enriched for 154 biological processes, the majority of which were related to cell metabolic processes and synaptic activity, as may be expected for neurons (Figure 14a). Bicluster 2 was similarly enriched for processes relating to synaptic activity, including axonal transport and synaptic signaling (Figure 14b).

The results of SSLB yield a number of observations that may warrant further scientific investigation. Firstly, while SSLB recovered the major cell types, it grouped together a number of the 47 sub-categories found by Z15. This was particularly the case for the interneuron cells, where SSLB found 5 subtypes (Z15 found 16), and the S1 pyramidal cells, where SSLB found 3 subtypes (Z15 found 12). It may be the case that SSLB has trouble finding more granular clusters, or potentially there really are fewer cell subtypes than identified by Z15.

Although SSLB collapsed many of the interneuron and S1 pyramidal subtypes, it found

many more subtypes of microglia and ependymal cells than Z15. This suggests that there could be a great deal of heterogeneity in expression levels in these classes of cells, a phenomenon which may prove to be of scientific interest.

There are a number of cells which Z15 did not assign to a subtype (colored in black in Figure 8a). Interestingly, SSLB grouped a number of the previously unclassified S1 pyramidal cells into the "S1PyrL23" subtype (Figure 8b).

Finally, we conducted gene ontology enrichment analysis[5] for all of the biclusters found by SSLB. In this analysis, 83% of the biclusters identified by SSLB were enriched for at least one biological process.

3.5.2. Comparison with BicMix and FABIA

We also applied both BicMix and FABIA to the Zeisel dataset. We used the default settings for both methods with initial number of clusters $K^* = 100$. For BicMix, we thresholded values less than 10^{-10} as recommended by Gao et al. (2016). For FABIA, we implemented the recommended post-processing thresholding step (Hochreiter et al., 2010). BicMix found $\widehat{K} = 94$ biclusters (Figure 9) while FABIA found $\widehat{K} = 99$ biclusters (Figure 13 in Section 3.7.6 of the Appendix). BicMix finds many of the smaller subtypes defined by Z15 but assigns the major cell type signals to dense biclusters. This is a result of the dichotomous nature of BicMix; it finds either completely dense or very sparse biclusters. In contrast, SSLB can adapt to the underlying sparsity, allowing it to also estimate such "medium"-sized biclusters. Meanwhile, FABIA finds many larger biclusters but does not do well at recovering the more granular cell subtypes. This is due to FABIA having the same thresholding parameter for each bicluster; it is unable to adapt to the differing levels of sparsity.

[5]Using `clusterProfiler` with FDR threshold of 0.05. We took the 5000 genes obtained after processing as the "background" genes for the overrepresentation test instead of the original number of 19,972 to avoid selection bias.

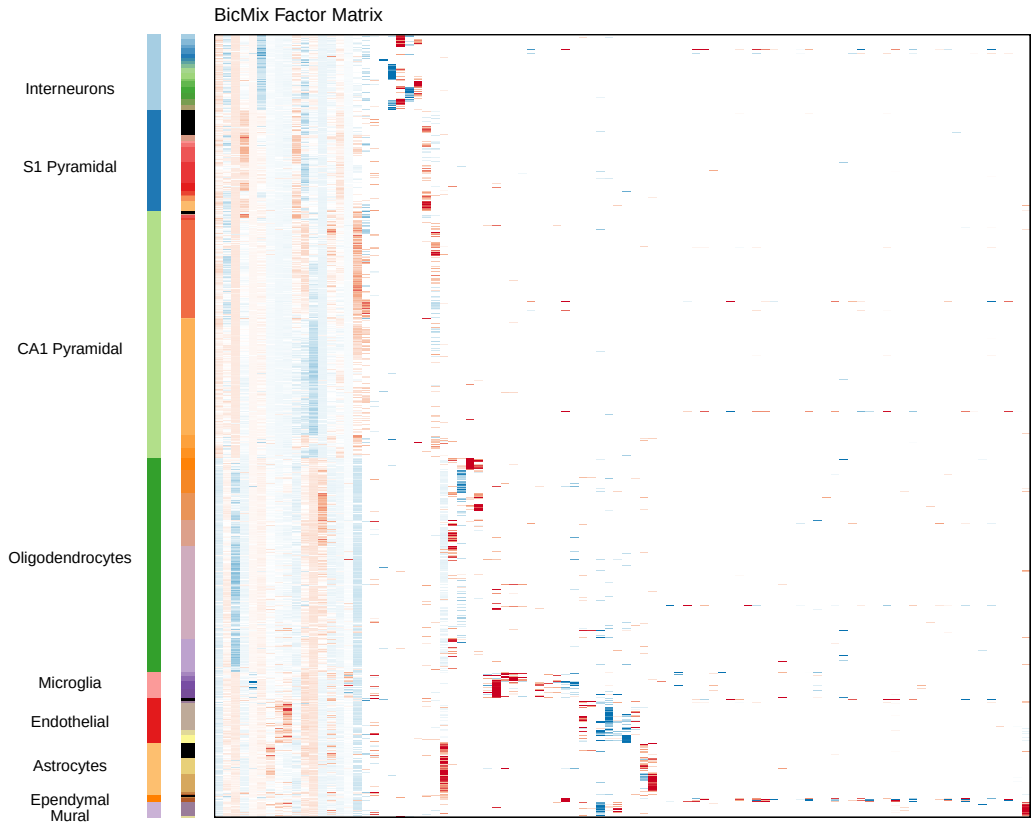


Figure 9: Factor matrix found by BicMix. On the side of the factor matrix are the cell types and subtypes found by Z15, respectively. The rows of the factor matrices have been ordered to correspond to the Zeisel cell types. Factor values have been capped for improved visualization.

3.6. Conclusion

In this chapter, we introduced a new method for biclustering called Spike-and-Slab Lasso Biclustering (SSLB). SSLB finds subsets of samples which co-vary on subset of features. These paired subsets manifest as rank-1 submatrices in the data, referred to as "biclusters" in this setting. To find these biclusters, SSLB conducts doubly-sparse factor analysis in which both the loadings and the factors are sparse. To induce this sparsity in the loadings and factors, SSLB uses the Spike-and-Slab Lasso prior of Ročková and George (2018). This prior is combined with an Indian Buffet Process prior to automatically choose the number of biclusters. SSLB utilizes a fast EM algorithm with a variational step to find the modes

of the posterior. This EM algorithm is rendered tractable by a novel augmentation of the Spike-and-Slab Lasso prior.

SSLB features a number of benefits over similar biclustering methods. Firstly, the adaptivity inherent in the Spike-and-Slab Lasso prior allows for SSLB to find a continuum of biclusters of different sizes. This is in contrast to other biclustering methods which have more restrictive assumptions on the sizes of the biclusters. Secondly, the Spike-and-Slab Lasso prior automatically thresholds negligible bicluster values to zero and so SSLB does not require a post-processing thresholding step, unlike other biclustering methods.

SSLB out-performs a number of alternative biclustering methods on a variety of simulated data. On the breast cancer microarray dataset of Van De Vijver et al. (2002); Van't Veer et al. (2002), SSLB finds biclusters corresponding to different subtypes of breast cancer. These biclusters also contained genes which were enriched for a variety of biological processes related to breast cancer. Finally, we applied SSLB to the mouse cortex and hippocampus single-cell RNA-sequencing dataset of Zeisel et al. (2015). SSLB recovered all the major cell classes found by Zeisel et al. (2015) as well as many of the cell subclasses. This performance was achieved despite the non-Gaussianity of the residual noise in the data, highlighting the potential robustness of SSLB to model misspecification. However, it would be interesting to explicitly extend SSLB to non-Gaussian residual noise models in future work.

3.7. Appendix

3.7.1. SSLB Algorithm

In this section, we provide details for the EM algorithm we use to find the modes of the posterior. Before outlining the EM algorithm, we first marginalize over the binary indicator variables $\boldsymbol{\Gamma}$ (associated with the loadings $\mathbf{B}$) to yield the non-separable Spike-and-Slab Lasso prior (Ročková and George, 2018). For each column $\boldsymbol{\beta}_k$, the log of this prior (up to

an additive constant) is:

$$\log \pi(\boldsymbol{\beta}_k) = \sum_{j=1}^{G} -\lambda_1|\beta_{jk}| + \log[p^*(0;\theta_{jk})/p^*(\beta_{jk};\theta_{jk})], \tag{3.13}$$

$$\text{where } p^*(\beta;\theta) = \theta\psi(\beta|\lambda_1)/[\theta\psi(\beta|\lambda_1) + (1-\theta)\psi(\beta|\lambda_0)] \tag{3.14}$$

and $\theta_{jk} = E[\theta_k|\boldsymbol{\beta}_{k\setminus j}]$ where $\boldsymbol{\beta}_{k\setminus j}$ denotes the vector $\boldsymbol{\beta}_k$ with the jth element removed. When G is large, $\boldsymbol{\beta}_{k\setminus j}$ is very similar to $\boldsymbol{\beta}_k$, so this expectation may be approximated by $E[\theta_k|\boldsymbol{\beta}_k]$.

We are now in a position to describe the EM algorithm. We find the expectation of $\mathbf{X}$ and factor indicators $\widetilde{\boldsymbol{\Gamma}}$ with respect to the complete log posterior and then maximize the resultant objective function:

$$Q(\boldsymbol{\Delta}) = \mathbb{E}_{\mathbf{X},\widetilde{\boldsymbol{\Gamma}}|\boldsymbol{\Delta}^{(t)},\mathbf{Y}}\left[\log \pi(\boldsymbol{\Delta}, \mathbf{X}, \widetilde{\boldsymbol{\Gamma}}|\mathbf{Y})\right], \tag{3.15}$$

where we have used the notation $\boldsymbol{\Delta} = \{\mathbf{B}, \boldsymbol{\Sigma}, \mathbf{T}, \boldsymbol{\nu}\}$ to denote the parameters over which we will maximize. For convenience, we will use the notation $\mathbb{E}_{\mathbf{X},\widetilde{\boldsymbol{\Gamma}}|\boldsymbol{\Delta}^{(t)},\mathbf{Y}}(Z) = \langle Z \rangle$.

Now, due to the separability of the parameters in the posterior, we may write

$$Q(\boldsymbol{\Delta}) = Q_1(\mathbf{B}, \boldsymbol{\Sigma}) + Q_2(\mathbf{T}, \boldsymbol{\nu}) + Q_3(\boldsymbol{\nu}) + C, \tag{3.16}$$

where $Q_1(\mathbf{B}, \boldsymbol{\Sigma}) = \langle \pi(\mathbf{B}, \boldsymbol{\Sigma}, \boldsymbol{\Gamma}, \mathbf{X}|\mathbf{Y}) \rangle$, $Q_2(\boldsymbol{\tau}, \boldsymbol{\nu}) = \langle \pi(\mathbf{X}, \mathbf{T}, \widetilde{\boldsymbol{\Gamma}}, \boldsymbol{\nu}|\mathbf{Y}) \rangle$, $Q_3(\boldsymbol{\nu}) = \langle \pi(\boldsymbol{\nu}, \widetilde{\boldsymbol{\Gamma}}|\mathbf{Y}) \rangle$ and $C \in \mathbb{R}$ is a constant.

The first term of the above objective function is:

$$\begin{aligned} Q_1(\mathbf{B}, \boldsymbol{\Sigma}) = C - \frac{1}{2}\sum_{i=1}^{N} \left\{ (\mathbf{y}_i - \mathbf{B}\langle \mathbf{x}_i \rangle)^T \boldsymbol{\Sigma}^{-1} (\mathbf{y}_i - \mathbf{B}\langle \mathbf{x}_i \rangle) + \text{tr}[\mathbf{B}'\boldsymbol{\Sigma}^{-1}\mathbf{B}(\langle \mathbf{x}_i \mathbf{x}_i' \rangle - \langle \mathbf{x}_i \rangle \langle \mathbf{x}_i \rangle')] \right\} \\ - \sum_{k=1}^{K^*} \log \pi(\boldsymbol{\beta}_k) - \frac{N+\eta+2}{2} \sum_{j=1}^{G} \log \sigma_j^2 - \sum_{j=1}^{G} \frac{\eta\xi}{2\sigma_j^2}, \end{aligned}$$

where $\pi(\boldsymbol{\beta}_k)$ is defined in (3.13). Next,

$$Q_2(\mathbf{T}) = -\frac{1}{2}\sum_{i=1}^{N}\left\{\langle \mathbf{x}_i\rangle^T \mathbf{D}_i \langle \mathbf{x}_i\rangle + \mathrm{tr}[\mathbf{D}_i(\langle \mathbf{x}_i\mathbf{x}_i'\rangle - \langle \mathbf{x}_i\rangle\langle \mathbf{x}_i\rangle')\right\} - \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K^*}\log\tau_{ik}$$
$$- \frac{1}{2}\sum_{i=1}^{N}\sum_{k=1}^{K^*}\left[\langle\widetilde{\gamma}_{ik}\rangle\widetilde{\lambda}_1^2 + (1-\langle\widetilde{\gamma}_{ik}\rangle)\widetilde{\lambda}_0^2\right]\tau_{ik}. \tag{3.17}$$

and finally,

$$Q_3(\boldsymbol{\nu}) = \sum_{k=1}^{K^*}\left[\langle\widetilde{\gamma}_k\rangle\log\prod_{l=1}^{k}\nu_l + (N-\langle\widetilde{\gamma}_k\rangle)\log\left(1-\prod_{l=1}^{K}\nu_l\right)\right]$$
$$+ \sum_{k=1}^{K^*}\left[(\widetilde{\alpha}+kd-1)\log\nu_k - d\log(1-\nu_k)\right]. \tag{3.18}$$

where $\langle\widetilde{\gamma}_k\rangle = \sum_{i=1}^{N}\langle\widetilde{\gamma}_{ik}\rangle$.

E-Step

The conditional posterior distribution of $\mathbf{x}_i$ is given by:

$$\pi(\mathbf{x}_i|\mathbf{B}^{(t)},\boldsymbol{\Sigma}^{(t)},\mathbf{T}^{(t)},\mathbf{y}_i) \sim N(\mathbf{V}^i\mathbf{B}'^{(t)}[\boldsymbol{\Sigma}^{(t)}]^{-1}\mathbf{y}_i,\mathbf{V}^i), \tag{3.19}$$

where $\mathbf{V}^i = [\mathbf{B}'^{(t)}[\boldsymbol{\Sigma}^{(t)}]^{-1}\mathbf{B}^{(t)} + \mathbf{D}_i^{(t)}]^{-1}$. Further, let $\mathbf{V} = \sum_{i=1}^{N}\mathbf{V}^i$.

We now determine the update for the indicators of the factors, $\widetilde{\boldsymbol{\Gamma}}$. Note that conditional on τ_{ik}, $\widetilde{\gamma}_{ik}$ is independent of x_{ik}. We have:

$$\begin{aligned}\langle\widetilde{\gamma}_{ik}\rangle &= P\left(\widetilde{\gamma}_{ik}=1|\mathbf{T},\widetilde{\boldsymbol{\theta}}\right)\\ &= \frac{\pi(\tau_{ik}|\widetilde{\gamma}_{ik}=1)\pi(\widetilde{\gamma}_{ik}=1|\widetilde{\theta}_k)}{\pi(\tau_{ik}|\widetilde{\gamma}_{ik}=1)\pi(\widetilde{\gamma}_{ik}=1|\widetilde{\theta}_k)+\pi(\tau_{ik}|\widetilde{\gamma}_{ik}=0)\pi(\widetilde{\gamma}_{ik}=0|\widetilde{\theta}_k)}\\ &= \frac{\widetilde{\theta}_k\widetilde{\lambda}_1^2 e^{-\widetilde{\lambda}_1^2\tau_{ik}/2}}{\widetilde{\theta}_k\widetilde{\lambda}_1^2 e^{-\widetilde{\lambda}_1^2\tau_{ik}/2}+(1-\widetilde{\theta}_k)\widetilde{\lambda}_0^2 e^{-\widetilde{\lambda}_0^2\tau_{ik}/2}}.\end{aligned} \tag{3.20}$$

M-Step

Let $\mathbf{y}^1, \ldots, \mathbf{y}^G$ be the columns of $\mathbf{Y}$. Denote $\langle \mathbf{X} \rangle = [\langle \mathbf{x}_1 \rangle, \ldots, \langle \mathbf{x}_N \rangle]$ and let $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_G$ be the rows of $\mathbf{B}$. Then

$$Q_1(\mathbf{B}, \boldsymbol{\Sigma}) = \sum_{j=1}^{G} Q_j(\boldsymbol{\beta}_j, \sigma_j) \tag{3.21}$$

where

$$Q_j(\boldsymbol{\beta}_j, \sigma_j) = -\frac{1}{2\sigma_j^2}\|\mathbf{y}^j - \mathbf{X}\boldsymbol{\beta}_j\|^2 - \frac{1}{2\sigma_j^2}\boldsymbol{\beta}_j^T \mathbf{V} \boldsymbol{\beta}_j - \sum_{k=1}^{K^*} \log \pi(\boldsymbol{\beta}_k) - \frac{N+\eta+2}{2} \log \sigma_j^2 - \frac{\eta\xi}{2\sigma_j^2} \tag{3.22}$$

To find a maximum of (3.22) with regard to $\boldsymbol{\beta}_j$, we use the refined thresholding scheme of Ročková and George (2018) with the extension to the unknown variance case given in Moran et al. (2018). Evaluation of $\log \pi(\boldsymbol{\beta}_k)$ requires the expectation of θ_k given the previous values of the loadings, $\boldsymbol{\beta}_k^{(t-1)}$; this yields the following update for θ_k (Ročková and George, 2018):

$$\theta_k^{(t)} = \frac{a + \|\boldsymbol{\beta}_k^{(t-1)}\|_0}{a + b + G}. \tag{3.23}$$

The update for σ_j^2 is:

$$\sigma_j^{2(t)} = \frac{\|\mathbf{y}^j - \mathbf{X}\boldsymbol{\beta}_j^{(t)}\|^2 + \boldsymbol{\beta}_j^{(t)T} \mathbf{V} \boldsymbol{\beta}_j^{(t)} + \eta\xi}{N + \eta + 2}. \tag{3.24}$$

The update for τ_{ik} is given by:

$$\tau_{ik}^{(t)} = \frac{-1 + \sqrt{1 + 4\widetilde{\lambda}_{ik}(\langle x_{ik} \rangle^2 + V_{kk}^i)}}{2\widetilde{\lambda}_{ik}} \tag{3.25}$$

where $\widetilde{\lambda}_{ik} = \langle \widetilde{\gamma}_{ik} \rangle \widetilde{\lambda}_1^2 + (1 - \langle \widetilde{\gamma}_{ik} \rangle) \widetilde{\lambda}_0^2$.

We now consider the update for the IBP stick-breaking parameters $\boldsymbol{\nu}$. This involves finding the $\boldsymbol{\nu}$ that maximize the objective in equation $Q_3(\boldsymbol{\nu})$. The difficulty in maximizing this objective is the non-linear term $\log\left(1 - \prod_{l=1}^{k} \nu_l\right)$. We find a lower bound for this term using a variational approximation inspired by Doshi et al. (2009).

This approximation begins with writing the non-linear term as a telescoping sum. Then, we introduce a parameter $\mathbf{q}_k = (q_{k1}, \ldots, q_{kk})$ where $\sum_{m=1}^{k} q_{km} = 1$, which allows the use of Jensen's inequality:

$$\begin{aligned}
\log\left(1 - \prod_{l=1}^{k} \nu_l\right) &= \log\left(\sum_{m=1}^{k} (1 - \nu_m) \prod_{l=1}^{m-1} \nu_l\right) \\
&= \log\left(\sum_{m=1}^{k} q_{km} \frac{(1 - \nu_m) \prod_{l=1}^{m-1} \nu_l}{q_{km}}\right) \\
&\geq \sum_{m=1}^{k} q_{km} \left[\log(1 - \nu_m) + \sum_{l=1}^{m-1} \log \nu_l\right] - \sum_{m=1}^{k} q_{km} \log q_{km}. \qquad (3.26)
\end{aligned}$$

To make the bound (3.26) as tight as possible, we maximize over the parameter $\mathbf{q}_k$ to obtain updates $\widehat{\mathbf{q}}_k$:

$$\widehat{q}_{km}^{(t)} = \frac{\left(1 - \nu_m^{(t-1)}\right) \prod_{l=1}^{m-1} \nu_l^{(t-1)}}{1 - \prod_{l=1}^{k} \nu_l^{(t-1)}}. \qquad (3.27)$$

The lower bound for the objective function for $\boldsymbol{\nu}$ at iteration t is now:

$$\begin{aligned}
Q_3(\boldsymbol{\nu}) \geq \sum_{k=1}^{K^*} \left[\langle \widetilde{\gamma}_k \rangle \sum_{l=1}^{k} \log \nu_l + (N - \langle \widetilde{\gamma}_k \rangle) \left[\sum_{m=1}^{k} q_{km}^{(t)} \left(\log(1 - \nu_m) + \sum_{l=1}^{m-1} \log \nu_l\right)\right]\right] \\
+ \sum_{k=1}^{K^*} [(\widetilde{\alpha} + kd - 1) \log \nu_k - d \log(1 - \nu_k)]. \qquad (3.28)
\end{aligned}$$

Maximizing the lower bound (3.28) over $\boldsymbol{\nu}$ then yields closed form updates:

$$\nu_k^{(t)} = \frac{r_k^{(t)}}{r_k^{(t)} + s_k^{(t)}} \qquad (3.29)$$

where

$$r_k^{(t)} = \sum_{m=k}^{K^*} \langle \widetilde{\gamma}_k \rangle + \sum_{m=k+1}^{K^*} (N - \langle \widetilde{\gamma}_k \rangle) \left(\sum_{i=k+1}^{m} q_{mi}^{(t)} \right) + \widetilde{\alpha} + kd - 1 \tag{3.30}$$

$$s_k^{(t)} = \sum_{m=k}^{K^*} (N - \langle \widetilde{\gamma}_k \rangle) q_{mk}^{(t)} - d. \tag{3.31}$$

3.7.2. Bicluster Quality Metrics

Here we provide the formulas for the (i) relevance; (ii) recovery; and (iii) consensus scores used to evaluate biclusters in the simulation studies. Each of these scores use the Jaccard index, a measure of similarity between two sets A and B, defined as:

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{3.32}$$

The Jaccard index naturally penalizes methods which find spurious bicluster elements. The relevance and recovery scores were proposed by Prelić et al. (2006) and are defined below. Denote bicluster C_k as the set non-zero entries of the vectorized matrix $\mathbf{x}^k \boldsymbol{\beta}^{kT}$. Let M_t be the set of true biclusters and let M_f be the set of biclusters found by a particular method. Then the relevance and recovery scores are given by:

$$\text{Relevance} = \frac{1}{|M_f|} \sum_{C_1 \in M_f} \max_{C_2 \in M_t} J(C_1, C_2),$$

$$\text{Recovery} = \frac{1}{|M_t|} \sum_{C_2 \in M_t} \max_{C_1 \in M_f} J(C_1, C_2).$$

The consensus score of Hochreiter et al. (2010) is computed as follows.

1. Compute the Jaccard similarity matrix, where the (i, j)th entry is the Jaccard similarity score (3.32) between the ith bicluster in M_t and the jth bicluster in M_f;

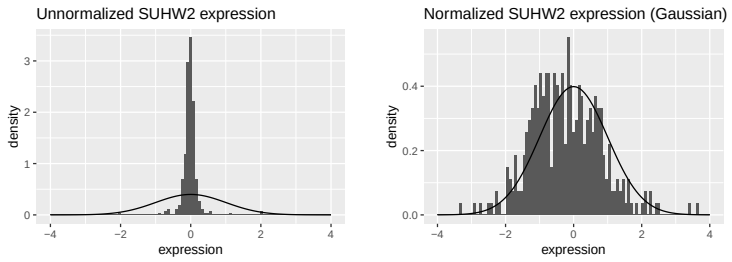
2. Find the optimal assignment (based on the highest Jaccard scores) of the true set of biclusters to the found set of biclusters using the Hungarian algorithm (Munkres,

1957);

3. Sum the similarity scores of the assigned biclusters and divide by $\max\{|M_t|, |M_f|\}$.

3.7.3. Processing Breast Cancer Data

Here, we provide more details on the processing of the breast cancer dataset in Section 3.4. We first removed genes with more than 10% of values missing and imputed the remaining missing values using the R package `impute` (Hastie et al., 2018). We chose not to project the quantiles of the gene expression levels to the standard normal distribution, as done by Gao et al. (2016). This is because the unnormalized gene expression values were mostly clustered around zero with heavy tails (Figure 10a). Although SSLB assumes that the errors are normally distributed, the gene loadings $\{\beta_{jk}\}_{j,k=1}^{G,K}$ are assumed to be drawn a priori from either a Laplacian spike concentrated around zero or a Laplacian slab. We assume that such a mixture model is flexible enough to model the gene expression levels exemplified in Figure 10a.



(a) (b)

Figure 10: Histogram of (a) unnormalized expression values for gene $SUHW2$, (b) quantile normalized expression values for gene $SUHW2$ with standard normal distribution as reference. For both histograms, a standard normal density is overlaid.

3.7.4. Supplementary Figures for Breast Cancer Dataset

Here, we provide supplementary figures for the analysis of the breast cancer microarray dataset in Section 3.4. Enrichment maps (Figure 12) were created using the R package `enrichplot` (Yu, 2018) and display the top 30 biological processes (with lowest FDR q-values satisfying threshold of 0.05) found in the gene ontology enrichment analysis as de-

scribed in Section 3.4.2.

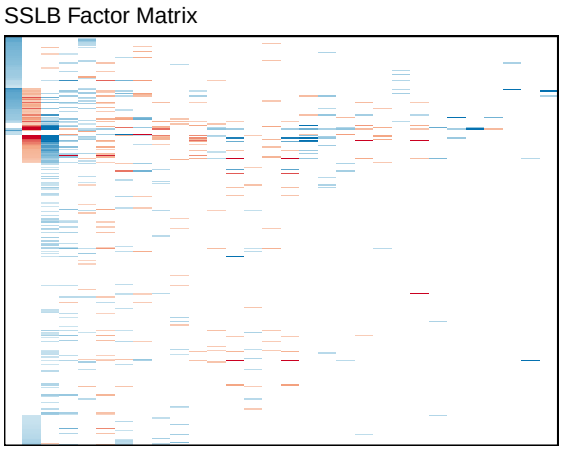
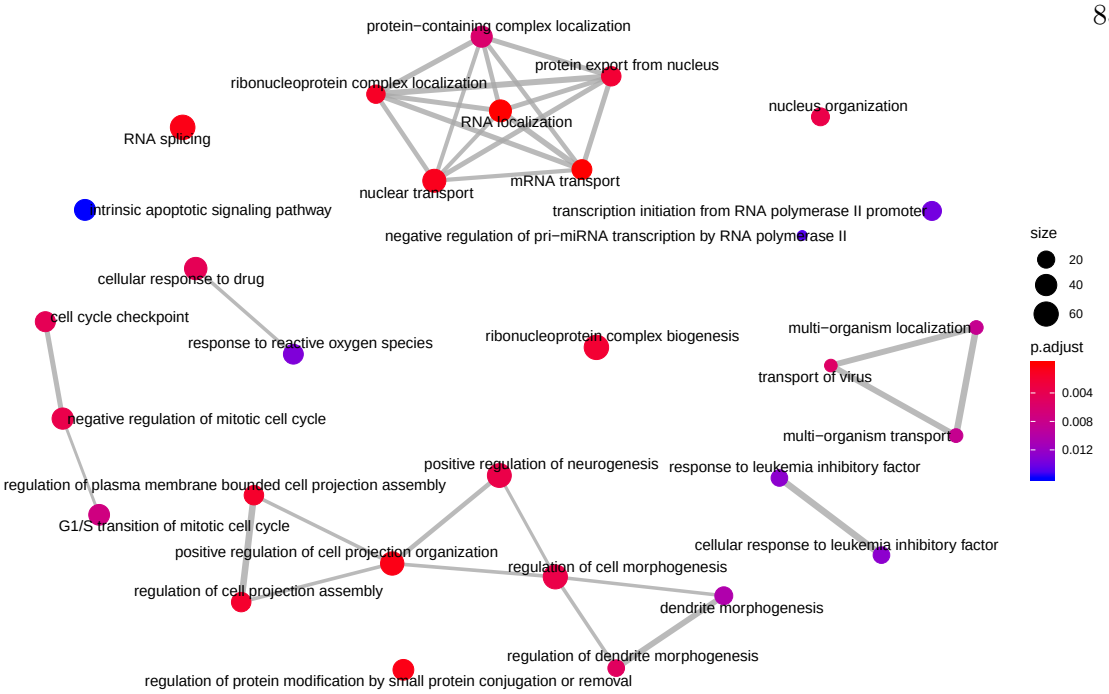
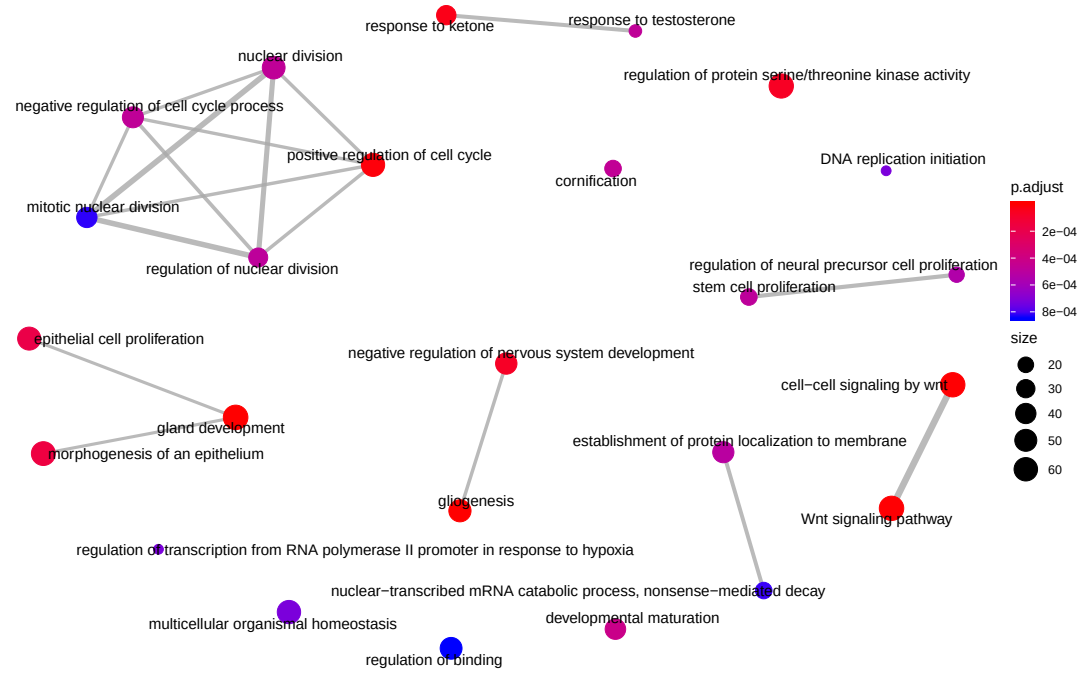


Figure 11: SSLB factor matrix where each row corresponds to a patient and each column corresponds to a bicluster. A patient belongs to a bicluster if they have a non-zero value in that column. Rows are ordered by clinical ER status; within ER status, rows are ordered by factor values in biclusters 1 and 2. All 30 biclusters found by SSLB are shown.



(a) Enrichment map for genes up-regulated in ER-negative patients.



(b) Enrichment map for genes up-regulated in HER2+ patients.

Figure 12: Breast cancer data: enrichment maps for SSLB genes (a) up-regulated in ER-negative patients, and (b) up-regulated in HER2+ patients. Nodes represent biological processes; size of node reflects number of genes in process which were found by the method. Edges connect genes that are active in different biological processes.

3.7.5. Processing Zeisel Dataset

Here, we describe how we processed the data in Section 3.5. We followed the same pipeline as Z15 but provide the details here for completeness.

Many RNA-seq studies normalize the raw count data to the unit RPKM (Reads Per Kilobase of transcript per Million mapped reads), which accounts for longer genes having more transcripts mapped to them simply due to their length (and not meaningful biological variability). This was unnecessary for this dataset as only the 5' end of each RNA was sequenced and thus the read number was not proportional to gene length (Islam et al., 2014). Additionally, many single-cell RNA-seq studies account for differing cell sizes as larger cells have more RNA. However, this normalization was not done for this dataset as such information is informative in clustering different cell types.

The scRNA-seq data is provided by Z15 at `http://linnarssonlab.org/cortex` and consists of molecule counts for 19,972 genes in 3005 individual cells.

Following Z15, we:

1. Removed all genes that have less than 25 molecules in total over all cells
2. Calculated correlation matrix over the genes and define a threshold as 90th percentile of this matrix ($\rho = 0.2091$). Removed all genes which have less than 5 other genes which correlate more than this threshold.

The next step of data processing was to identify the noisiest genes. Assuming that most of the variability of the genes across the cells can be attributed to the underlying biological processes, these genes are the ones which are most informative for clustering of cells. The strategy of Z15 was to search for genes whose noise - measured by coefficient of variation (CV, standard deviation divided by mean) - was high compared to a Poisson distribution with inflated CV. The rationale for this was outlined in Islam et al. (2014) which used the same single-cell RNA-seq protocol as Z15 but for mouse embryonic stem cells. First, Islam

et al. (2014) noted that the technical noise distribution of ERCC (External RNA Controls Consortium) spike-in molecules (which have no biological variability) followed that of a Poisson, but its CV was inflated by constant factor. The CVs of endogenous genes were inflated above those of the ERCCs, suggesting that this variation is driven by biological factors rather than the variation induced by loss of transcripts in cDNA synthesis.

Z15 implemented the same procedure to identify genes with the greatest biological variability. We followed this procedure: for the genes remaining after the aforementioned data cleaning steps, the mean and CV was calculated. The noise model

$$\log_2(CV) = \log_2(mean^{\alpha} + k)$$

was fit using the software `ceftools`[6]. The best fit was found to be $\alpha = -0.55$ and $k = 0.64$. Next all genes were ranked by their distance from the fit line and the top 5000 genes with the largest distance were selected as informative for further clustering.

Finally, we normalized the gene counts using quantile normalization (using the R package `preprocessCore` (Bolstad, 2018)). Note we used the commonly used "average distribution" as the reference distribution to which to project the quantiles of the raw gene expression levels. The average distribution is obtained by taking the average of each quantile across the samples (Bolstad et al., 2003).

3.7.6. Supplementary Figures for Zeisel Dataset

Here, we provide supplementary figures for the analysis of the mouse single-cell RNA sequencing dataset in Section 3.5. Enrichment maps (Figures 14 and 15) were created using the R package `enrichplot` (Yu, 2018) and display the top 30 biological processes (with lowest FDR q-values satisfying threshold of 0.05) found in the gene ontology enrichment analysis as described in Section 3.5.1.

[6] `https://github.com/linnarsson-lab/ceftools`

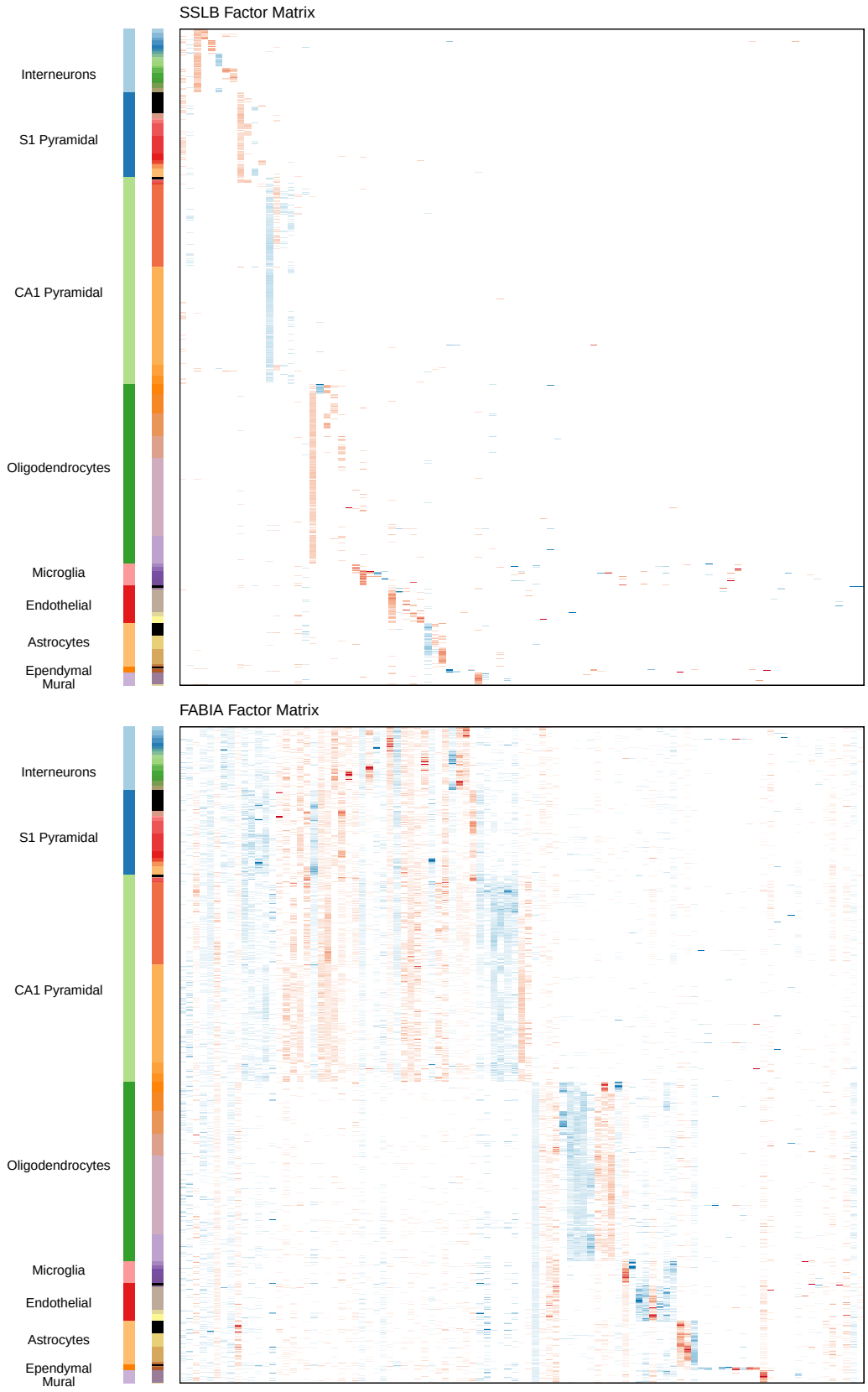
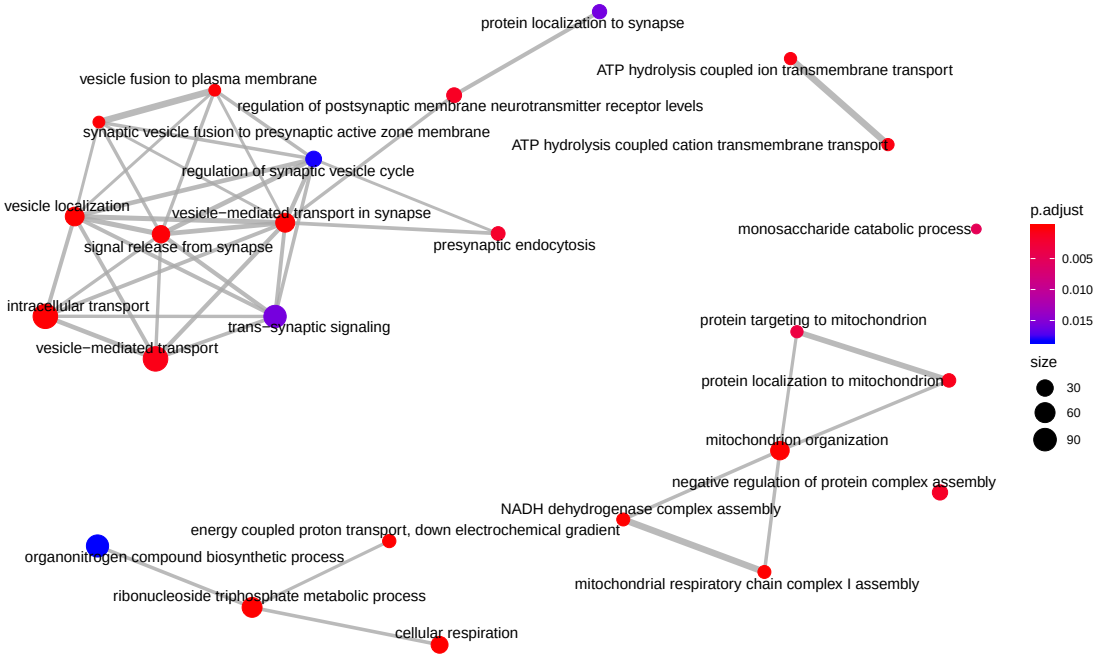
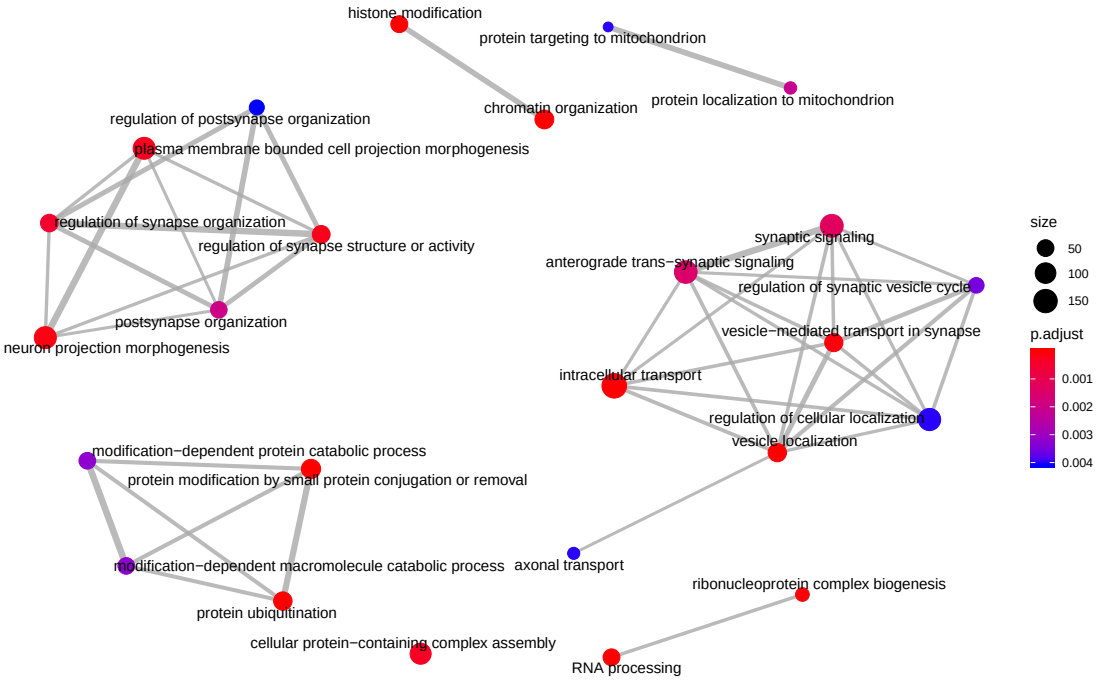


Figure 13: Zeisel dataset: Factor matrix found by SSLB (top) and FABIA (bottom). On the side of the factor matrix are the cell types and subtypes found by Z15, respectively. The rows of the factor matrices have been ordered to correspond to the Zeisel cell types. Factor values have been capped for improved visualization.



(a) SSLB Bicluster 1



(b) SSLB Bicluster 2

Figure 14: Zeisel dataset: enrichment maps for SSLB genes in (a) bicluster 1 and (b) bicluster 2. Each bicluster contains a mixture of interneurons, S1 pyramidal neurons and CA1 pyramidal neurons. Nodes represent biological processes; size of node reflects number of genes in process which were found by the method. Edges connect genes that are active in different biological processes.

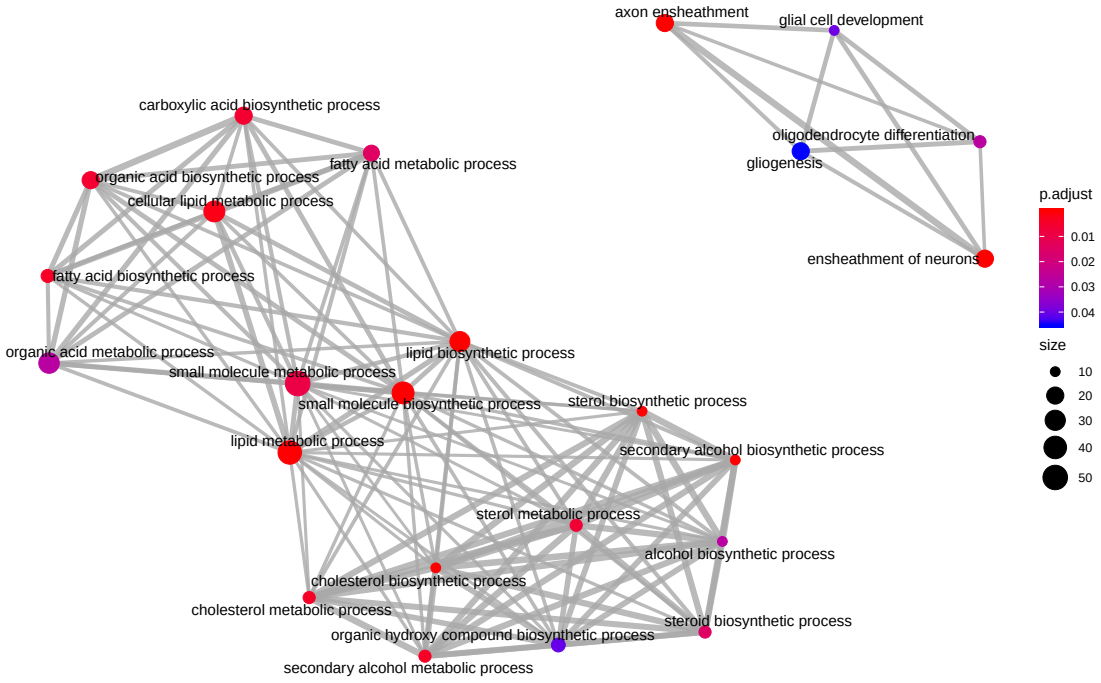


Figure 15: Zeisel dataset: enrichment map for genes in SSLB bicluster 44. Bicluster 44 contains 17 oligodendrocyte cells. Nodes represent biological processes; size of node reflects number of genes in process which were found by the method. Edges connect genes that are active in different biological processes.

CHAPTER 4 : Nonlinear Factor Analysis via BART

4.1. Introduction

Suppose we have observed a matrix of data $\mathbf{Y} \in \mathbb{R}^{N \times G}$ where each row corresponds to a sample and each column corresponds to a feature. That is, for sample i, we observe a vector of features $\mathbf{y}_i \in \mathbb{R}^G$, for $i = 1, \ldots, N$. In many applications, the number of features, G, is very large, but these features are assumed to be driven by a much lower-dimensional latent factor. Classical factor analysis finds such a low-dimensional representation of each $\mathbf{y}_i \in \mathbb{R}^G$, denoted by $\mathbf{x}_i \in \mathbb{R}^K$, with $K \ll G$. Typically, factor analysis methods assume that observation $\mathbf{y}_i$ and factors $\mathbf{x}_i$ are linearly related via a common loadings matrix $\mathbf{B} \in \mathbb{R}^{G \times K}$. That is,

$$\mathbf{y}_i = \mathbf{B}\mathbf{x}_i + \boldsymbol{\varepsilon}_i, \quad i = 1, \ldots, N, \tag{4.1}$$

where $\boldsymbol{\varepsilon} \sim N_G(\mathbf{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = \text{diag}\{\sigma_1^2, \ldots, \sigma_G^2\}$. Unlike the usual regression setup, in factor analysis both the loadings $\mathbf{B}$ and the factors $\mathbf{x}_i$ are unknown.

The assumption that $\mathbf{y}_i$ is linearly related to $\mathbf{x}_i$ may sometimes be too restrictive. Instead, it may be that the mean of $\mathbf{y}_i$ lies on a much lower dimensional manifold, but not necessarily a linear one. In this case, we simply assume $\mathbf{y}_i$ and $\mathbf{x}_i$ are related via a potentially non-linear mapping $f : \mathbb{R}^K \to \mathbb{R}^G$ as

$$\mathbf{y}_i = f(\mathbf{x}_i) + \boldsymbol{\varepsilon}_i, \quad i = 1, \ldots, N \tag{4.2}$$

where $\boldsymbol{\varepsilon}_i \sim N_G(\mathbf{0}, \boldsymbol{\Sigma})$ with $\boldsymbol{\Sigma} = \text{diag}\{\sigma_j^2\}_{j=1}^G$.

The problem we consider in this chapter is two-fold: (i) to find the low-dimensional factors $\mathbf{x}_i$, and (ii) to find the mapping f from the factors $\mathbf{x}_i$ to the features $\mathbf{y}_i$. To accomplish this task, we develop an MCMC algorithm which alternates between sampling from the posteriors of $\mathbf{x}_i$ and a functional approximation to f. This latter step utilizes Bayesian Additive

Regression Trees (BART), introduced by Chipman et al. (2010) (hereafter CGM10). We refer to our method as factor analysis BART (faBART).

BART is a method for nonparametric regression which uses a sum-of-trees model to estimate a broad class of functions. BART has shown tremendous performance in a variety of prediction tasks and has become particularly popular for estimating heterogeneous average treatment effects in causal inference (Hill, 2011). More recently, theoretical support for BART has also emerged (Ročková and van der Pas, 2017; Ročková and Saha, 2018). There have been a number of extensions to the original model including Heteroscedastic BART, in which samples can have different variances, and Monotonic BART, for estimation of monotonically increasing or decreasing functions, to name a few. Up until now, however, all methods for BART have assumed a known set of covariates $\mathbf{x}_i$. Here, we estimate both the mapping f and the unobserved factors $\mathbf{x}_i$.

Nonlinear factor analysis may be viewed as a nonlinear dimensionality reduction method that has a specific likelihood, given in (4.2). Our framework confers a number of benefits over traditional methods for nonlinear dimensionality reduction. Firstly, as we develop an MCMC algorithm to obtain samples from the posterior, we naturally obtain uncertainty quantification for our parameters of interest. Secondly, faBART specifies a generative model and so can predict out-of-sample responses, unlike many other dimensionality reduction methods which are not model based. Thirdly, BART has been extended to also conduct variable selection on the covariates (Linero, 2018). While this extension is for observed covariates, a similar strategy may potentially be used here to determine the dimensionality of the latent space, K. For many dimensionality reduction methods, the choosing of K remains an open area of research. Finally, the default parameter settings of BART yield excellent performance on a wide range of data, negating the need for extensive hyperparameter tuning as often required for neural networks.

The chapter is structured as follows. In Section 4.2, we provide a review of the original BART model and algorithm. In Section 4.3, we review a number of methods for nonlin-

ear dimensionality reduction. In Section 4.4, we describe the faBART model and MCMC algorithm. In Section 4.6, we consider three simulated examples where the true model is known and highlight the ability of faBART to recover this structure. In Section 4.7, we consider two canonical datasets from the dimensionality reduction literature, demonstrating the performance of faBART in effectively visualizing complex, high-dimensional data.

4.2. Review of BART

In this section, we describe the original BART model and algorithm of CGM10. BART assumes the univariate response Y is modeled by:

$$Y = f(\mathbf{x}) + \varepsilon \quad \varepsilon \sim N(0, \sigma^2), \tag{4.3}$$

where f is an unknown function and $\mathbf{x} = (x_1, \ldots, x_p)^T \in \mathbb{R}^N$ are observed covariates. To approximate the function f, CGM10 use a sum of regression trees:

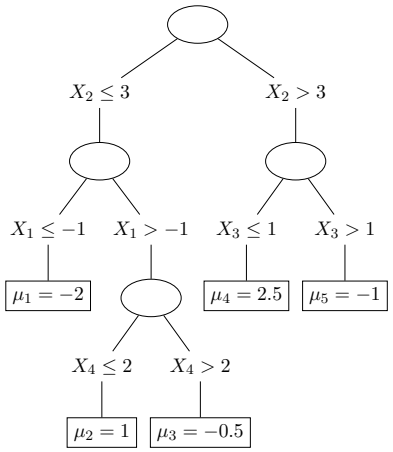
$$f(\mathbf{x}) \approx h(\mathbf{x}) = \sum_{l=1}^{L} g_l(\mathbf{x}; T_l, M_l), \tag{4.4}$$

where $g : \mathbb{R}^p \to \mathbb{R}$ and T_l is a binary regression tree which consists of interior nodes, each with a decision rule, and terminal nodes. Each of the B terminal nodes of T_l are assigned a parameter μ_{lb}; these parameters are collected in the set $M_l = \{\mu_{l1}, \ldots, \mu_{lB}\}$. The decision rules which make up the interior of T_j are generally based on a single covariate and are of the form $\{x_j \leq c\}$ vs $\{x_j > c\}$ for some cut-point $c \in \mathbb{R}$ (Figure 16a). Although each decision rule is based on a single covariate, the tree itself can include decision rules for multiple covariates and so can accommodate interactions. For an observation Y_i with covariates $\mathbf{x}_i$, the function $g_l(\mathbf{x}_i, T_l, M_l)$ finds the terminal node b associated with $\mathbf{x}_i$ (after passing through the decision nodes) and outputs the parameter μ_{lb} of that terminal node. The sum-of-trees function $h(\mathbf{x}_i)$ is then the sum of these outputs over the different trees, T_l, and parameter

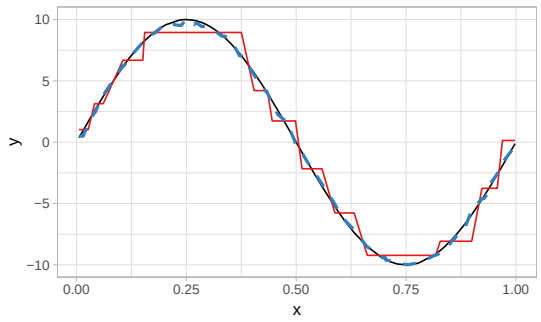
sets, M_l:

$$h(\mathbf{x}_i) = \sum_{l=1}^{L} \mu_{lb(i,l)} \tag{4.5}$$

where $b(i,l)$ is the terminal node associated with $\mathbf{x}_i$ for tree T_l. This sum-of-trees model gives BART the flexibility to model complex response surfaces without knowledge of the form of the surface. Figure 16b displays the model $y = 10\sin(2\pi x)$ (black line) and the BART fit using only one tree (red line) and then 200 trees (blue dashed line).



(a) Decision tree.



(b) BART example when both Y and $\mathbf{x}$ are observed. Black line is the true model is $y = 10\sin(2\pi x)$. Red line is BART fit with one tree. Blue dashed line is BART fit from 200 trees.

Figure 16: Illustrations for BART

4.2.1. Prior Specification

CGM10 specified the following generative model for the trees and node parameters. First, a tree, T_l is drawn from the prior $p(T_l)$, for $l = 1, \ldots, L$. Conditioned on this tree, the terminal node parameters $\{\mu_{lb}\}_{b=1}^{B_l}$ are then drawn independently from the prior $p(\mu_{lb}|T_l)$. The flexibility of BART is a result of regularization at two levels of this prior hierarchy. Firstly, the prior on the trees, $p(T_l)$, encourages the trees to have few terminal nodes. Secondly, the prior on the terminal node parameters, $p(\mu_{lb}|T_l)$ encourages each of the μ_{lb} to be on the order of $1/\sqrt{L}$, where L is the total number of trees. Hence, when L is large, the contribution of each μ_{lb} to the fit of Y is very small.

We now outline these priors in more detail. The prior on the trees, $p(T_l)$, is specified implicitly by a stochastic process rather than an explicit closed form expression (Chipman et al., 1998). This process generates a tree T_l as follows:

1. T_l is set to be a tree with a single terminal node, denoted by η.

2. The terminal node η is split to create two nodes with probability:

$$p_{SPLIT}(\eta, T_l) = \alpha(1 + d_\eta)^{-\beta}, \tag{4.6}$$

   where d_η is the depth of node η. Else, η remains a terminal node. CGM10 suggest $\alpha = 0.95$ and $\beta = 2$ as default values; this places high probability on deep nodes not splitting (i.e. remaining terminal), thus providing the aforementioned regularization on tree size.

3. If the node η is split, it is assigned a decision rule ρ according to the distribution $p_{RULE}(\rho|\eta, T_l)$, and the left and right children nodes of η are then created. A decision rule ρ consists of a predictor, x_j, and a cutpoint c. CGM10 take $p_{RULE}(\rho|\eta, T_l)$ to be the distribution obtained from choosing x_j uniformly over the set of available predictors, and then choosing the cutpoint c uniformly from the observed values of x_j.

4. The process is repeated for the new children nodes of η until no further nodes are split.

For the terminal node parameters, the prior is given by:

$$\mu_{lb} \sim N(0, \sigma_\mu^2) \quad \text{where } \sigma_\mu = 0.5/2\sqrt{L}, \tag{4.7}$$

when the responses have been scaled to be in the range $[-0.5, 0.5]$. As discussed earlier, this prior results in strong regularization on the magnitude of μ_{lb} when the number of trees L is large.

To complete the prior specification, the residual variance is taken to be independent of the trees and node parameters *a priori* and is assigned an inverse-χ^2 prior, which is calibrated to the observed variation in the responses.

4.2.2. BART MCMC Algorithm

CGM10 develop a backfitting MCMC algorithm to obtain posterior estimates of the trees T_l, terminal node parameters, M_l, and residual variance, σ^2. The backbone of the algorithm is a Gibbs sampler which alternates sampling from the following conditional distributions:

$$(T_l, M_l)|T_{(l)}, M_{(l)}, \sigma, y, \quad l = 1, \ldots, L \tag{4.8}$$

$$\sigma|T_1, \ldots, T_l, M_1, \ldots, M_l, y, \tag{4.9}$$

where $T_{(l)}$ denotes the set of all trees, excluding the lth tree, and $M_{(l)}$ is similarly defined.

Within this Gibbs sampler, CGM10 use a Metropolis-Hastings step to draw from the full conditional distribution of (T_l, M_l). The "backfitting" designation of the algorithm comes from the observation that the conditional distribution $p(T_{(l)}, M_{(l)}|T_{(l)}, M_{(l)}, \sigma, y)$ depends only upon $(T_{(l)}, M_{(l)}, y)$ through the vector of partial residuals, R_l, defined by:

$$R_l \equiv y - \sum_{k \neq l} g(\mathbf{x}; T_k, M_k). \tag{4.10}$$

Then, CGM10 draw from (4.8) in two steps:

$$T_l|R_l, \sigma, \tag{4.11}$$

$$M_l|T_l, R_l, \sigma. \tag{4.12}$$

CGM10 obtain a draw of T_l in (4.11) using a Metropolis-Hastings sampling scheme. This scheme proceeds by first drawing a tree proposal, T_l^*, by performing one of four possible operations on the previous tree, T_l. These operations are: (i) growing two new children nodes, (ii) collapsing two nodes, (iii) changing a splitting rule for a node; or (iv) swapping

a parent and a child's splitting rule. The proposal T_l^* is then either accepted or rejected based on the usual Metropolis-Hastings acceptance ratio. Next, a draw from (4.12) is a set of independent draws from a normal distribution, due to the conjugacy of the prior. With the draw (4.8) completed, a draw of σ in (4.9) is simply a draw from an inverse-χ^2 distribution.

4.3. Related Work

We now provide a review of a number of related nonlinear dimensionality reduction methods.

4.3.1. Kernel PCA

Principal components analysis (PCA) is a standard method for linear dimensionality reduction with widespread use across many disciplines of science and engineering. For a data matrix $\mathbf{Y} \in \mathbb{R}^{N \times G}$ centered to have column means zero, PCA computes the following decomposition:

$$\mathbf{y}_i = \mathbf{W}\mathbf{x}_i, \quad i = 1, \ldots, N, \tag{4.13}$$

where $\mathbf{W} \in \mathbb{R}^{G \times K}$ is the matrix consisting of the first K eigenvectors of $\mathbf{Y}^T\mathbf{Y}$ (in decreasing order by the magnitude of associated eigenvalues) and $\mathbf{x}_i \in \mathbb{R}^K$ is called the vector of scores. In this way, PCA is very similar to the traditional factor analysis model (4.1); indeed, Tipping and Bishop (1999) showed that the maximum likelihood estimator for $\mathbf{B}$ in (4.1) converges to the PCA loadings matrix $\mathbf{W}$ in the limit when the column variances $\sigma_j^2 \to 0$ for $j = 1, \ldots, G$. PCA also features the following appealing interpretation: it finds the coordinate axis system to which the data points are most closely aligned (in Euclidean distance).

A limitation of PCA, however, is that it only finds a low-dimensional representation of the data, $\mathbf{x}_i$, that is linearly related to the observed data $\mathbf{y}_i$. As such, PCA cannot find non-linear low-dimensional structures that are embedded in the high-dimensional observed

data. To overcome this limitation, Schölkopf et al. (1997) developed Kernel PCA. Kernel PCA begins with the specification of a kernel matrix, $\mathbf{K} \in \mathbb{R}^{N \times N}$, where

$$\mathbf{K}_{ij} = \Phi(\mathbf{y}_i)^T \Phi(\mathbf{y}_j), \quad 1 \leq i, j \leq N, \tag{4.14}$$

for a user-specified function $\Phi : \mathbb{R}^G \to \mathbb{R}^M$. Then, instead of finding the eigenvectors of $\mathbf{X}^T\mathbf{X}$, kernel PCA finds the eigenvectors of this kernel matrix.

4.3.2. Gaussian process latent variable models

Lawrence (2005) introduced the Gaussian process latent variable model (GP-LVM). GP-LVMs allow for a non-linear mapping between the observed features and latent factors though the model

$$\mathbf{y}_i \sim N(f(\mathbf{x}_i), \sigma^2), \quad i = 1, \ldots, N, \tag{4.15}$$

where f is drawn from a Gaussian process with zero mean and covariance function $k(\cdot, \cdot)$. That is, for a set of $\mathbf{X} = \{\mathbf{x}_1^T, \ldots, \mathbf{x}_N^T\}$, a realization from f is a realization of a multivariate Gaussian with mean zero and covariance matrix $\mathbf{K} \in \mathbb{R}^{N \times N}$ with (i, j)th entry equal to $k(\mathbf{x}_i, \mathbf{x}_j)$. Similarly to Kernel PCA, GP-LVMs require the specification of a kernel function $k(\cdot, \cdot)$. A difference, however, is that GP-LVMs specify a kernel over the latent factors, $\{\mathbf{x}_i\}_{i=1}^N$, unlike Kernel PCA where the kernel defined for the observed data, $\{\mathbf{y}_i\}_{i=1}^N$. The latent factors are then estimated via a scaled conjugate gradient algorithm.

4.3.3. Variational Autoencoders

Variational autoencoders (VAE, Kingma and Welling, 2013) assume $\mathbf{y}_i$ is related to the latent factors $\mathbf{x}_i$ in the following model:

$$\mathbf{y}_i = f_\mu(\mathbf{x}_i; \boldsymbol{\theta}) + f_{\sigma^2}(\mathbf{x}_i; \boldsymbol{\theta})\boldsymbol{\varepsilon}_i, \quad \boldsymbol{\varepsilon}_i \sim N(0, \mathbf{I}) \tag{4.16}$$

where $f_\mu : \mathbb{R}^K \to R^G$ and $f_{\sigma^2} : \mathbb{R}^K \to \mathbb{R}^G$ are multi-layer perceptrons, parameterized by $\boldsymbol{\theta}$. A multi-layer perceptron (MLP) is a fully connected neural network with one hidden layer; in particular Kingma and Welling (2013) define:

$$f_\mu(\mathbf{x}_i; \boldsymbol{\theta}) = \mathbf{W}_2\mathbf{h} + \mathbf{b}_2 \tag{4.17}$$

$$\log f_\sigma(\mathbf{x}_i; \theta) = \mathbf{W}_3\mathbf{h} + \mathbf{b}_3 \tag{4.18}$$

$$\mathbf{h} = \tanh(\mathbf{W}_1\mathbf{x}_i + \mathbf{b}_1), \tag{4.19}$$

where $\mathbf{h}$ is the hidden layer and $\boldsymbol{\theta} = \{\mathbf{W}_1, \mathbf{W}_2, \mathbf{W}_3, \mathbf{b}_1, \mathbf{b}_2, \mathbf{b}_3\}$ are the parameters to be optimized. Note that the mean and variance functions f_μ and f_{σ^2}, respectively, share the final layer of the network, given in (4.19). While each datapoint $\mathbf{y}_i$ has an individual latent vector $\mathbf{x}_i$, the weight parameters, $\boldsymbol{\theta}$, of the neural network are shared across data points.

The goal is to find the posterior distribution of $\mathbf{X}$; unfortunately, this is intractable. Kingma and Welling (2013) resolve this problem by approximating $p(\mathbf{x}_i|\mathbf{y}_i)$ by the variational density:

$$q(\mathbf{x}_i; \mathbf{y}_i, \phi) \sim N(g_\mu(\mathbf{y}_i; \boldsymbol{\phi}), g_{\sigma^2}(\mathbf{y}_i; \boldsymbol{\phi})\mathbf{I}), \tag{4.20}$$

where $g_\mu : \mathbb{R}^G \to \mathbb{R}^K$, $g_{\sigma^2} : \mathbb{R}^G \to \mathbb{R}^K$ are again MLPs, defined similarly to 4.19 and parameterized by $\boldsymbol{\phi}$. Kingma and Welling (2013) then minimize the Kullback-Leibler divergence between $q(\mathbf{x}_i; \mathbf{y}_i, \phi)$ and $p(\mathbf{x}_i|\mathbf{y}_i)$ using stochastic gradient descent. To improve the gradient estimates for this algorithm, they also use a novel re-parameterization technique for the latent factors.

4.3.4. t-SNE

The idea behind t-distributed Stochastic Neighbor Embedding (t-SNE, Maaten and Hinton, 2008) is that "similar" responses $\mathbf{y}_i$ and $\mathbf{y}_j$ should have latent factors $\mathbf{x}_i$ and $\mathbf{x}_j$, respectively, which are also "similar". For the responses, Maaten and Hinton (2008) define a similarity

matrix, $\{\mathbf{P}_{ij}\}_{i,j=1}^{N} \in \mathbb{R}^{N\times N}$, using Gaussian kernels that are re-weighted to correspond to probabilities:

$$\mathbf{P}_{ij} = \frac{p_{i|j} + p_{j|i}}{2}, \tag{4.21}$$

$$\text{where} \quad p_{i|j} = \frac{\exp\{-\|\mathbf{y}_j - \mathbf{y}_i\|^2/2\sigma_j^2\}}{\sum_{l\neq j}\exp\{-\|\mathbf{y}_j - \mathbf{y}_l\|^2/2\sigma_j^2\}}. \tag{4.22}$$

For the unobserved factors, a similarity matrix, denoted by $\mathbf{Q}$, is also required. A key contribution of Maaten and Hinton (2008) is that instead of again using Gaussians for $\mathbf{Q}$ as they do for $\mathbf{P}$, they use a t-distribution with one degree of freedom. Using the t-distribution here allows the factors to "spread out" in the latent space. This spreading out ameliorates the crowding problem which informally states that it is impossible to preserve relative distances between high-dimensional points in a much lower dimensional space. In the latent factor space, t-SNE therefore sacrifices representing the global structure of the original data in favor of preserving the local structure. In terms of calculating such a representation: t-SNE finds the factors which minimize the Kullback-Leibler divergence between the response similarity matrix $\mathbf{P}$ and the factor similarity matrix $\mathbf{Q}$.

4.4. Nonlinear Factor Analysis via BART

We now describe our method for nonlinear factor analysis via BART (faBART). The observed data is the matrix, $\mathbf{Y} \in \mathbb{R}^{N\times G}$, where each row corresponds to a sample and each column to a feature. For each $\mathbf{y}_i \in \mathbb{R}^G$, we seek a low-dimensional representation $\mathbf{x}_i \in \mathbb{R}^K$. We model each feature separately; that is, conditional on the unobserved matrix of factors, $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T \in \mathbb{R}^{N\times K}$, the columns of $\mathbf{Y}$ are independent. In other words, we assume that each feature is driven by the same underlying set of factors but that each feature has a different mapping from the latent space to the observed space. Further, we assume that all sources of dependence between the features are due to the latent factors. These assumptions are very similar to the usual linear factor analysis model; the difference here is that we are allow for a flexible, possibly non-linear mapping between $\mathbf{x}_i$ and $\mathbf{y}_i$.

We denote the jth column of $\mathbf{Y}$ by $\mathbf{y}_{\cdot j}$. Then, $\mathbf{y}_{\cdot j}$ is modeled as

$$\mathbf{y}_{\cdot j} = h_j(\mathbf{x}_i) + \boldsymbol{\varepsilon}_j, \quad \text{for } j = 1, \ldots, G, \tag{4.23}$$

where $h_j(\mathbf{x}_i)$ is a BART sum-of-trees model (4.4) for the regression of $\mathbf{y}_{\cdot j}$ on $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_N]^T$, and $\boldsymbol{\varepsilon}_j \sim N(0, \sigma_j^2\ \mathbf{I}_N)$. We use the notation $h_{ij} = h_j(\mathbf{x}_i)$; the matrix $\mathbf{H} = \{h_{ij}\}_{i,j=1}^{N,G}$ then defines the mean of the observed matrix $\mathbf{Y}$ under the model (4.23).

The likelihood is given by

$$L(\mathbf{Y}|\mathbf{X}, \mathbf{H}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{NG/2}|\boldsymbol{\Sigma}|^{N/2}} \exp\left\{ -\sum_{j=1}^{G}\sum_{i=1}^{N} \frac{(y_{ij} - h_{ij})^2}{2\sigma_j^2} \right\}. \tag{4.24}$$

Despite the nonlinear mapping between $\mathbf{X}$ and $\mathbf{Y}$, the likelihood itself is Gaussian and so, given the current BART fit, $\mathbf{H}$, it is easily computable. This tractability allows us to develop a Metropolis algorithm to obtain samples from the posterior of $\mathbf{X}$. We describe this algorithm in more detail in the next section.

For the prior on $\mathbf{X}$, we use an independent normal with variance τ^2:

$$\mathbf{x}_i \overset{ind}{\sim} N(0, \tau^2 \mathbf{I}_K). \tag{4.25}$$

Although linear factor analysis methods generally take $\tau^2 = 1$ for the prior factor variance, we recommend a larger value of τ^2 for a relatively non-informative prior on the factors. As we will see in Section 4.6, a large τ^2 allows for the factors to be more mobile in the MCMC exploration of the latent space. Placing a diffuse inverse-gamma prior on τ^2 resulted in poor mixing of the MCMC algorithm and so we recommend a fixed value.

For the prior on the noise variances, σ_j^2, we take a scaled inverse-χ^2:

$$\sigma_j^2 \sim \text{inv-}\chi^2(\nu, \lambda), \quad j = 1, \ldots, G. \tag{4.26}$$

We calibrate the degrees of freedom, ν, and scale, λ, using an informal empirical Bayes strategy inspired by CGM10. We first take $\nu = 3$, a value which results in a relatively diffuse prior, but not so small as to favor extremely small values of σ_j^2. We denote the sample variance of each of the columns by $\{s_j^2\}_{j=1}^G$. Our strategy assumes that each column of $\mathbf{Y}$ is driven by at least one non-zero factor; that is, every column contains some signal and is not "pure noise". We then calibrate λ based on the smallest s_j^2, which would contain the smallest amount of signal and thus provide a better estimate of the noise. Specifically, we find the 5% quantile of the s_j^2 and then find the value of λ such that this quantile is the 90% quantile of the prior distribution (4.26). Although this strategy essentially assumes that the noise variance for each of the columns is the same, the degrees of freedom, ν, is small enough to place prior mass on larger values of σ_j^2 so as not to unduly constrain the variance.

4.4.1. MCMC Algorithm

The posterior distribution of the factors, $\mathbf{X}$, is given by:

$$p(\mathbf{X}|\mathbf{Y}) = \prod_{i=1}^{N} p(\mathbf{x}_i|\mathbf{y}_i) \tag{4.27}$$

$$= \prod_{i=1}^{N} \frac{1}{|\mathbf{\Sigma}|^{1/2}} \exp\left\{ -\sum_{j=1}^{G} \frac{(y_{ij} - h_{ij})^2}{2\sigma_j^2} - \frac{1}{2\tau^2} \sum_{k=1}^{K} x_{ik}^2 \right\}. \tag{4.28}$$

Given the observed data, $\mathbf{Y}$, we obtain samples from the posterior of the factors using a Metropolis algorithm. For each of the factor rows $\mathbf{x}_i$, we draw a proposal, $\widetilde{\mathbf{x}}_i$ from a spherical Gaussian centered at the previous draw:

$$\widetilde{\mathbf{x}}_i \sim N(\mathbf{x}_i^{(t)}, c^2 \mathbf{I}_N), \tag{4.29}$$

where c^2 is a tuning parameter. We take $c = 2.38/\sqrt{K}$ as recommended by Gelman et al. (1996). We have found this random walk proposal to be effective; in Section 4.9.2 of the Appendix, we also consider an alternate proposal distribution which encourages factors $\mathbf{x}_i$

and $\mathbf{x}_j$ to be close if $\mathbf{y}_i$ and $\mathbf{y}_j$ are similar. However, this proposal yielded similar results to our random walk proposal (4.29).

To accept or reject the proposed factors, we need to evaluate the likelihood (4.24). This requires the posterior mean of the fitted values, $\mathbf{H}$, and variance, $\mathbf{\Sigma}$, from the BART sum-of-trees model, conditioned on the proposed factors $\{\widetilde{\mathbf{x}}_i\}_{i=1}^N$. We estimate each column $\mathbf{h}_{\cdot j}$ separately as in (4.23). The fitted values $\mathbf{h}_{\cdot j}$ are a function of L decision trees, each with a set of terminal node parameters, together denoted by $\{(T_l, M_l)\}_{l=1}^L$ (precise definitions in Section 4.2). As such, drawing posterior samples of the fitted values amounts to drawing samples from the posterior of the trees and terminal node parameters. For this purpose, we use the original BART algorithm outlined in Section 4.2. However, instead of drawing one sample of $\{(T_l, M_l)\}_{l=1}^L$, we draw 100 samples. After removing the first 50 draws as burn-in, we calculate the approximate posterior mean of the fitted values resulting from the remaining 50 draws, denoted by $\widetilde{\mathbf{h}}_{\cdot j}$. We obtain draws of the fitted values using the R package `dbarts` (Dorie et al., 2018) with $L = 50$ trees.

Finally, we calculate the acceptance ratio, α_i, for each of the proposed factors, $\widetilde{\mathbf{x}}_i$. At iteration $t + 1$, this is given by:

$$\alpha_i = \min\left\{1, \frac{L(\mathbf{y}_i|\widetilde{\mathbf{x}}_i, \widetilde{\mathbf{h}}_i, \widetilde{\mathbf{\Sigma}})\pi(\widetilde{\mathbf{x}}_i)}{L(\mathbf{y}_i|\mathbf{x}_i^{(t)}, \mathbf{h}_i^{(t)}, \mathbf{\Sigma}^{(t)})\pi(\mathbf{x}_i^{(t)})}\right\} \tag{4.30}$$

where $\mathbf{x}_i^{(t)}$, $\mathbf{h}_i^{(t)}$ and $\Sigma^{(t)}$ are the draws from the previous iteration. For each $i = 1, \ldots, N$, we then accept $\widetilde{\mathbf{x}}_i$ with probability α_i. We either accept or reject each row of the factor matrix $\mathbf{X}$ separately instead of the entire matrix. Updating the factors row-wise is also common in linear factor analysis methods.

The Metropolis algorithm is displayed in Algorithm 2. In Section 4.9.1 of the Appendix, we also considered an elliptical slice sampler to obtain draws of $\mathbf{X}$. However, the Metropolis algorithm was ultimately more effective. The lack of differentiability in the sum-of-trees function also precludes a Hamiltonian Monte Carlo sampling algorithm; however, the un-

observed factors lie in a much lower dimensional space than the observed data where the Metropolis sampler appears to mix well.

The final output of our MCMC algorithm consists of the discovered factors, $\widehat{\mathbf{X}}$, and the non-parametric estimate of the mapping, $\mathbf{H}$. This is in contrast to ordinary factor analysis where we also obtain estimates of the loadings matrix, $\mathbf{B}$, which characterizes the linear form. In our setting, one may obtain a parametric model for the mapping by data analysis of the relationship between the discovered factors, $\widehat{\mathbf{X}}$, and either the observed data, $\mathbf{Y}$, or the non-parametric estimate of the mapping, $\mathbf{H}$. This strategy is illustrated in the examples of Section 4.6.

Algorithm 2 Metropolis algorithm for faBART

Input: Number of factors, K

Initialize: $\mathbf{x}_i^{(0)} \sim N(0, \mathbf{I}_K)$, fitted values $\mathbf{H}^{(0)}$ based on factors $\{\mathbf{x}_i^{(0)}\}_{i=1}^N$

For $t = 1, \ldots, T$:

1. For $i = 1, \ldots, N$, draw $\widetilde{\mathbf{x}}_i \sim N(\mathbf{x}_i^{(t-1)}, c^2\mathbf{I})$
2. For $j = 1, \ldots, G$:
   - Calculate the fitted values from BART, $\widetilde{\mathbf{h}}_{\cdot j}$, based on the factors $\{\widetilde{\mathbf{x}}_i\}_{i=1}^N$
3. For each $i = 1, \ldots, N$:
   - Calculate acceptance ratio:

$$\alpha_i = \min\left\{1, \frac{L(\mathbf{y}_i|\widetilde{\mathbf{x}}_i, \widetilde{\mathbf{h}}_i, \widetilde{\mathbf{\Sigma}})\pi(\widetilde{\mathbf{x}}_i)}{L(\mathbf{y}_i|\mathbf{x}_i^{(t-1)}, \mathbf{h}_i^{(t-1)}, \mathbf{\Sigma}^{(t-1)})\pi(\mathbf{x}_i^{(t-1)})}\right\}$$

   - Draw $u \sim \mathcal{U}[0,1]$. Set

$$\mathbf{x}_i^{(t)} = \begin{cases} \widetilde{\mathbf{x}}_i & \text{if } u < \alpha_i \\ \mathbf{x}_i^{(t-1)} & \text{otherwise.} \end{cases} \quad \text{and} \quad \mathbf{h}_i^{(t)} = \begin{cases} \widetilde{\mathbf{h}}_i & \text{if } u < \alpha_i \\ \mathbf{h}_i^{(t-1)} & \text{otherwise.} \end{cases}$$

4.5. Identifiability

In this section, we pause the development of faBART to discuss the issue of identifiability in the nonlinear factor analysis model (4.2). It is well known that for linear factor analysis, the estimated factors and loadings are unidentifiable up to a rotation of the factor and loading matrices. That is, the model (4.1) cannot distinguish between the following parameterizations:

$$\mathbf{Y} = \mathbf{X}\mathbf{B}^T + \mathbf{E} = (\mathbf{X}\mathbf{P})(\mathbf{B}\mathbf{P})^T + \mathbf{E} \tag{4.31}$$

where $\mathbf{P} \in \mathbb{R}^{K\times K}$ is a rotation matrix, i.e. $\mathbf{P}\mathbf{P}^T = \mathbf{I}$. To overcome this issue, researchers have proposed a number of different solutions: for example, restricting the loadings matrix $\mathbf{B}$ to be upper-triangular (Aguilar and West, 2000), or orthogonal, or placing sparsity inducing priors on these matrices to restrict the space of matrices under consideration.

As may be expected from its flexibility, nonlinear factor analysis faces a greater identifiability issue. As a simple illustration, suppose the features of the ith sample, denoted by $\mathbf{y}_i = (y_{i1}, \ldots, y_{iG})^T \in \mathbb{R}^G$, are being driven by a single latent factor, $x_i \in \mathbb{R}$, as follows:

$$y_{ij} = f_j(x_i) + \varepsilon_{ij}, \quad j = 1, \ldots, G, \tag{4.32}$$

where $f(x) = (f_1(x), \ldots, f_G(x))^T$ is the true mapping from the latent to observed data, and ε_{ij} is the noise. Then, the model (4.32) is indistinguishable from the parameterization:

$$y_{i1} = \widetilde{x}_i + \varepsilon_{i1}, \tag{4.33}$$

$$y_{ij} = \widetilde{f}_j(\widetilde{x}_i) + \varepsilon_{ij}, \quad j = 2, \ldots, G, \tag{4.34}$$

where $\widetilde{x}_i = f_1(x_i)$ and $\widetilde{f}_j = f_j \circ f_1^{-1}$.

To render the model (4.2) identifiable, Yalcin and Amemiya (2001) suggest the following

parameterization:

$$y_{ij} = x_{ij} + \varepsilon_{ij}, \quad j = 1, \ldots, K, \tag{4.35}$$

$$y_{ij} = f_j(\mathbf{x}_i) + \varepsilon_{ij}, \quad j = K+1, \ldots, G. \tag{4.36}$$

That is, the means of the first K columns of the observed data $\mathbf{Y}$ are constrained to be equal to the (unobserved) factors while the remaining $G-K$ columns may be nonlinear functions of these factors. This strategy serves to "anchor" the factors, allowing the relationship between the factors and the observed data to be identified, relative to the first K columns of $\mathbf{Y}$. More generally, the parameterization (4.36) can be seen as an analogue to the linear factor analysis strategy of restricting the loadings matrix to be upper triangular.

A problem with the model (4.36), however, is that it depends strongly on which columns of $\mathbf{Y}$ are set equal to a factor. For instance, the model (4.36) sets the first and second columns of $\mathbf{Y}$ to be equal to two different factors, when it is possible that these columns are actually driven by the same underlying factor. In the linear factor analysis model, Carvalho et al. (2008) also restrict the loadings matrix to be upper-triangular but allow for uncertainty as to which columns are set equal to a factor in their evolutionary stochastic search strategy.

A similar modeling assumption to (4.36) may prove to be a fruitful strategy for faBART, provided it also allows for uncertainty regarding which columns are set equal to a factor. We leave the development of such a strategy to future work. In this chapter, our focus is to highlight the potential of faBART for nonlinear factor analysis. In the next section, we consider simulated examples in which there is a linear component (in addition to nonlinear) to help "anchor" the faBART algorithm.

4.6. Parametric Examples

4.6.1. Example 1

In this section, we consider a simple example with $N = 100$ samples, $G = 5$ features and latent factor dimension $K = 1$. That is, the signal for all responses in the observed data is being driven by a single factor.

The data is generated as follows. Each element of the factor vector $\mathbf{x} = \{x_i\}_{i=1}^N$ is drawn independently from a Uniform$[-3, 3]$ distribution. The first three columns of $\mathbf{Y}$ are linearly related to $\mathbf{x}$, the fourth column is a function of $\mathbf{x}^2$ and the fifth column is a function of $\sin(\mathbf{x}^2)$. That is, each row of the observed matrix is generated as

$$\mathbf{y}_i = (x_i,\ 2x_i,\ 3x_i,\ 4x_i^2,\ 5\sin(\pi/2 \cdot x_i))^T + \boldsymbol{\varepsilon}_i, \quad i = 1, \ldots, N, \tag{4.37}$$

where $\boldsymbol{\varepsilon}_i$ denotes the ith row of the noise matrix $\mathbf{E} \in \mathbb{R}^{N \times G}$ and is generated as $\boldsymbol{\varepsilon}_i \overset{ind}{\sim} N(\mathbf{0}, \sigma^2 \mathbf{I}_G)$ with $\sigma^2 = 0.5$.

With $\mathbf{Y}$ and the factor dimension $K = 1$ as the only inputs, we ran faBART for 2000 iterations with a burn-in period of 1000 iterations. The prior variance for the factors was set to $\tau^2 = 10$. The estimated factor vector $\widehat{\mathbf{x}}$ was calculated as the mean of the factor samples after burn-in.

Figure 17 displays scatterplots of each of the columns of the observed $\mathbf{Y}$ versus the factor, $\widehat{\mathbf{x}}$, found by faBART. Using the true parametric forms which generated the data (4.37), we proceeded to fit these models to the plots. Of course, the forms would need to be decided upon in real non-simulated applications. However, from simple observation of the scatterplots (Figure 17), a quadratic and a sine curve would be the natural choices to model columns four and five of $\mathbf{Y}$, respectively, even if the true model were not known.

More specifically, we found the fitted models as follows. First, we ran a linear regression

with the first column, $\mathbf{y}_{\cdot 1}$ as the response and $\widehat{\mathbf{x}}$ as the single predictor:

$$\mathbf{y}_{\cdot 1} = \beta \widehat{\mathbf{x}}. \tag{4.38}$$

We then calculated a re-scaled factor, $\widetilde{\widehat{\mathbf{x}}} \leftarrow \beta \widehat{\mathbf{x}}$, such that the slope between $\mathbf{y}_{\cdot 1}$ and this new factor, $\widetilde{\widehat{\mathbf{x}}}$ was equal to one. The reason for this re-scaling step is that the faBART mapping (4.23) is unidentifiable up to a scale change of $\mathbf{x}$. As such, we are only interested in whether faBART can recover the *relative* relationships between the factor and the columns of $\mathbf{Y}$. Re-scaling the factor so that the first plot has a slope of one allows us to better highlight these relationships. We then proceeded to fit a linear regression of each of the columns, $\mathbf{y}_{\cdot 2}, \mathbf{y}_{\cdot 3}$, separately against the re-scaled factor $\widetilde{\widehat{\mathbf{x}}}$. For the columns with a true non-linear relationship, $\mathbf{y}_{\cdot 4}$ and $\mathbf{y}_{\cdot 5}$, we ran a regression against $\widetilde{\widehat{\mathbf{x}}}^2$ and $\sin(\pi/2 \cdot \widetilde{\widehat{\mathbf{x}}})$, respectively. Remarkably, the estimated factor from faBART recovered the relative relationship between the true (unobserved) factors and the observed data matrix, $\mathbf{Y}$. Unlike faBART, linear factor analysis would require more than one latent factor dimension (likely three) to capture this structure of variation in $\mathbf{Y}$.

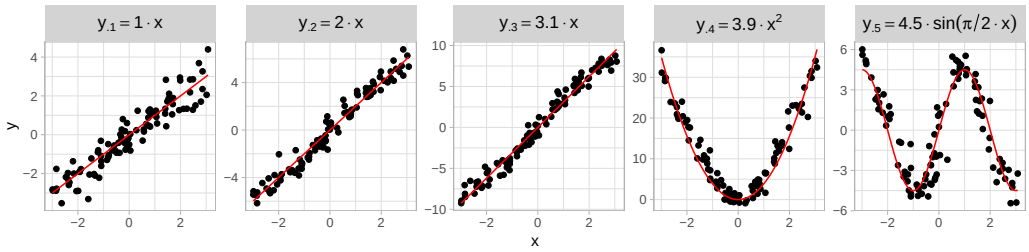


Figure 17: Example 1: Each subplot is a plot of a column $\mathbf{y}_{\cdot j}$ of the observed data vs the factors $\widetilde{\widehat{\mathbf{x}}}$ found by faBART for $j = 1, \ldots, G$. The true models are: (i) $y_{i1} = x_i$; (ii) $y_{i2} = 2x_i$; (iii) $y_{i3} = 3x_i$; (iv) $y_{i4} = 4x_i^2$; and (v) $y_{i5} = 5\sin(\pi/2 \cdot x_i)$. The fitted model (up to scale change of $\mathbf{x}$) are displayed in the subplot titles.

4.6.2. Example 2

In this example, we increase the number of latent factors to $K = 2$. We additionally set the number of samples to $N = 100$ and features to $G = 6$. The two factors are generated

independently from a Uniform$[-3, 3]$ distribution. The feature vectors, $\mathbf{y}_i$, are generated as

$$\mathbf{y}_i = (x_{i1},\ 2x_{i1},\ 3x_{i1}^2,\ 4x_{i2},\ 5x_{i2},\ 6\sin(\pi/2 \cdot x_{i2}))^T + \boldsymbol{\varepsilon}_i, \quad i = 1, \ldots, N, \tag{4.39}$$

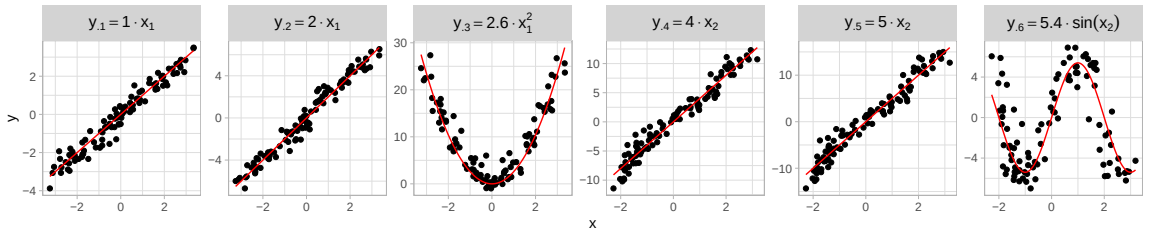
where $\boldsymbol{\varepsilon}_i$ denotes the ith row of the noise matrix $\mathbf{E} \in \mathbb{R}^{N \times G}$ and is generated as $\boldsymbol{\varepsilon}_i \overset{ind}{\sim} N(\mathbf{0}, \sigma^2 \mathbf{I}_G)$ with $\sigma = 0.5$. That is, there are two factors driving the variation in $\mathbf{Y}$, each of which have a nonlinear component.

We ran faBART for 2000 iterations with a burn-in period of 1500 iterations. The prior variance for the factors was set to $\tau^2 = 10$. As a comparison, we also considered the performance of a variational autoencoder (Kingma and Welling, 2013) on this data. Specifically, we implemented a Gaussian variational autoencoder with one hidden layer (consisting of 5 latent variables) and a rectified linear unit (ReLU) for the activation function. For this variational autoencoder, we additionally set the error variance to the true value: $\boldsymbol{\Sigma} = 0.5^2 \mathbf{I}_G$. For both faBART and the variational autoencoder, we set the number of factors to the truth, $K = 2$.

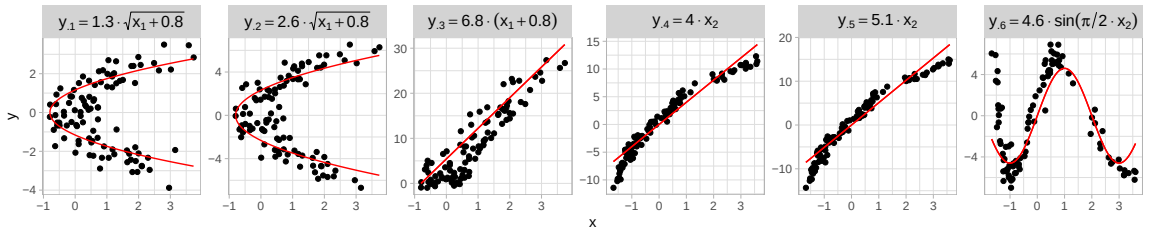
For both faBART and the variational autoencoder, we constructed scatterplots of the observed columns versus the estimated factors, as detailed in the previous section. Again, faBART correctly found the true underlying structure between the observed data and the factors (Figure 18a). The variational autoencoder recovered the second factor, albeit with some curvature in the mapping from the factor to the fourth and fifth columns of $\mathbf{Y}$ (Figure 18b). Further, the variational autoencoder found a factor $\widetilde{\mathbf{x}}_{\cdot 1}$ which appears to be a quadratic of the true factor: i.e. $\widetilde{\mathbf{x}}_{\cdot 1} = \mathbf{x}_{\cdot 1}^2$. As such, the variational autoencoder found an inverse quadratic relationship between the first factor and columns one and two of $\mathbf{Y}$, and a linear relationship between the first factor and the third column of $\mathbf{Y}$.

4.6.3. Example 3

Our final simulated parametric example extends the previous example to include an interaction term between the factors. Specifically, we have $N = 100$ samples, $G = 7$ features



(a) faBART results.



(b) Variational autoencoder results.

Figure 18: Example 2: Each subplot is a plot of a column $\mathbf{y}_{\cdot j}$ of the observed data vs the factors found by each method for $j = 1, \ldots, G$. The true models are: (i) $y_{i1} = x_{i1}$; (ii) $y_{i2} = 2x_{i1}$; (iii) $y_{i3} = 3x_{i1}^2$; (iv) $y_{i4} = 4x_{i2}$; (v) $y_{i5} = 5x_{i2}$; and (vi) $y_{i6} = 6\sin(\pi/2 \cdot x_{i2})$. The fitted model (up to scale change of each column of $\mathbf{X}$) are displayed in the subplot titles.

and $K = 2$ factors. The data is generated as follows. The elements of the factor matrix, $\mathbf{X}$, are each drawn independently from a Uniform$[-3, 3]$ distribution. Each row of the data matrix $\mathbf{Y}$ is generated as:

$$\mathbf{y}_i = (x_{i1},\ 2x_{i1},\ 3x_{i1}^2,\ 4x_{i2},\ 5x_{i2},\ 6\sin(\pi/2 \cdot x_{i2}),\ 7x_{i1} \cdot x_{i2})^T + \boldsymbol{\varepsilon}_i \tag{4.40}$$

where $\boldsymbol{\varepsilon}_i$ denotes the ith row of the noise matrix $\mathbf{E} \in \mathbb{R}^{N \times G}$ and is generated as $\boldsymbol{\varepsilon}_i \overset{ind}{\sim} N(\mathbf{0}, \sigma^2 \mathbf{I}_G)$ with $\sigma = 0.5$. That is, the first three columns of $\mathbf{Y}$ are functions of the first factor, $\mathbf{x}_{\cdot 1}$, the next three columns of $\mathbf{Y}$ are functions of the second factor, $\mathbf{x}_{\cdot 2}$, and the final column is an interaction of both factors.

With $\mathbf{Y}$ and the factor dimension $K = 2$ as the only inputs, we ran faBART for 2000 iterations with a burn-in period of 1000 iterations. The prior variance for the factors was set to $\tau^2 = 10$. The estimated factor matrix, $\widehat{\mathbf{X}}$, was calculated as the mean of the factor samples after burn-in. We again considered the performance of a Gaussian variational

autoencoder with one hidden layer consisting of 6 latent variables; we found this number of latent variables to have the best performance from the set of choices: $\{3, 4, 5, 6, 7\}$. For the activation functions, we used a ReLU function. Additionally, for the variational autoencoder, we set the error variance to the true value: $\mathbf{\Sigma} = 0.5^2\mathbf{I}_G$.

Again, faBART was able to recover both the latent factors *and* the mapping between the factors and the observed data (Figure 19a). The variational autoencoder was able to recover the second factor and the sine relationship between this factor and the sixth column of $\mathbf{Y}$; however, it was not able to recover the first factor or the interaction relationship in the final column of $\mathbf{Y}$ (Figure 19b). It is certainly possible that with more tuning of the variational autoencoder (or more hidden layers) that it may be able to recover this relationship; however, a benefit of BART (and faBART) is that it requires very little tuning "out of the box" while still maintaining excellent performance.

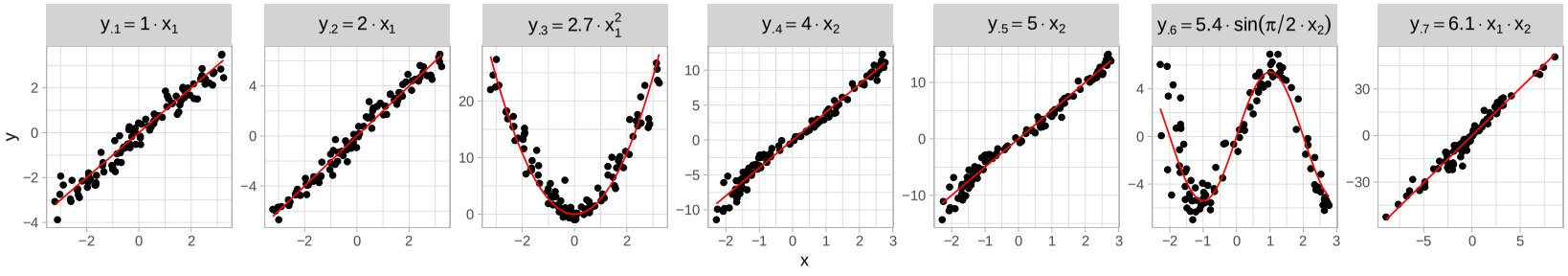
4.7. Visualization Examples

In this section, we consider two canonical datasets used in dimensionality reduction and data visualization: (i) the (simulated) "Swiss-roll" dataset; and (ii) MNIST, a dataset of handwritten digits. These data are highly stylized; for example, the Swiss-roll data contains zero noise. To allow faBART to handle such data, we develop a modified version, called tempered faBART.

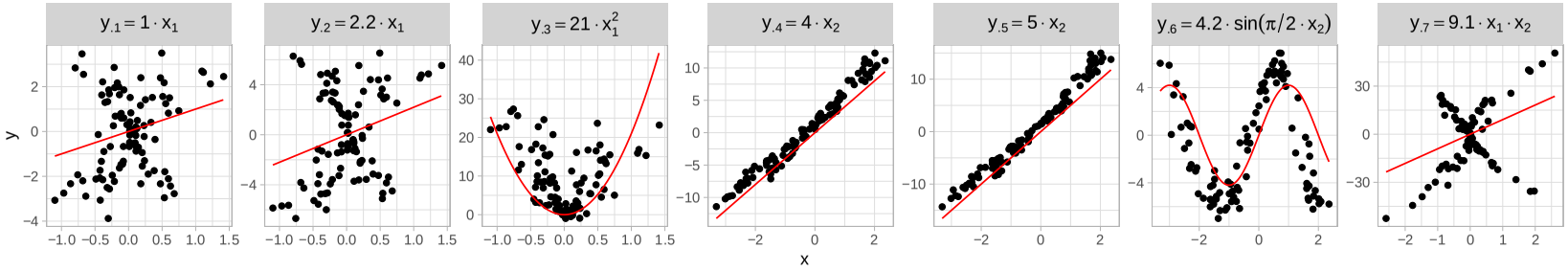
4.7.1. Tempered faBART

The tempered faBART algorithm is very similar to the original Algorithm 1. The only modification is the acceptance ratio: instead of α_i, we use a tempered version, α_i^{temp}, in which we raise the likelihood to the power of a. This tempered acceptance ratio is given by:

$$\alpha_i^{temp} = \exp\left\{ a \sum_{j=1}^{G} \left[\frac{(y_{ij} - h_{ij}^{(t)})^2}{2\sigma_j^{(t)2}} - \frac{(y_{ij} - \widetilde{h}_{ij})^2}{2\sigma_j^2} + \log\left(\frac{\sigma_j^{(t)}}{\widetilde{\sigma}_j} \right) \right] + \sum_{k=1}^{K} \frac{x_{ik}^{(t)2} - \widetilde{x}_{ik}^2}{2\tau^2} \right\}, \tag{4.41}$$



(a) faBART results.



(b) Variational autoencoder results.

Figure 19: Example 3: Each subplot is a plot of a column $\mathbf{y}_{\cdot j}$ of the observed data vs the factors found by VAE for $j = 1, \ldots, G$. The true models are: (i) $y_{i1} = x_{i1}$; (ii) $y_{i2} = 2x_{i1}$; (iii) $y_{i3} = 3x_{i1}^2$; (iv) $y_{i4} = 4x_{i2}$; (v) $y_{i5} = 5x_{i2}$; (vi) $y_{i6} = 6\sin(x_{i2})$; and (vii) $y_{i7} = 7x_{i1} \cdot x_{i2}$. The fitted model (up to scale change of each column of $\mathbf{X}$) are displayed in the subplot titles.

after which we take $\alpha_i^{temp} \leftarrow \min\{1, \alpha_i^{temp}\}$.

Tempering the acceptance ratio has been used to great extent in MCMC sampling (see, for example, Neal, 1996). However, the tempered faBART strategy differs in two ways from many previous methods. Firstly, most tempering methods raise both the likelihood *and* the prior to the power of a, whereas here we only raise the likelihood. Secondly, previous methods generally take $0 \leq a \leq 1$. Instead, we take $a \geq 1$.

The reason tempering methods take $0 \leq a \leq 1$ is to "flatten" the target distribution, helping the algorithm to better navigate highly multimodal posteriors. For the visualization examples we consider here, we instead take $a \geq 1$ to make the landscape more "spiked" to emphasize the structure of interest. This is especially the case for the Swiss-roll data, which contains zero noise.

Further, we raise only the likelihood to the power of a. To see why this is useful, consider again (4.41): having a larger value of a places more weight on the likelihood component, encouraging the MCMC algorithm to accept $\widetilde{\mathbf{x}}_i$ which yield fitted values $\widetilde{\mathbf{h}}_i$ that are very close to $\mathbf{y}_i$.

4.7.2. Swiss-Roll

The "Swiss-roll" dataset is a two-dimensional spiral manifold which lies in three-dimensional space (Figure 20a). Popularized by Tenenbaum et al. (2000) and Roweis and Saul (2000), the goal is to find a two-dimensional representation of the data which "un-wraps" the roll. Here, however, we simply consider the dataset to gain insight into the workings of faBART. Note that we do not promote faBART as necessarily the best method to use for visualization, but it is informative to consider how it performs on such data.

We considered a sample of size $N = 1000$, generated as follows: for $i = 1, \ldots, N$:

1. Draw $u_i \sim U[0,1], v_i \sim U[0,1]$;

2. Compute $\varphi = 3/2\pi(1 + 2u_i)$;

3. Set $\mathbf{y}_i = [\varphi \cos(\varphi), 21v_i, \varphi \sin(\varphi)]^T$.

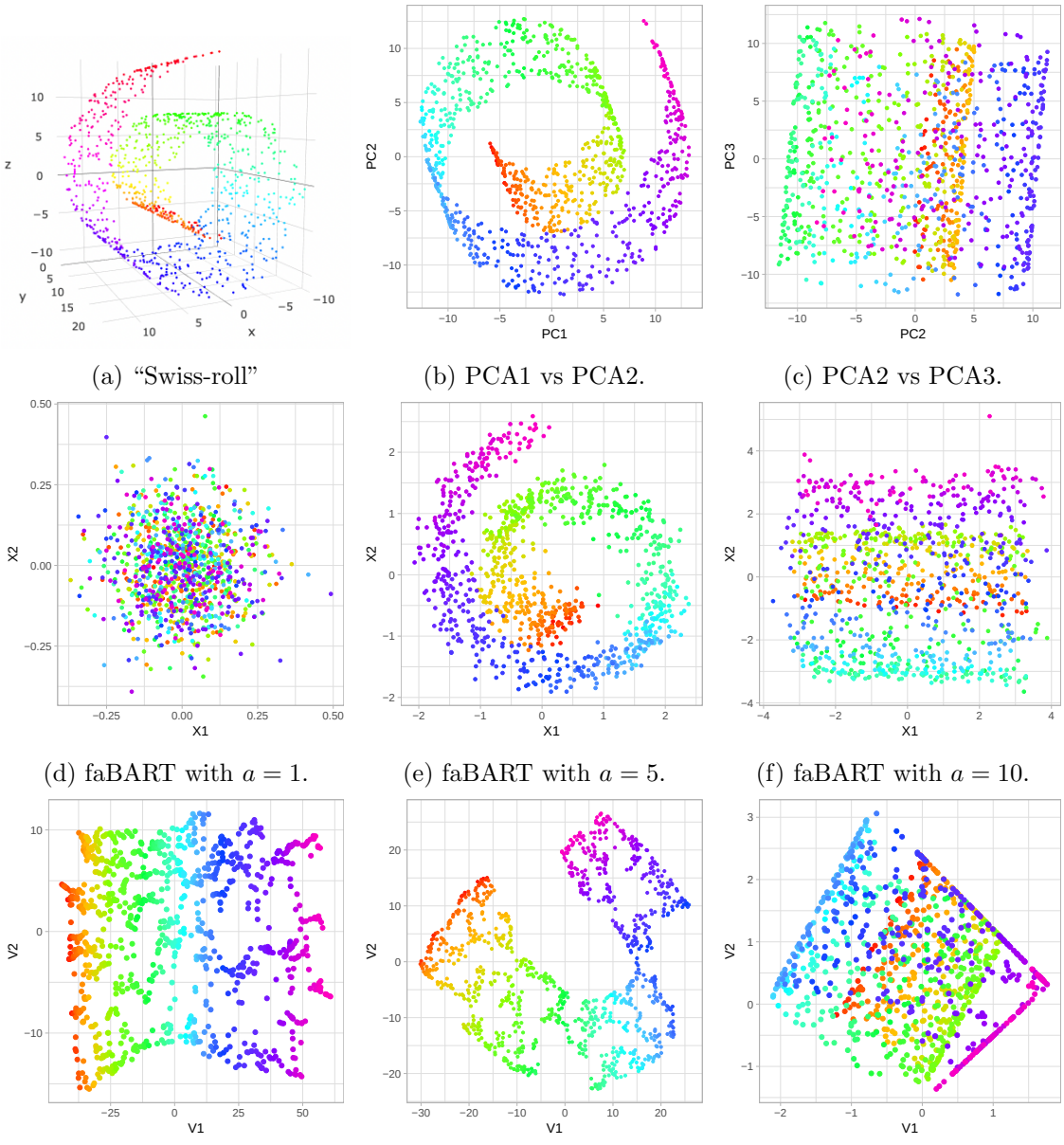
Note that the observed $\mathbf{y}_i$ has zero additive noise; all variation in $\mathbf{y}_i$ is due to the signal.

We first consider the performance of PCA on this dataset. While PCA does not "unwrap" the roll, it still successfully finds meaningful latent structure in the data. The first two principle components represent the variation in the x-z plane, "flattening" the Swiss-roll to a one-dimensional swirl (Figure 20b). The second and third principle components, meanwhile, "flatten" the roll in a direction orthogonal to the swirl (Figure 20c). Essentially, these latent representations are views of the Swiss-roll from two different angles: from the side and from above.

We ran tempered faBART for 1000 iterations with a burn-in of 500 iterations with $K = 2$. The prior variance was set to $\tau^2 = 1$ and we considered three different values of the tempering parameter: (i) $a = 1$, (ii) $a = 5$, and (iii) $a = 10$. When $a = 1$, the factors do not find any meaningful structure (Figure 20d). When we increase $a = 5$, the faBART factors replicate the "swirl" found by PCA (Figure 20e). With a further increase to $a = 10$, the faBART factors "view" the Swiss-roll from above in a manner reminiscent of the second and third principle components (Figure 20f). Thus, the tempering strategy is important to direct faBART towards factors which result in a better reconstruction of $\mathbf{Y}$.

Although faBART allows for a nonlinear mapping between $\mathbf{y}_i$ and $\mathbf{x}_i$, it essentially minimizes a similar objective to PCA: the sum of squared errors between the observed $\mathbf{Y}$ and its reconstruction from the latent factors. As a result, it is not too unsurprising that they both yield similar results on the Swiss-roll. It is important to note, however, that faBART finds such structure with no assumptions on the form of the mapping between $\mathbf{x}_i$ and $\mathbf{y}_i$.

As a further comparison to faBART, we additionally display the latent representations found by Isomap (Tenenbaum et al., 2000), t-SNE and a variational autoencoder (Figures 20g, 20h and 20i, respectively). Isomap was designed to model large geodesic distances and consequently can "unwrap" the Swiss-roll; however, both t-SNE and a variational



(a) "Swiss-roll"

(b) PCA1 vs PCA2.

(c) PCA2 vs PCA3.

(d) faBART with $a = 1$.

(e) faBART with $a = 5$.

(f) faBART with $a = 10$.

(g) Isomap.

(h) t-SNE.

(i) Variational Autoencoder.

Figure 20: Comparison of methods for Swiss-roll dataset.

autoencoder do not. Details for the implementation of each of these methods are provided in Section 4.9.4 of the Appendix.

4.7.3. MNIST

MNIST is a dataset of labeled handwritten digits (LeCun et al., 1998). It is often used as a benchmark for dimensionality reduction techniques, where the goal is to find a two-dimensional latent representation of the data which reflects the true labels of the digits (digit examples shown in Figure 21). An alternate goal for this dataset is to accurately classify the handwritten digits; however, here we simply consider visualization of the data. Each digit is contains 28×28 pixels, yielding a feature dimension of $G = 784$. We consider a randomly drawn subset of the data of size $N = 1000$.

We ran tempered faBART for 1000 iterations with a burn-in period of 500 iterations. The prior variance was set to $\tau^2 = 1$ and the tempering parameter $a = 50$. In the two-dimensional latent factors found by faBART, clear separation of different digits can be observed (Figure 22a). In particular, faBART grouped together the "ones" and "sevens" particularly well, while finding more diffuse groups for other digits, including the "zeros", "twos" and "fours" . Compared to the PCA embedding, faBART appeared slightly better at separating the digits (Figure 22b). Note that the faBART model did not include any clustering component; placing a mixture prior on the factors to better find latent clusters may be interesting future work.

Both t-SNE and the variational autoencoder find latent representations with more separation between the digit classes than faBART (Figures 22c and 22d, respectively). In particular, t-SNE separates almost all the digits into distinct groups (Figure 22c). We reiterate that the goal here is not to promote faBART as the best method for visualization, but to investigate how it performs on such structured data. t-SNE in particular is expressly designed for interpretable data visualization.

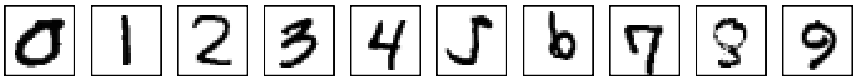
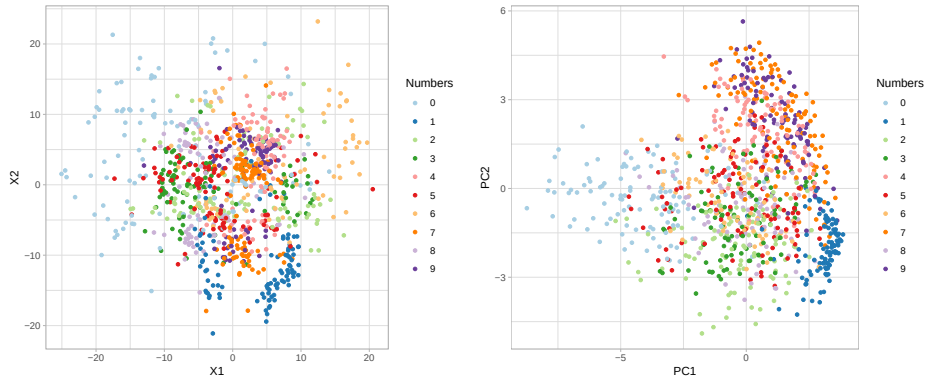


Figure 21: A sample of digits labelled 0 to 9 in the MNIST dataset. Each digit is a 28 × 28 matrix of values between 0 (white) and 255 (black).



(a) faBART (b) PCA

(c) t-SNE (d) Variational Autoencoder

Figure 22: Two-dimensional latent representations of the MNIST data found by each method.

4.8. Conclusion

In this chapter, we developed faBART, a method for nonlinear factor analysis where both the factors and the mapping between the factors and observed data are unknown. The faBART MCMC algorithm alternates between sampling the from the posterior of the latent factors and a functional approximation to the unknown mapping. The latter step utilizes BART of Chipman et al. (2010), a method for non-parametric regression which uses a sum-of-trees model to estimate a broad class of functions.

On a number of simulated datasets, we demonstrated that faBART can successfully find both the unobserved factors and their functional relationship to the observed data $\mathbf{Y}$. On two canonical datasets used in dimension reduction, we highlighted that faBART can find meaningful low-dimensional embeddings of the data.

We note that faBART requires estimating the fitted values from BART, $\mathbf{H}$, at every iteration of the MCMC algorithm and as such, has a high computational cost. However, there have been recent developments in the speeding up of the original BART algorithm which may prove useful here (He et al., 2018). Moreover, this computationally burdensome step is embarrassingly parallel and so may benefit from the use of multiple computing cores.

4.9. Appendix

4.9.1. Elliptical slice sampler

We also consider an elliptical slice sampler to draw from the posterior for $\mathbf{X}$ (Murray et al., 2010). The elliptical slice sampler is feasible as it only requires the prior for $\mathbf{X}$ to be a Gaussian, while the likelihood may be any function of $\mathbf{X}$. The elliptical slice sampler always accepts a draw of $\mathbf{X}$. Given the previous draw $\mathbf{X}^{(t)}$, the process for sampling a new draw $\widetilde{\mathbf{X}}$ is as follows:

1. Sample an ellipse $\boldsymbol{\nu} \sim MN(0, \tau^2\mathbf{I})$

2. Calculate log-likelihood threshold:

$$u \sim \text{Uniform}[0, 1] \tag{4.42}$$

$$\log y \leftarrow \log L(\mathbf{Y}|\mathbf{X}^{(t)}, \mathbf{H}^{(t)}, \boldsymbol{\Sigma}^{(t)}) + \log u \tag{4.43}$$

3. Draw an initial proposal and define a bracket:

$$\theta \sim \text{Uniform}[0, 2\pi] \tag{4.44}$$

$$[\theta_{min}, \theta_{max}] \leftarrow [\theta - 2\pi, \theta] \tag{4.45}$$

4. $\widetilde{\mathbf{X}} \rightarrow \mathbf{X}^{(t)} \cos(\theta) + \boldsymbol{\nu} \sin(\theta)$.

5. If $\log L(\mathbf{Y}|\widetilde{\mathbf{X}}, \widetilde{\mathbf{H}}, \widetilde{\boldsymbol{\Sigma}}) > \log y$, return $\widetilde{\mathbf{X}}$

6. Else, shrink the bracket and try a new point:

   (a) If $\theta < 0$, then $\theta_{min} \rightarrow \theta$

   (b) Else, $\theta_{max} \rightarrow \theta$

7. $\theta \sim \text{Uniform}[\theta_{min}, \theta_{max}]$

8. Go to 4.

However, we found that the elliptical slice sampler would often get "stuck" as it continues to loop until accepting a draw. We postulate that this is because the BART likelihood is not differentiable with respect to $\mathbf{X}$. In contrast, the Metropolis algorithm appears better able to "jump" up or down the steps in the BART likelihood.

4.9.2. Distance-based proposal

We also consider an alternate proposal distribution to the spherical Gaussian used in Algorithm 2. This proposal distribution is also a multivariate Gaussian centered at the previous

draw, $\mathbf{X}^{(t)}$; however, it uses a covariance matrix that is a distance matrix of $\mathbf{Y}$. This is to encourage draws $\widetilde{\mathbf{x}}_i$ and $\widetilde{\mathbf{x}}_j$ that are similar if their corresponding feature vectors $\mathbf{y}_i$ and $\mathbf{y}_j$ are similar. Specifically, we considered the matrix normal proposal:

$$\widetilde{\mathbf{X}} \sim \mathcal{MN}(\mathbf{X}^{(t)}, c^2\mathbf{D}, \mathbf{I}_{K\times K}), \tag{4.46}$$

where $\mathbf{D} \in \mathbb{R}^{N\times N}$ is a distance matrix of $\mathbf{Y}$ and c^2 is a tuning parameter. We considered a distance matrix with a Gaussian kernel:

$$\mathbf{D}_{ij} = \exp\left\{-\frac{1}{2}\|\mathbf{y}_i - \mathbf{y}_j\|^2\right\}, \quad 1 \le i, j \le N. \tag{4.47}$$

Ultimately, however, this proposal distribution yielded similar results to the spherical Gaussian in the visualization examples in Section 4.7, and somewhat worse results in the parametric examples in Section 4.6.

4.9.3. Additional Linear Example

Here, we provide an additional simulated example of faBART. We consider the linear setup used by Ročková and George (2016) to illustrate their method for sparse (linear) factor analysis. The dimensions we consider are smaller, however, with $N = 100$, $G = 105$ and $K = 5$. The data is generated as:

$$\mathbf{y}_i = \mathbf{B}\mathbf{x}_i + \boldsymbol{\varepsilon}_i, \quad \boldsymbol{\varepsilon}_i \sim N(0, \sigma^2\mathbf{I}), \quad i = 1, \ldots, N \tag{4.48}$$

where the factors are drawn from a standard normal $\mathbf{x}_i \sim N(0, \mathbf{I})$. The columns of the loadings matrix $\mathbf{B}$ each have 25 elements equal to one and the remaining elements zero; each column shares five overlapping non-zero elements with the adjacent columns (Figure 23). As a result, $\mathbf{B}$ is not an orthogonal matrix. The goal is to recover the loadings matrix $\mathbf{B}$, using only the observed data $\mathbf{Y}$.

We ran faBART for 2000 iterations with a burn-in period of 1500 iterations. The prior

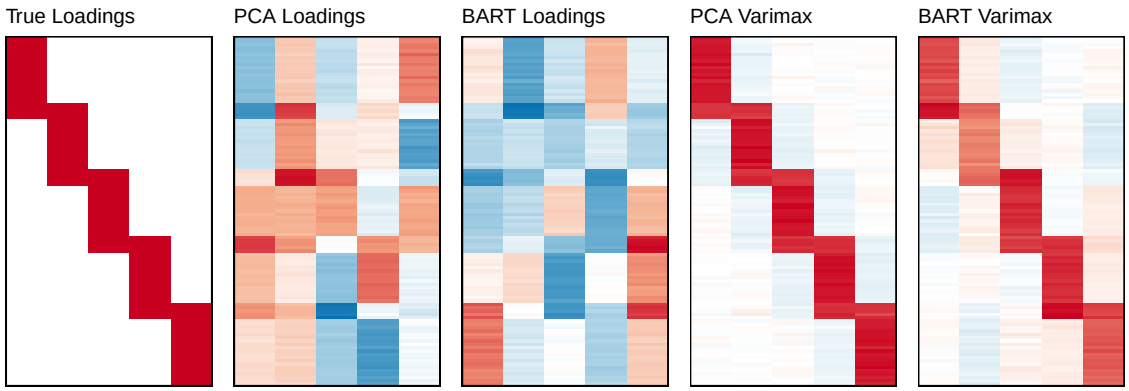


Figure 23: Section 9.3: From left to right: (i) true $\mathbf{B}$ loadings matrix; (ii) PCA loadings matrix; (iii) faBART loadings matrix $\widehat{\mathbf{B}}$; (iv) PCA loadings matrix after varimax rotation; (v) faBART loadings matrix after varimax rotation.

variance for the factors was set to $\tau^2 = 10$. Of course, faBART provides posterior draws of the latent factors $\mathbf{x}_i$ and the fitted values $\mathbf{h}_i$ only, and not the loadings matrix $\mathbf{B}$ as no linear relationship between the observed $\mathbf{y}_i$ and latent $\mathbf{x}_i$ is assumed. As a result, we estimate how well faBART captures this true linear relationship by finding the implicit loadings matrix $\widehat{\mathbf{B}} = [\widehat{\mathbf{X}}^T\widehat{\mathbf{X}}]^{-1}\widehat{\mathbf{X}}\mathbf{Y}$, where $\widehat{\mathbf{X}}$ is the posterior mean of the factors calculated from the samples after burn-in (Figure 23).

As a comparison, we also show the PCA loadings matrix (including only the first five components). In Figure 23 we see that the loadings matrices from PCA and faBART are not sparse; neither model explicitly models sparsity in the loadings, however, so this is to be expected. As such, we implement a varimax rotation (Kaiser, 1958) to both PCA and faBART to better visualize the latent structure found by each method (Figure 23). The varimax step rotates the loadings to a coordinate system where they are either large or very small. After this varimax step, we can see that faBART, along with PCA, recovers the original sparse structure of the loadings matrix. We note that even after the varimax rotation, faBART features larger loading values than PCA in sparse regions of the loadings matrix. However, PCA explicitly assumes a linear relationship between the data and the loadings matrix, unlike faBART, and so may be expected to better reconstruct the true loadings matrix in this example.

4.9.4. Implementation settings

Here, we provide the implementation details for the methods used in this chapter. All methods were implemented using R.

- t-SNE: we used the `Rtsne` package (Krijthe, 2015) with the default settings.
- Variational autoencoder: we used a script from the R Studio GitHub, accessed from: `https://github.com/rstudio/keras/blob/master/vignettes/examples/variational_autoencoder.R`. For the Swiss-roll dataset, we changed the dimension of the hidden layer to two, and the loss function to mean squared error. For MNIST, we retained the original settings.
- Isomap: we used the function `Isomap` with the default settings from the `RDRToolbox` package (Bartenhagen, 2018).

BIBLIOGRAPHY

O. Aguilar and M. West. Bayesian dynamic factor models and portfolio allocation. *Journal of Business & Economic Statistics*, 18(3):338–357, 2000.

A. Armagan, M. Clyde, and D. B. Dunson. Generalized beta mixtures of Gaussians. In *Advances in Neural Information Processing Systems*, pages 523–531, 2011.

C. Bartenhagen. *RDRToolbox: A package for nonlinear dimension reduction with Isomap and LLE.*, 2018. R package version 1.32.0.

M. J. Bayarri, J. O. Berger, A. Forte, and G. García-Donato. Criteria for Bayesian model choice with application to variable selection. *The Annals of Statistics*, 40(3):1550–1577, 2012.

A. Belloni, V. Chernozhukov, and L. Wang. Pivotal estimation via square-root lasso in nonparametric regression. *The Annals of Statistics*, 42(2):757–788, 2014.

A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: the order-preserving submatrix problem. *Journal of Computational Biology*, 10(3-4):373–384, 2003.

J. O. Berger, L. R. Pericchi, and J. A. Varshavsky. Bayes factors and marginal distributions in invariant situations. *Sankhya Ser. A*, 60:307–321, 1998.

S. Bergmann, J. Ihmels, and N. Barkai. Iterative signature algorithm for the analysis of large-scale gene expression data. *Phys Rev E Stat Nonlin Soft Matter Phys*, 67(3):031902, 2003.

A. Bhadra, J. Datta, N. G. Polson, and B. Willard. Default Bayesian analysis with global-local shrinkage priors. *Biometrika*, 103(4):955–969, 2016.

A. Bhadra, J. Datta, N. G. Polson, and B. Willard. Horseshoe Regularization for Feature Subset Selection. *ArXiv e-prints*, Feb. 2017.

A. Bhattacharya and Y. Cui. A GPU-accelerated algorithm for biclustering analysis and detection of condition-dependent coexpression network modules. *Scientific Reports*, 7(1): 4162, 2017.

A. Bhattacharya and R. K. De. Bi-correlation clustering algorithm for determining a set of co-regulated genes. *Bioinformatics*, 25(21):2795–2801, 2009.

B. Bolstad. *preprocessCore: A collection of pre-processing functions*, 2018. URL `https://github.com/bmbolstad/preprocessCore`. R package version 1.44.0.

B. Bolstad, R. Irizarry, M. strand, and T. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185–193, 01 2003.

D. Bozdağ, J. D. Parvin, and U. V. Catalyurek. A biclustering method to discover co-regulated genes using diverse gene expression datasets. In *Bioinformatics and Computational Biology*, pages 151–163. Springer, 2009.

D. Bozdağ, A. S. Kumar, and U. V. Catalyurek. Comparative analysis of biclustering algorithms. In *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 265–274. ACM, 2010.

C. M. Carvalho, J. Chang, J. E. Lucas, J. R. Nevins, Q. Wang, and M. West. High-dimensional sparse factor modeling: applications in gene expression genomics. *Journal of the American Statistical Association*, 103(484):1438–1456, 2008.

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

I. Castillo and A. van der Vaart. Needles and straw in a haystack: Posterior concentration for possibly sparse sequences. *The Annals of Statistics*, 40(4):2069–2101, 2012.

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

Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the International Conference on Intelligent Systems for Molecular Biology*, volume 8, pages 93–103, San Diego, U.S.A., 2000.

H. A. Chipman, E. I. George, and R. E. McCulloch. Bayesian CART model search. *Journal of the American Statistical Association*, 93(443):935–948, 1998.

H. A. Chipman, E. I. George, and R. E. McCulloch. Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298, 2010.

M. A. Clyde and G. Parmigiani. Protein construct storage: Bayesian variable selection and prediction with mixtures. *Journal of Biopharmaceutical Statistics*, 8(3):431–443, 1998.

M. A. Clyde, J. Ghosh, and M. L. Littman. Bayesian adaptive sampling for variable selection and model averaging. *Journal of Computational and Graphical Statistics*, 20(1):80–101, 2011. doi: 10.1198/jcgs.2010.09049.

P. A. De Castro, F. O. de França, H. M. Ferreira, and F. J. Von Zuben. Evaluating the performance of a biclustering algorithm applied to collaborative filtering-a comparative analysis. In *7th International Conference on Hybrid Intelligent Systems (HIS 2007)*, pages 65–70. IEEE, 2007.

M. Denitto, M. Bicego, A. Farinelli, and M. A. Figueiredo. Spike and slab biclustering. *Pattern Recognition*, 72:186–195, 2017.

S. K. Deshpande, V. Ročková, and E. I. George. Simultaneous Variable and Covariance Selection with the Multivariate Spike-and-Slab Lasso. *ArXiv e-prints*, Aug. 2017.

V. Dorie, H. Chipman, and R. McCulloch. *dbarts: Discrete Bayesian Additive Regression Trees Sampler*, 2018. R package version 0.9-8.

F. Doshi, K. Miller, J. Van Gael, and Y. W. Teh. Variational inference for the Indian buffet process. In *Artificial Intelligence and Statistics*, pages 137–144, 2009.

K. Eren, M. Deveci, O. Küçüktunç, and Ü. V. Çatalyürek. A comparative analysis of biclustering algorithms for gene expression data. *Briefings in Bioinformatics*, 14(3):279–292, 2012.

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

N. Fan, N. Boyko, and P. M. Pardalos. Recent advances of data biclustering with application in computational neuroscience. In *Computational Neuroscience*, pages 85–112. Springer, 2010.

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

S. Fruehwirth-Schnatter and H. F. Lopes. Sparse Bayesian factor analysis when the number of factors is unknown. *arXiv preprint arXiv:1804.04231*, 2018.

C. Gao, I. C. McDowell, S. Zhao, C. D. Brown, and B. E. Engelhardt. Context specific and differential gene co-expression networks via Bayesian biclustering. *PLoS Computational Biology*, 12(7):e1004791, 2016.

A. Gelman. Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 2004.

A. Gelman, G. O. Roberts, and W. R. Gilks. Efficient Metropolis jumping rules. *Bayesian Statistics*, 5(599-608):42, 1996.

A. Gelman, J. B. Carlin, H. S. Stern, D. B. Dunson, A. Vehtari, and D. B. Rubin. *Bayesian Data Analysis*, volume 2. CRC press Boca Raton, FL, 2014.

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

E. I. George and R. E. McCulloch. Approaches for Bayesian variable selection. *Statistica Sinica*, 7:339–373, 1997.

T. L. Griffiths and Z. Ghahramani. Infinite latent feature models and the Indian buffet process. In *Proceedings of the 18th International Conference on Neural Information Processing Systems*, pages 475–482. MIT Press, 2005.

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

J. A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972.

T. Hastie, R. Tibshirani, B. Narasimhan, and G. Chu. *impute: Imputation for microarray data*, 2018. R package version 1.56.0.

J. He, S. Yalov, and P. R. Hahn. Accelerated Bayesian additive regression trees. *arXiv preprint arXiv:1810.02215*, 2018.

J. L. Hill. Bayesian nonparametric modeling for causal inference. *Journal of Computational and Graphical Statistics*, 20(1):217–240, 2011.

S. Hochreiter, U. Bodenhofer, M. Heusel, A. Mayr, A. Mitterecker, A. Kasim, T. Khamiakova, S. Van Sanden, D. Lin, and W. Talloen. FABIA: factor analysis for bicluster acquisition. *Bioinformatics*, 26(12):1520–1527, 2010.

D. Horta and R. J. G. B. Campello. Similarity measures for comparing biclusterings. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 11(5):942–954, Sept. 2014. ISSN 1545-5963.

M. Huang, J. Wang, E. Torre, H. Dueck, S. Shaffer, R. Bonasio, J. I. Murray, A. Raj, M. Li, and N. R. Zhang. Saver: gene expression recovery for single-cell RNA sequencing. *Nature Methods*, 15(7):539, 2018.

S. Islam, A. Zeisel, S. Joost, G. La Manno, P. Zajac, M. Kasper, P. Lönnerberg, and S. Linnarsson. Quantitative single-cell RNA-seq with unique molecular identifiers. *Nature Methods*, 11(2):163–166, 2014.

H. Jeffreys. *The Theory of Probability*. Oxford University Press, 3 edition, 1961.

H. F. Kaiser. The varimax criterion for analytic rotation in factor analysis. *Psychometrika*, 23(3):187–200, 1958.

P. Kellokumpu-Lehtinen, M. Talpaz, D. Harris, Q. Van, R. Kurzrock, and Z. Estrov. Leukemia-inhibitory factor stimulates breast, kidney and prostate cancer cell proliferation by paracrine and autocrine pathways. *International Journal of Cancer*, 66(4):515–519, 1996.

D. P. Kingma and M. Welling. Auto-encoding variational Bayes. *arXiv preprint arXiv:1312.6114*, 2013.

J. H. Krijthe. *Rtsne: T-Distributed Stochastic Neighbor Embedding using Barnes-Hut Implementation*, 2015. URL `https://github.com/jkrijthe/Rtsne`. R package version 0.15.

N. Lawrence. Probabilistic non-linear principal component analysis with Gaussian process latent variable models. *Journal of Machine Learning Research*, 6(Nov):1783–1816, 2005.

L. Lazzeroni and A. Owen. Plaid models for gene expression data. *Statistica Sinica*, pages 61–86, 2002.

Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

J. T. Leek and J. D. Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics*, 3(9):e161, 2007.

F. Liang, R. Paulo, G. Molina, M. A. Clyde, and J. O. Berger. Mixtures of g priors for Bayesian variable selection. *Journal of the American Statistical Association*, 103(481): 410–423, 2008.

A. R. Linero. Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association*, 113(522):626–636, 2018.

C. Liu, D. B. Rubin, and Y. N. Wu. Parameter expansion to accelerate EM: the PX-EM algorithm. *Biometrika*, 85(4):755–770, 1998.

P.-L. Loh. Statistical consistency and asymptotic normality for high-dimensional robust m-estimators. *The Annals of Statistics*, 45(2):866–896, 2017.

L. v. d. Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 1(1):24–45, 2004.

R. Mazumder, J. H. Friedman, and T. Hastie. Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the American Statistical Association*, 106(495):1125–1138, 2011.

T. J. Mitchell and J. J. Beauchamp. Bayesian variable selection in linear regression. *Journal of the American Statistical Association*, 83(404):1023–1032, 1988.

G. E. Moran, V. Ročková, and E. I. George. Variance prior forms for high-dimensional Bayesian variable selection. *Bayesian Analysis*, 2018.

A. Mucherino, P. Papajorgji, and P. M. Pardalos. *Data mining in agriculture*, volume 34. Springer Science & Business Media, 2009.

J. Munkres. Algorithms for the assignment and transportation problems. *Journal of the Society for Industrial and Applied Mathematics*, 5(1):32–38, 1957.

I. Murray, R. Adams, and D. MacKay. Elliptical slice sampling. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 541–548, 2010.

R. M. Neal. Sampling from multimodal distributions using tempered transitions. *Statistics and Computing*, 6(4):353–366, 1996.

A. A. Onitilo, J. M. Engel, R. T. Greenlee, and B. N. Mukesh. Breast cancer subtypes based on ER/PR and Her2 expression: comparison of clinicopathologic features and survival. *Clinical Medicine & Research*, 7(1-2):4–13, 2009.

V. A. Padilha and R. J. Campello. A systematic comparative evaluation of biclustering techniques. *BMC Bioinformatics*, 18(1):55, 2017.

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

R. Peeters. The maximum edge biclique problem is NP-complete. *Discrete Applied Mathematics*, 131(3):651–654, 2003.

J. Piironen and A. Vehtari. On the hyperprior choice for the global shrinkage parameter in the horseshoe prior. In *Artificial Intelligence and Statistics*, pages 905–913, 2017.

N. G. Polson and J. G. Scott. Shrink globally, act locally: sparse Bayesian regularization and prediction. *Bayesian Statistics*, 9:501–538, 2010.

A. Prelić, S. Bleuler, P. Zimmermann, A. Wille, P. Bühlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, 2006.

A. V. Rangan, C. C. McGrouther, J. Kelsoe, N. Schork, E. Stahl, Q. Zhu, A. Krishnan, V. Yao, O. Troyanskaya, and S. Bilaloglu. A loop-counting method for covariate-corrected low-rank biclustering of gene-expression and genome-wide association study data. *PLoS Computational Biology*, 14(5):e1006105, 2018.

T. Reya, S. J. Morrison, M. F. Clarke, and I. L. Weissman. Stem cells, cancer, and cancer stem cells. *Nature*, 414(6859):105, 2001.

C. P. Robert, N. Chopin, and J. Rousseau. Harold Jeffreys's Theory of Probability Revisited. *Statistical Science*, pages 141–172, 2009.

V. Ročková. Bayesian estimation of sparse signals with a continuous spike-and-slab prior. *The Annals of Statistics*, 46(1):401–437, 2018.

V. Ročková and E. I. George. EMVS: The EM approach to Bayesian variable selection. *Journal of the American Statistical Association*, 109(506):828–846, 2014.

V. Ročková and E. I. George. Fast Bayesian factor analysis via automatic rotations to sparsity. *Journal of the American Statistical Association*, 111(516):1608–1622, 2016.

V. Ročková and E. I. George. The Spike-and-Slab LASSO. *Journal of the American Statistical Association*, 113(521):431–444, 2018.

V. Ročková and E. Saha. On theory for BART. *arXiv preprint arXiv:1810.00787*, 2018.

V. Ročková and S. van der Pas. Posterior concentration for Bayesian regression trees and their ensembles. *arXiv preprint arXiv:1708.08734*, 2017.

V. Ročková and G. Moran. *SSLASSO: The Spike-and-Slab LASSO (R Package)*, 2017. URL `https://cran.r-project.org/package=SSLASSO`.

V. Ročková and G. Moran. *EMVS: The Expectation-Maximization Approach to Bayesian Variable Selection (R Package)*, 2018. URL `https://cran.r-project.org/package=EMVS`.

S. T. Roweis and L. K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

B. Schölkopf, A. Smola, and K.-R. Müller. Kernel principal component analysis. In *International Conference on Artificial Neural Networks*, pages 583–588. Springer, 1997.

M. Schroeder, B. Haibe-Kains, A. Culhane, C. Sotiriou, G. Bontempi, and J. Quackenbush. *breastCancerNKI: Genexpression dataset published by van't Veer et al. [2002] and van de Vijver et al. [2002] (NKI).*, 2011. URL `http://compbio.dfci.harvard.edu/`. R package version 1.12.0.

A. A. Shabalin, V. J. Weigman, C. M. Perou, and A. B. Nobel. Finding large average submatrices in high dimensional data. *The Annals of Applied Statistics*, 3(3):985–1012, 2009.

N. Städler, P. Bühlmann, and S. Van De Geer. l1 penalization for mixture regression models. *Test*, 19(2):209–256, 2010.

T. Sun and C.-H. Zhang. Comments on: l1-penalization for mixture regression models. *Test*, 19(2):270–275, 2010.

T. Sun and C.-H. Zhang. Scaled sparse linear regression. *Biometrika*, 2012.

Y. W. Teh, D. Grür, and Z. Ghahramani. Stick-breaking construction for the Indian buffet process. In *Artificial Intelligence and Statistics*, pages 556–563, 2007.

J. B. Tenenbaum, V. De Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, 2000.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

M. J. Van De Vijver, Y. D. He, L. J. Van't Veer, H. Dai, A. A. Hart, D. W. Voskuil, G. J. Schreiber, J. L. Peterse, C. Roberts, and M. J. Marton. A gene-expression signature as a predictor of survival in breast cancer. *New England Journal of Medicine*, 347(25): 1999–2009, 2002.

S. van der Pas, J.-B. Salomond, and J. Schmidt-Hieber. Conditions for posterior contraction in the sparse normal means problem. *Electronic Journal of Statistics*, 10(1):976–1000, 2016.

L. J. Van't Veer, H. Dai, M. J. Van De Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. Van Der Kooy, M. J. Marton, and A. T. Witteveen. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415(6871):530, 2002.

M. L. Whitfield, L. K. George, G. D. Grant, and C. M. Perou. Common markers of proliferation. *Nature Reviews Cancer*, 6(2):99, 2006.

M. Xiong, B. Li, Q. Zhu, Y.-X. Wang, and H.-Y. Zhang. Identification of transcription factors for drug-associated gene modules and biomedical implications. *Bioinformatics*, 30 (3):305–309, 2013.

Y. Xu, J. Lee, Y. Yuan, R. Mitra, S. Liang, P. Müller, and Y. Ji. Nonparametric Bayesian bi-clustering for next generation sequencing count data. *Bayesian Analysis*, 8(4):759–780, 2013.

I. Yalcin and Y. Amemiya. Nonlinear factor analysis as a statistical method. *Statistical Science*, 16(3):275–294, 2001.

G. Yu. *enrichplot: Visualization of Functional Enrichment Result*, 2018. URL `https://github.com/GuangchuangYu/enrichplot`. R package version 1.2.0.

G. Yu, L.-G. Wang, Y. Han, and Q.-Y. He. clusterprofiler: an R package for comparing biological themes among gene clusters. *OMICS: A Journal of Integrative Biology*, 16(5): 284–287, 2012.

A. Zeisel, A. B. Muñoz-Manchado, S. Codeluppi, P. Lönnerberg, G. La Manno, A. Juréus, S. Marques, H. Munguba, L. He, and C. Betsholtz. Cell types in the mouse cortex and hippocampus revealed by single-cell RNA-seq. *Science*, 347(6226):1138–1142, 2015.

T. Zhan, N. Rindtorff, and M. Boutros. Wnt signaling in cancer. *Oncogene*, 36(11):1461, 2017.

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

C.-H. Zhang and T. Zhang. A general theory of concave regularization for high-dimensional sparse estimation problems. *Statistical Science*, pages 576–593, 2012.

M. H. Zhang, H. T. Man, X. D. Zhao, N. Dong, and S. L. Ma. Estrogen receptor-positive breast cancer molecular signatures and therapeutic potentials. *Biomedical Reports*, 2(1): 41–52, 2014.

Y. Zhu, X. Shen, and C. Ye. Personalized prediction and sparsity pursuit in latent factor models. *Journal of the American Statistical Association*, 111(513):241–252, 2016.

H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.