the University of Warwick

9th September 2016

Orthogonalisation methods for fast computing of Bayesian model selection

Author:
Leonardo Petrini

Supervisor:
David Rossell

In the Big Data generation, performing Bayesian model selection in a computationally fast manner is a main challenge in Statistics. In this work we explore methods to orthogonalise a general form Gram matrix $X^T X$, and employ them to carry out scalable Bayesian model selection and averaging in a linear regression context. For this purpose, both PCA-related techniques and the novel DECO method are examined, for increasing p and correlation amongst the predictors. Finally, we consider both situations where the number of variables p is small and large.

Keywords: Model Selection, Decorrelation, Bayesian Model Averaging, Shrinkage, Sparse PCA.

Acknowledgements

I would like to express my deep gratitude to Professor David Rossell, who has proved not only to be a great example and guide, but also a vibrant person always on search for improvement. Without his great supervision and intuition, such an enjoyable experience would have never been possible. In addition, I want to thank Professor Omiros Papaspiliopoulos and Professor Chenlei Leng for their persistent support, and great insights at various stages of this dissertation. Working with you all made me feel part of a team, and it allowed me to see the research world from the inside, as I have always dreamed of doing.

1. Introduction

Technological advances in the last decades have generated remarkably larger datasets, which have created a number of challenges for modern statisticians, particularly in terms of high-dimensional inference and computational scalability. In this work, we focus our attention on variable selection in linear regression. By variable selection, we refer to the practice of selecting a subset of variables from a large set of features, to help explain or predict an observed continuous outcome in a regression setting.

One main challenge related to this endeavour is to find a statistically sound methodology to correctly extract such information. There has been an intensive research effort both in the frequentist and the Bayesian frameworks in recent times, resulting in an improved understanding of the properties of the various methodological approaches currently available. A second issue, which is the main aim of this thesis, is addressing computational bottlenecks. This is important because some of the formulations theoretically known to lead to better inference involve intensive calculations, forcing practitioners to use less precise but faster methods. In this work, we'll review such techniques, and bond them together with sparse orthogonalisation methods to balance computational speed and quality of inference.

The rest of the paper is organised as follows. In Chapter 2, we review the main framework and adopted methods, so that we first start describing some of the Bayesian variable selection methods used at present, and some of their main properties. For each method, both advantages and disadvantages are briefly discussed, putting particular attention on the computational effort. Further, a novel approach (Papaspiliopoulos & Rossell, 2016) which assumes block-diagonal $X^T X$ to efficiently solve selection problems for large $p \approx n$ will be reviewed.

The basic idea underlying this thesis is that this computational framework can be used in combination with orthogonalisation techniques to deliver fast approximations to variable selection.

To this end, transformation methods to render the Gram Matrix orthogonal are also analysed in Chapter 2. Further, Chapter 3 provides empirical results comparing the different available methods, with both simulated and real data; both situations in which $p \leq n$ and $p > n$ are treated. Finally, Chapter 4 offers some concluding remarks and future possible research directions.

2. Problem overview, and related previous work

2.1 Variable selection

An important problem in Statistics is how to choose an optimal model from a set of plausible models. In the context of linear regression, the aim is to select the best subset of variables among p potential predictors. Ideally, we would like to perform this task in both a statistical and computationally efficient way, making sure that we pick the truly relevant variables and that they explain a large proportion of the variability in the response. We focus our attention on the Bayesian paradigm, which we now outline, although we also briefly refer to some related work in the penalised likelihood literature.

2.1.1 Bayesian variable selection

Consider the normal linear model $y \sim \mathcal{N}(X\beta, \phi I)$ where $y = (y_1, \ldots, y_n)^T \in \mathbb{R}^n$ is a vector containing the response variable, X is an n x p design matrix with n individuals and p covariates, $\beta \in \mathbb{R}^p$ is the regression coefficients vector, and $\phi \in \mathbb{R}^+$ is the residual variance. Further, denote as x_j the jth column of X.

We define an auxiliary indicator variable $\gamma = (\gamma_1, \ldots, \gamma_p)$ such that $\gamma_j = I(\beta_j \neq 0)$ indicates presence, and $\gamma_j = I(\beta_j = 0)$ indicate absence of covariate j in the model. Further, let $|\gamma| = \sum_{j=1}^{p} \gamma_j$ be the number of variables included in model γ. We denote by X_γ the n x $|\gamma|$ submatrix of X and β_γ are the corresponding regression coefficients. Throughout this paper, the null model for which $\gamma_j = 0$ for all variables will be denoted by $\emptyset$.

The main quantity of interest is $p(\gamma \mid y) = \frac{p(\gamma)p(y|\gamma)}{p(y)} = \frac{p(\gamma)\int p(y|\beta_\gamma,\gamma)p(\beta_\gamma|\gamma)d\beta_\gamma}{p(y)}$ where $p(y \mid \gamma)$ is called the integrated marginal likelihood, and $p(\beta_\gamma \mid \gamma)$ is the prior density on β under model γ.

Given this model, variable selection involves deciding which of the regression coefficients are equal to zero. The approach described so far is known as "spike and slab" (Mitchell & Beauchamp, 1988), and is based on defining the prior β_j as a mixture of two distributions, in which the "spike" part is described by a point mass at $\beta_j = 0$, while the "slab" component is a continuous distribution, and specifically a Normal distribution. Formally, this prior can be written as:

$$p(\beta \mid \phi, \tau, \gamma) = \prod_{j=1}^{p} (1 - \gamma_j) I(\beta_j = 0) + \gamma_j \mathcal{N}(\beta; 0, \phi\tau_j) \tag{2.1}$$

A second type of approach, known as Stochastic Search Variable Selection (SSVS) (George & McCulloch, 1993), replaces the discrete point mass with a second Normal distribution:

$$p(\beta \mid \phi, \tau, \tau^*, \gamma) = \prod_{j=1}^{p} (1 - \gamma_j)\mathcal{N}(\beta; 0, \phi\tau_j^*) + \gamma_j \mathcal{N}(\beta; 0, \phi\tau_j) \tag{2.2}$$

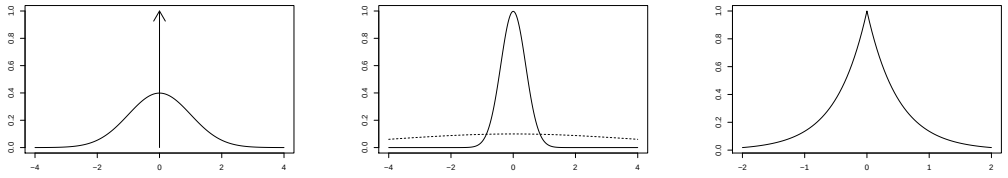
where $\tau_j \gg \tau_j^*$, so that β_j is close to zero when $\gamma_j = 0$, while τ_j is large enough to allow a large range of values of β_j when $\gamma_j = 1$.

A third alternative takes advantage of Shrinkage priors, in which the fundamental expression now becomes:

$$p(\beta \mid \phi, \tau, \gamma) = \int \mathcal{N}(\beta; 0, \phi\tau) p(\tau) d\tau \tag{2.3}$$

where $p(\tau)$ shrinks the values of the β_j's. In this setting, several priors can be adopted, such as the Laplacian (LASSO) (Tibshirani, 1996 ; Tipping, 2001), and the Horseshoe (Carvalho *et al.* , 2009) distributions, that in the Bayesian framework can be seen as assigning an informative prior to the regression coefficients.

To illustrate, Figure 2.1 includes the three cases considered, in which (a) represents the point-mass approach, (b) the SSVS Normal mixture prior, and (c) the shrinkage Laplacian prior.



(A) Normal and point mass (B) Mixture of Normals (C) Laplacian

FIGURE 2.1: Illustration of common priors for β_j

A convenient choice used in the following sections is to set $p(\beta_\gamma \mid \tau, \gamma) = \mathcal{N}(\beta_\gamma; 0, \tau\phi(X_\gamma^T X\gamma)^{-1})$ as Zellner's prior for β, where τ is a prior dispersion parameter, while an inverse gamma prior is adopted for ϕ. Such specification has the advantage of returning a closed-form expression for $p(y \mid \gamma)$ that depends on a simple goodness of fit statistic.

Other quantities that may be of interest for inference are $p(\gamma_j = 1 \mid y, \phi, \rho, \tau)$ the marginal inclusion probabilities, and $E(\beta_j \mid y, \phi, \rho, \tau)$ the estimated regression coefficients with variances $Var(\beta_j \mid y, \phi, \rho, \tau)$, where $\rho = P(\gamma_j = 1)$.

When p is small, one can simply enumerate all the 2^p possible models in order to evaluate $p(\gamma \mid y)$ and find $\gamma^* = \underset{\gamma}{\operatorname{argmax}}\, p(\gamma \mid y)$. However, when p is large more sophisticated algorithms are required. Such techniques are going to be explored in the next sections.

2.1.2 Overview of model search methods

There is a large literature on strategies for model search. We now briefly outline the main families of techniques, both in the frequentist and in the Bayesian world, stressing their advantages and limits, both from a mathematical and a computational point of view.

Deterministic model search: best subset methods (Miller, 2002) to find γ^* have been very well studied in the literature, giving rise to very interesting methods. This is done by firstly identifying a subset of $k < p$ predictors considered as important, to then explore the reduced set of variables by using least squares. Especially in the high dimensional regime, it is often required that the underlying true β coefficients are sparse. Mathematically, the problem involves optimising:

$$\underset{\beta}{\operatorname{argmin}} \frac{1}{2}||y - X\beta||_2^2 \tag{2.4}$$

subject to $||\beta||_0 \leq k$, where $||\beta||_0$ counts the number of non zero elements in β. This cardinality constraint makes problem 2.4 NP-hard (Natarajan, 1995), which can for example be solved through "leap and bound" based techniques (Furnival & Wilson, 2000) without examining all possible combinations, however this strategy doesn't scale to problems with $p > 30$. Nonetheless, best subset approach has been recently rediscovered by Bertsimas *et al.* (2016) , adopting mixed integer optimisation (MIO) to aid the usage of these methods in settings with moderately large p.

Although this class of techniques successfully finds the best model, it is not applicable for very large p as aforementioned. Specifically, a first issue is related to computational cost: exploring a large combination of models can be very time consuming.

Markov Chain Monte Carlo methods: a class of flexible techniques, allowing the user to calculate numerical approximation of otherwise analytically intractable integrals. This is done by generating a Markov chain with transition kernel $K(\gamma^{(t)} \mid \gamma^{(t-1)})$ whose stationary distribution is $p(\gamma \mid y)$. Classical algorithms include:

- The Gibbs sampler, introduced by Geman & Geman (1984) and further developed by Gelfand *et al.* (1990) , can be used whenever it is possible to directly sample from the "full conditionals", meaning the distribution conditional upon everything except the variable being sampled at each step.
- The Metropolis-Hastings sampler (Metropolis *et al.* , 1953 ; Hastings, 1970) is based on proposing values sampled from a known distribution, which are then either accepted or rejected with a certain probability as if they were drawn from the target distribution. This can be particularly useful whenever it's not possible to draw from the full conditionals.

There are many popular MCMC algorithms for variable selection that leverage on these two main algorithms. To mention a few, reversible Jump MCMC (Green, 1995) allows the chain to sample (β, γ), in which the acceptance probability corresponds to a classical Metropolis-Hastings framework, adjusted for the change in dimension that can occur at every step in the algorithm. MC3 (Madigan *et al.* , 1995), which simply employs a Gibbs sampler to explore discrete space, resulting in a very interesting approach for model selection. More recently, Shotgun Stochastic Search (Hans *et al.* , 2007) diverges

from traditional MCMC, evaluating many candidate models in parallel at each iteration, instead of moving sequentially from one model to the next. Moreover, it explores neighbours of each model selected, aiming to end up in regions of space that contains multiple high-probability models.

Although very general, such techniques have limited scalability for large p, given the high computation cost associated with MCMC techniques for high-dimensional data, it is thus interesting to seek alternatives.

Shrinkage methods: in this scenario (Chipman *et al.* , 1997), a model involving all p predictors is fitted, but the estimated coefficients are shrunken towards zero, with the goal of achieving sparsity. One of the most famous popular methods is the so-called LASSO (Tibshirani, 1996 ; Chen *et al.* , 2001), solving:

$$\underset{\beta}{\operatorname{argmin}} \frac{1}{2}||y - X\beta||_2^2 + \rho||\beta_j||_1 \tag{2.5}$$

where the L_1 norm helps producing a sparse solution, and ρ is its tuning parameter. Although computationally attractive, the quality of the inference provided by LASSO is not optimal in presence of noise and correlated variables. To address this issue, effective non-convex optimisation based methods (Mazumder *et al.* , 2012) have been proposed, where the goal is to find:

$$\underset{\beta}{\operatorname{argmin}} \frac{1}{2}||y - X\beta||_2^2 + \sum_j p(|\beta_j|; \psi, \eta) \tag{2.6}$$

where $p(|\beta_j|; \psi; \eta)$ is a nonconvex function in β with ψ and η denoting the degree of regularization and nonconvexity, respectively. Examples of nonconvex penalties include the minimax concave penalty (MCP)(Zhang, 2010), and the smoothly clipped absolute deviation (SCAD) (Fan & Li, 2001).

2.1.3 Overview of computational strategies

Recently, Papaspiliopoulos & Rossell (2016) proposed a scalable Bayesian variable selection computational framework, which can be applied when the Gram matrix $X^T X$ is block diagonal. Such approach finds the highest posterior probability model of any given size $|\gamma| = 0, 1, \ldots, p$ without applying any numerical approximation, at $\mathcal{O}(K2^{p^*})$ cost, making it very appealing in a high-dimensional framework, where p^* represents the number of variables in the largest block. This is done by distributing the optimisation problem across blocks, and by computing a single univariate integral deterministically it returns $p(\gamma \mid y)$ for as many γ as required.

Under the assumption of a block diagonal design, $x_i^T x_j = 0$ if variables (i, j) belong to different blocks. Let $k = 1, \ldots, K$ be the block labels, and for each block k, define $b[k] \in \{0, 1\}^p$ as a binary vector that indicates which variables are in block k. To illustrate, $b[k]_j = 1$ if and only if variable j is present in block k. Further, for any two binary vectors b and γ, denote by $b\gamma$ their element-wise multiplication. Subscripting a vector or matrix by a binary vector selecting the corresponding vector elements or matrix columns, so that $X_{b\gamma}$ and $\beta_{b\gamma}$ denote the active variables in block k under model γ, and their regression coefficients respectively. Here we limit discussion to Zellner's prior for the prior density $p(\beta_{\gamma b} \mid \tau, \phi)$. Finally, it is assumed that the variable inclusion

indicators γ_j are independent, so that $p(\gamma) = \prod_{j=1}^{p} p(\gamma_j)$ and both $\omega = P(\gamma_j = 1)$ and τ are fixed constants.

In this scenario, Papaspiliopoulos & Rossell show that the marginal posteriors $p(\gamma \mid y, \omega, \tau)$ and $p(\phi \mid y, \omega, \tau)$ can be computed in closed-form expressions, and hence computed without using any Monte Carlo approximation, under the block-diagonal assumption. Specifically, the authors show that the optimisation problem depends on the data only through the goodness of fit statistic $u(y, \gamma) = y^T X_\gamma (X_\gamma^T X_\gamma)^{-1} X_\gamma^T y$ which is proportional to the squared Mahalanobis distance between the least-squares estimator of β_γ and 0. Note that in a block-diagonal design, the following result is also true:

$$\sum_{k=1}^{K} u(y, \gamma b[k]) = u(y, \gamma) \tag{2.7}$$

Based on these main ideas, a two-step algorithm to do exact variable selection was developed. The first step finds the best variable configurations of sizes $1, \ldots, |b[k]|$ within each block, while the second step combines these results across blocks, identifying the most probable models of different sizes. In the case of diagonal $X^T X$ the algorithm can be run in a single step that has cost $\mathcal{O}(p)$.

The framework also efficiently computes other posterior summaries such as $p(\gamma \mid y)$, $p(\gamma_j = 1 \mid y)$, and $\mathbb{E}(\beta \mid y)$.

2.2 Orthogonalisation methods

The main idea for the current thesis is to capitalise on the framework of Papaspiliopoulos & Rossell, by applying a preliminary orthogonalisation transform to X and potentially also y, in order to ideally obtain $\widetilde{X}$ and $\widetilde{y}$ with the following two characteristics. Firstly, the transformed data matrix $\widetilde{X}$ should have diagonal or block diagonal $\widetilde{X}^T \widetilde{X}$, either exactly or approximately. Secondly, selecting non-zero regression coefficients associated to the columns in $\widetilde{X}$ must help us selecting columns in the original data matrix X.

2.2.1 Principal component analysis

Principal component analysis (PCA), firstly introduced by Pearson (1901) and further explored by Hotelling (1933) , has been widely used as a dimension reduction tool returning an uncorrelated representation of the data matrix X. PCA sequentially projects X onto orthogonal vectors that maximise the projected variance, such that only the first few components (projected vectors) are able to explain a large proportion of the variability, successfully reducing the dimensions.

More formally, PCA defines a new data matrix $\widetilde{X} = XE$, where E is an orthonormal matrix whose p-th column is the p-th eigenvector of $X^T X$, ordered decreasingly according to their corresponding eigenvalues $\lambda_1 \geq \lambda_2 \geq \ldots, \lambda_p \geq 0$. Thus, the principal components contained in the columns of $\widetilde{X}$ are an uncorrelated linear transformation of the original data, and they can be used as input for the technique in Section 2.1.3.

The PCA problem can be formulated as a matrix approximation problem:

$$\underset{E}{\operatorname{argmin}} \, ||X - \widetilde{X} E^T||^2 \tag{2.8}$$

where $\widetilde{X}$ is n x q, and E is q x q for some given natural number $q \leq p$.

Alternatively, PCA also arises as the solution to the problem of maximising $trCov(\widetilde{X}) = trCov(XE) = tr(E^T X^T XE)$:

$$\underset{E}{\operatorname{argmax}}\, tr(E^T X^T XE) \tag{2.9}$$

subject to $E^T E = I_q$, where E is a p x q matrix, and q is the number of eigenvectors we wish to extract. In terms of the regression framework, the model $y = X\beta + \varepsilon$ now becomes:

$$y = XEE^T\beta + \varepsilon = \widetilde{X}\widetilde{\beta} + \varepsilon \tag{2.10}$$

thus, we can retrieve the regression coefficients in terms of the original variables by $\beta = E\widetilde{\beta}$.

Although PCA successfully orthogonalises X, it fails in selecting variables in the original data matrix, since every component has a non-zero loading for every variable. That is, even when $\widetilde{\beta}$ has a single non-zero element, all entries in β are non-zero, thus zeroing out elements in $\widetilde{\beta}$ does not achieve the desired goal of selecting columns in X.

2.2.2 Sparse principal component analysis

Several methods have been proposed to augment the interpretability of the PCA solution. A popular one is the so-called Sparse PCA (Zou *et al.* , 2006), which aims at finding a loadings matrix with many zero entries. The general idea (Jolliffe *et al.* , 2003) is to adopt a regression approach to PCA, including a penalisation term to the objective function to achieve sparsity:

$$\underset{E}{\operatorname{argmin}}\, ||X - \widetilde{X}E^T||^2 + \sum_{i=1}^{q} \rho_i |e_i| \tag{2.11}$$

where ρ_i are the regularization parameters, whose selected values drives the level of sparsity in the solution. One can solve the optimisation and find a new data matrix $\widetilde{X} = XE$, where E is a q x q matrix, in which each column e_i is a sparse loadings vector. Similarly to classical PCA, in the regression framework we can retrieve the original coefficients by using the identity $\beta = E\widetilde{\beta}$.

A different direction has been recently explored (Benidis *et al.* , 2016), by developing an Orthogonal Sparse PCA, in which eigenvectors preserve their orthogonality and one seeks to maximise the variance of the solution akin to (2.9):

$$\underset{E}{\operatorname{argmax}}\, tr(E^T X^T XED) - \sum_{j=1}^{q} \rho_j \sum_{i=1}^{p} g_m^{\epsilon}(e_{ij}) \tag{2.12}$$

subject to $E^T E = I_q$, where D is diagonal matrix giving weights to the eigenvectors. The problem in (2.12) involves a maximisation of a non-concave discontinuous function over a non-convex set, which can't be handled directly. In order to solve this, g_m^{ϵ} serves

as an approximation of such discontinuity in the L_0 norm, where $m > 0$ is a parameter that controls the approximation. Further, a minorisation-maximisation (MM) algorithm is adopted until convergence.

Although all the aforementioned techniques greatly augment the interpretability of the PCA output, they have one major drawback in common, being that the uncorrelatedness property of classical PCA is lost, that is $\widetilde{X}^T\widetilde{X}$ is no longer a diagonal matrix, rendering this approach sub-optimal to use as an input to the orthogonal variable selection algorithm proposed in Section 2.1.3.

A potentially interesting alternative that we explore in this thesis is the novel Least Square Sparse PCA (Merola, 2014). This method, nicknamed as LSSPCA, derives uncorrelated principal components that minimise the LS criterion, by constraining the zero norm (also known as the cardinality) of the loadings, which gives:

$$\underset{E}{\operatorname{argmin}} ||X - \widetilde{X}E^T||^2 = \underset{E}{\operatorname{argmax}} \sum_{j=1}^{q} \frac{e_j^T X^T X X^T X e_j}{e_j^T X^T X e_j} \tag{2.13}$$

subject to $L_0(e_j) \leq c_j$ and $E^T X^T X E = I_q$, where $L_0(e_j)$ represents the zero norm of the j-th loadings vector, and $c_j < p$ are the maximal cardinalities allowed. Note how the second condition implies that $\widetilde{X}^T\widetilde{X} = I_q$, successfully obtaining a diagonal transformed Gram matrix.

By adding the cardinality constraint, the solution to (2.13) is a NP-hard non-convex problem (Moghaddam *et al.* , 2005), which is computationally prohibitive. The author adopts a greedy Backward Elimination (BE) algorithm, iteratively setting to zero the smallest sparse loadings from a solution until only the ones larger than a given threshold are left. Depending on the choice of the threshold, trimming may yield solutions with too few non-zero loadings, making the choice of the threshold of crucial importance. Finally, the BE algorithm can be executed only on a subset of indices, if there is evidence for excluding some variables from a component a priori.

2.2.3 DECOrrelation technique

Wang *et al.* (2016) recently proposed a framework nicknamed DECO, based on decorrelating both the features and the response, in which one may subsequently partition the dataset in m subsets, allocated to m distributed workers. Further, the authors perform variable selection using a LASSO-type approach, giving rise to a fast algorithm, able to cope with highly correlated feature space.

When $p \leq n$, the DECO procedure is simply taking advantage of the singular value decomposition. Specifically, if $X = UDV^T$, then the regression model $y = X\beta + \varepsilon$ can now be written as:

$$\underbrace{\sqrt{p}D^{-1}U^T y}_{\widetilde{y}} = \underbrace{\sqrt{p}V^T}_{\widetilde{X}}\beta + \underbrace{\sqrt{p}D^{-1}U^T\varepsilon}_{\widetilde{\varepsilon}} \tag{2.14}$$

where U is a n x p orthogonal matrix, D is a p x p diagonal matrix and V is a p x p orthogonal matrix. Notice how the transformed Gram matrix is now orthogonal, since:

$$\widetilde{X}^T\widetilde{X} = pVV^T = pI_p \tag{2.15}$$

Alternatively, when $p > n$ or otherwise, the authors suggest introducing a different decorrelation matrix, namely $F = (XX^T/p + rI_n)^{-1/2}$, where r is a ridge refinement introduced to guarantee robustness, as the rank of XX^T after standardisation is $n - 1$. In this instance, the regression model becomes:

$$Fy = FX\beta + F\varepsilon \Rightarrow \widetilde{y} = \widetilde{X}\beta + \widetilde{\varepsilon} \tag{2.16}$$

The choice of r can introduce a different level of sensitivity to the model selection procedure, and shall thus be chosen accordingly to the desired objective; the effect of the choice of r on the results will be explored in Chapter 3.

A key feature for our purpose is that β remains unchanged in the right hand side of (2.19), and hence unlike what has been pointed out for the PCA techniques, with DECO we don't need to transform the regression coefficients for $\widehat{X}$ to those for X. A limitation is that $\widetilde{\varepsilon}$ are wrongly assumed to be independent, but the authors argue that this issue vanishes asymptotically due to the least-squares consistency.

Finally, as the authors state that the two decorrelation techniques return very close results, we will adopt in the following the second formulation both for $p \geq n$ and $p < n$, as it is applicable for both instances.

2.3 Posterior inclusion probabilities after linear transforms

As introduced in the previous sections on the various sparse PCA techniques, by applying the diagonal-block design algorithm of Section 2.1.3 to the transformed model $\widetilde{y} = \widetilde{X}\widetilde{\beta} + \widetilde{\varepsilon}$, the obtained marginal inclusion probabilities $\widehat{p}(\widetilde{\gamma}_j = 1 \mid y)$ are no longer related to the coefficients β_j's, but to the transformed $\widetilde{\beta}_j$'s.

It is therefore natural to seek a simple way to transform back the obtained $\widehat{p}(\widetilde{\gamma}_j = 1 \mid y)$ to $\widehat{p}(\gamma_j = 1 \mid y)$. In order to do so, Since $\beta = E\widetilde{\beta}$ we have that:

$\beta_j = \sum_{k=1}^{p} e_{jk}\widetilde{\beta}_k$, hence $\beta_j = 0$ if and only if for any $e_{jk} \neq 0$ we have $\widetilde{\beta}_k = 0$. That is,

$$\begin{aligned}
&\widehat{p}(\gamma_j = 0 \mid y) = P(\beta_j = 0 \mid y) = P\left(\bigcap_{e_{jk}\neq 0} \widetilde{\beta}_k = 0 \mid y\right) = \\
&\int P\left(\bigcap_{e_{jk}\neq 0} \widetilde{\beta}_k = 0 \mid y, \phi\right) p(\phi \mid y) d\phi = \\
&\int p(\phi \mid y) \prod_{k:e_{jk}\neq 0} P(\widetilde{\gamma}_k = 0 \mid y, \phi) d\phi,
\end{aligned} \tag{2.17}$$

where the last equality follows from the fact that $\widetilde{X}^T\widetilde{X}$ is diagonal. In words, $\beta_j = 0$ if and only if all of the $\widetilde{X}$'s on which x_j has a loading have no effect on y. The integrand

in Equation (2.17) has a simple expression under Zellner's prior. Thus, $\widehat{p}(\gamma_j = 1 \mid y) = 1 - \widehat{p}(\gamma_j = 0 \mid y)$

To illustrate, consider running a simple Bayesian model selection (BMS) on simulated $p = 5$ variables generated from a multivariate Normal with centered around zero and with correlation matrix $\Sigma = I_p$ for $n = 100$ observations. Further, y is generated so that the true β^* coefficients are all zero, but the last one which is $\beta_5^* = 1$.

$\widetilde{\beta}_j$	$\widehat{p}_S(\widetilde{\gamma}_j = 1 \mid y)$	$\widehat{\mathbb{E}}_S(\widetilde{\beta}_j \mid y)$
1	0.00	0.00
2	0.99	0.60
3	0.00	-0.00
4	0.99	-0.91
5	0.00	-0.00

TABLE 2.1: BMS: "SPCA" - $\widetilde{\beta}$

β_j	$\widehat{p}_S(\gamma_j = 1 \mid y)$	$\widehat{\mathbb{E}}_S(\beta_j \mid y)$
1	1.00	0.08
2	0.00	0.00
3	0.01	-0.00
4	0.01	0.00
5	1.00	1.09

TABLE 2.2: BMS: "SPCA" - β

The results are included in Tables 2.1 and 2.2, where $\widehat{p}_S(\widetilde{\gamma}_j = 1 \mid y)$ is the marginal inclusion probability of principal component j, while $\widehat{\mathbb{E}}_S(\widetilde{\beta}_j \mid y)$ is its estimate. Similarly, $\widehat{p}_S(\gamma_j = 1 \mid y)$ represents the marginal inclusion probability of variable j under the SPCA technique, while $\widehat{\mathbb{E}}_S(\beta_j \mid y)$ is its estimate, in terms of the original variables.

Commenting on the results, one can notice how the aforementioned reasoning empirically works, since as of Table 2.2, the marginal inclusion probabilities in terms of the original variables are sound, and so are the connected estimates.

	$\widetilde{x}_1$	$\widetilde{x}_2$	$\widetilde{x}_3$	$\widetilde{x}_4$	$\widetilde{x}_5$
x_1	0.00	-0.79	0.00	-0.61	0.00
x_2	0.00	0.00	-1.00	0.00	0.00
x_3	-0.67	0.00	0.10	0.00	0.75
x_4	0.75	0.00	0.00	0.00	0.67
x_5	0.00	0.61	0.00	-0.79	0.00

TABLE 2.3: Loadings Matrix

To further provide a precise explanation on how the transformation occurs, the loadings matrix E is printed in Table 2.3. As the Bayesian model selection in terms of $\widetilde{X}$ selects $\widetilde{x}_2$ and $\widetilde{x}_4$ with a very high probability (see Table 2.1), it is now clear why β_1 and β_5 have a marginal inclusion probability of 1. Moreover, since $\widetilde{x}_2$ and $\widetilde{x}_4$ have non zero values only for x_1 and x_5, it is natural that β_1 and β_5 are the only non zero coefficients as of Table 2.2.

Consider now how to obtain joint posterior model probabilities $P(\gamma \mid y)$. Let $A_\gamma = \{k : \exists j \text{ s.t. } \gamma_j = 0, e_{jk} \neq 0\}$ be the set of variables in $\widetilde{X}$ that load onto at least one of the inactive variables in γ. For model γ to hold one needs that $\widetilde{\beta}_k = 0$ for all $k \in A_\gamma$, as this implies that all elements in $\beta \setminus \beta_\gamma$ are 0, and further one needs to require that $\beta_\gamma \neq 0$.

Denote by $B_j(\gamma) = \{k : e_{jk} \neq 0\}$ the set of variables in $\widetilde{X}$ that load onto β_j, then:

$$P(\gamma \mid y) = P\left(\bigcap_{k \in A_\gamma} \widetilde{\beta}_k = 0 \bigcap_{j:\gamma_j=1} \bigcup_{B_j(\gamma)\setminus A_\gamma} \widetilde{\beta}_k \neq 0 \mid y\right) =$$

$$\int P\left(\bigcap_{k \in A_\gamma} \widetilde{\beta}_k = 0 \mid y, \phi\right) P\left(\bigcap_{j:\gamma_j=1} \bigcup_{B_j(\gamma)\setminus A_\gamma} \widetilde{\beta}_k \neq 0 \mid y, \phi\right) p(\phi \mid y) d\phi$$

$$\int \prod_{k \in A_\gamma} P\left(\widetilde{\beta}_k = 0 \mid y, \phi\right) P\left(\bigcap_{j:\gamma_j=1} \bigcup_{B_j(\gamma)\setminus A_\gamma} \widetilde{\beta}_k \neq 0 \mid y, \phi\right) p(\phi \mid y) d\phi \tag{2.18}$$

which, unlike Equation (2.17), is not trivial and can be computationally cumbersome. Since the aim of this work is to make things easier and computationally appealing, the results following in the Chapter 3 are not going to either transform $p(\gamma \mid y)$ from $p(\widetilde{\gamma} \mid y)$ or going to report such values.

2.4 Proposed model selection framework

Given all the considerations done thus far, it is now possible to outline a general framework that is going to be empirically tested in Chapter 3. For practical reasons, we will focus solely on SPCA, LSSPCA, and DECO transformation techniques outlined in Section 2.2.

Algorithm 1 Bayesian variable selection after decorrelation

Data Transformations
Define:
- Raw Data: $\widetilde{y} = y$, $\widetilde{X} = X$
- SPCA: $\widetilde{y} = y$, $\widetilde{X} = XE$
- LSSPCA: $\widetilde{y} = y$, $\widetilde{X} = XE^*$
- DECO: $\widetilde{y} = Fy$, $\widetilde{X} = FX$

Scalable Bayesian Model Selection
Obtain posterior mode $\widehat{\gamma}^*$ for $\widetilde{y} = \widetilde{X}\widetilde{\beta} + \widetilde{\varepsilon}$ using algorithm of Papaspiliopoulos & Rossell
Obtain $\widehat{p}(\widetilde{\gamma}_j = 1 \mid y)$ and $\widehat{\mathbb{E}}(\widetilde{\beta} \mid y)$, assuming diagonal $\widetilde{X}^T\widetilde{X}$, and uncorrelated $\widetilde{\varepsilon}$
Obtain:
- Raw Data: $\widehat{\mathbb{E}}_R(\beta \mid y) = \mathbb{E}(\widetilde{\beta} \mid y)$
- SPCA: $\widehat{\mathbb{E}}_S(\beta \mid y) = E\,\mathbb{E}(\widetilde{\beta} \mid y)$
- LSSPCA: $\widehat{\mathbb{E}}_L(\beta \mid y) = E^*\,\mathbb{E}(\widetilde{\beta} \mid y)$
- DECO: $\widehat{\mathbb{E}}_D(\beta \mid y) = \mathbb{E}(\widetilde{\beta} \mid y)$
- $\widehat{p}(\gamma_j = 1 \mid y)$ for each technique as of Section 2.3

3. Application

3.1 Algorithm

In this Chapter, we illustrate the proposed framework after using the transformations introduced in Section 2.2 in order to obtain Bayesian Model Averaging (BMA) estimates and marginal inclusion posterior probabilities, by using the following simulation study algorithm, which is based upon Algorithm 1:

Algorithm 2 Simulation study

Initialisation
Input $p, n, \phi^*, \beta^*, \Sigma$
Data Simulation
Simulate $x_j \sim \mathcal{N}(0, \Sigma)$; $\varepsilon_j \sim \mathcal{N}(0, I_n\phi)$ $j = 1, \ldots, n$ independent
Scale and center the columns of X to zero mean, unit variance
Generate $y = X\beta + \varepsilon$
Full Enumeration and MCMC
Obtain posterior mode γ^* for Full Enumeration
Obtain $p(\gamma_j = 1 \mid y)$ and $\mathbb{E}(\beta \mid y)$ for Full Enumeration
Obtain posterior mode $\widehat{\gamma}^*$ for MCMC
Obtain $\widehat{p}(\gamma_j = 1 \mid y)$ and $\widehat{\mathbb{E}}(\beta \mid y)$ for MCMC
Data Transformations
Define:
    - Raw Data: $\widetilde{y} = y$, $\widetilde{X} = X$
    - SPCA: $\widetilde{y} = y$, $\widetilde{X} = XE$
    - LSSPCA: $\widetilde{y} = y$, $\widetilde{X} = XE^*$
    - DECO: $\widetilde{y} = Fy$, $\widetilde{X} = FX$
Scalable Bayesian Model Selection
Obtain posterior mode $\widehat{\gamma}^*$ for $\widetilde{y} = \widetilde{X}\widetilde{\beta} + \widetilde{\varepsilon}$ using algorithm of Papaspiliopoulos & Rossell
Obtain $\widehat{p}(\widetilde{\gamma}_j = 1 \mid y)$ and $\widehat{\mathbb{E}}(\widetilde{\beta} \mid y)$, assuming diagonal $\widetilde{X}^T\widetilde{X}$, and uncorrelated $\widetilde{\varepsilon}$
Obtain:
    - Raw Data: $\widehat{\mathbb{E}}_R(\beta \mid y) = \mathbb{E}(\widetilde{\beta} \mid y)$
    - SPCA: $\widehat{\mathbb{E}}_S(\beta \mid y) = E\,\mathbb{E}(\widetilde{\beta} \mid y)$
    - LSSPCA: $\widehat{\mathbb{E}}_L(\beta \mid y) = E^*\,\mathbb{E}(\widetilde{\beta} \mid y)$
    - DECO: $\widehat{\mathbb{E}}_D(\beta \mid y) = \mathbb{E}(\widetilde{\beta} \mid y)$
    - $\widehat{p}(\gamma_j = 1 \mid y)$ for each technique as of Section 2.3

where Σ is a p x p simulated correlation matrix, in which $\sigma_{ii} = 1$ while $\sigma_{ij} = \rho$ for $i \neq j$, and β^* is a p dimensional vector, containing the true simulated coefficients of each variable. Further, E is the loadings matrix for classical SPCA, while E^* indicates the loadings matrix for LSSPCA, as described in Section 2.2.2.

Different settings for LSSPCA and SPCA can be selected, however for the purpose of illustration in the following simulations SPCA is constrained to have a $k = 2$ cardinality, meaning that every eigenvector is to have two non-zero entries. Further, LSSPCA thresholding parameter is set at 0.35, meaning that entries of eigenvectors are to be set to zero whenever their value is below this boundary. Finally, for both techniques in the $p > n$ case only up to 100 principal components are computed.

Although such specification leads to a very high degree of sparsity, it allows to avoid the very extreme instance in which only one non zero entry is available per eigenvector. This is done in order to prevent the algorithm reaching models that are too parsimonious, and hence not including relevant variables. The aim for such techniques is to either include in the final model only the right predictors, or one of their supersets and hence work as a pre-screening methodology.

To present a benchmark of what the ideal Bayesian model selection would be without taking into account computational efficiency, in the presented examples "Enumeration" and "MCMC" information are also included.

By "Enumeration" we refer to the practice of fully enumerating all the 2^p models, which returns the exact Bayesian solution. Given the high computational effort associated with this method, for $p \geq 50$ only models up to 10 variables are going to be considered. It is hence important to point out how for $p \geq 50$ "Enumeration" is not a full enumeration, but only a partial one. Nevertheless, the result are to suffer very few from this short-cut, since by doing so we are really not considering models that are expected to have a posterior probability very close to zero.

By "MCMC" we refer to Bayesian model selection for linear models using non-local priors. The algorithm uses a Gibbs scheme and can handle $p \gg n$ cases. Although this scheme adopts a MCMC idea, the solutions should be very close to the fully enumerated ones. It is hence a very good proxy, particularly for the $p > n$ instance, in which even for our relatively cheap simulations, enumerating all the 2^p models becomes infeasible, as mentioned above.

Finally, for DECO we simply set $r = 1$ for this set of simulations. Throughout this chapter, we employ a Zellner Prior with parameter $\tau = n$ for the coefficients, a Binomial$(1/p)$ Prior on the model indicator space, and a IG(0.01,0.01) Prior for the residual variance.

3.2 R functions

For the purpose of this chapter, we created two functions, which are described in this section. For reproducibility, the R code is in Appendix A, and the routines are taken from the R package `mombf` version 1.8.0.

---

truebma — *Transforms* $p(\widetilde{\gamma}_j = 1 \mid y)$ *to* $p(\gamma_j = 1 \mid y)$.

Description

Transforms the BMA output of functions "postModeOrtho" or "postModeBlockDiag" contained in `mombf` package, obtaining marginal inclusion probabilities and Bayesian model averaging in terms of the original variables as explained in Section 2.3. This is used whenever a PCA related technique is employed.

Usage

```
truebma(p,pm,rot)
```

Arguments

`p` number of variables
`pm` output object of functions "postModeOrtho" or "postModeBlockDiag"

`rot` loadings matrix

Values

`bma` dataframe containing $\widehat{p}(\gamma_j = 1 \mid y)$ and $\widehat{\mathbb{E}}(\beta \mid y)$ for $j = 1, \ldots, p$

---

simOrthofit *Reproduces Algorithm 1*

Description

Performs Bayesian variable selection and averaging simulations, reproducing Algorithm 1. It incorporates in its routine function `truebma`, "MCMC" and "Enumeration".

Usage

```
simOrthofit(rho,p,n, ridge=1, nsim, enum=F, nvars, phi=1, nonzeros=3,
coef=c(5,.75,1),priorCoef=zellnerprior(tau=n),priorDelta=modelbinomprior(p=1/p)
,priorVar= igprior(0.01,0.01))
```

Arguments

`rho` correlation between original variables.
`p` number of variables.
`n` number of observations.
`ridge` specify value for ridge refinement parameter in DECO. Default is 1.
`nsim` number of simulations.
`enum` whether to include completely enumerated model selection. Default is FALSE.
`nvars` when enum=TRUE, up to how many 2^{nvars} variables the algorithm should consider.
`phi` initial value of `phi`. Default is 1.
`nonzeros` number of non zero valued predictors. Default is 3.
`coef` vector of non zero coefficients. Default is c(.5,.75,1).
`priorCoef` Prior distribution for the coefficients. Default is zellnerprior(tau=n).
`priorDelta` Prior on model indicator space. Default is modelbinomprior(p=1/p).
`priorVar` Prior on residual variance. Default is igprior(0.01,0.01).

Values

`RawMSE` Raw Data MSE for $j = 1, \ldots, p$.
`SPMSE` SPCA MSE for $j = 1, \ldots, p$.
`LSMSE` LSSPCA MSE for $j = 1, \ldots, p$.
`decoMSE` DECO MSE for $j = 1, \ldots, p$.
`fullMSE` Full Enumeration MSE for $j = 1, \ldots, p$.
`msMSE` MCMC MSE for $j = 1, \ldots, p$.
`decot` $\widehat{\mathbb{E}}_D(\beta \mid y)$ for $j = 1, \ldots, p$ for every simulation.
`decotpp` dataframe containing $\widehat{p}(\gamma_j = 1 \mid y)$ for $j = 1, \ldots, p$ for every simulation − DECO.
`Rawt` $\widehat{\mathbb{E}}_R(\beta \mid y)$ for $j = 1, \ldots, p$ for every simulation.
`Rawtpp` dataframe containing $\widehat{p}(\gamma_j = 1 \mid y)_R$ for $j = 1, \ldots, p$ for every simulation − Raw Data.

`SPt` $\widehat{\mathbb{E}}_S(\beta \mid y)$ for $j = 1, \ldots, p$ for every simulation.
`SPtpp` dataframe containing $\widehat{p}(\gamma_j = 1 \mid y)_S$ for $j = 1, \ldots, p$ for every simulation.
`LSt` $\widehat{\mathbb{E}}_L(\beta \mid y)$ for $j = 1, \ldots, p$ for every simulation.
`LStpp` dataframe containing $\widehat{p}(\gamma_j = 1 \mid y)$ for $j = 1, \ldots, p$ for every simulation − LSSPCA.
`mst` MCMC's $\widehat{\mathbb{E}}_M(\beta \mid y)$ for $j = 1, \ldots, p$ for every simulation.
`mstpp` dataframe containing MCMC's $\widehat{p}(\gamma_j = 1 \mid y)$ for $j = 1, \ldots, p$ for every simulation − MCMC.
`fullt` Full Enumeration's $\mathbb{E}(\beta \mid y)$ for $j = 1, \ldots, p$ for every simulation.
`fulltpp` dataframe containing Full Enumeration's $p(\gamma_j = 1 \mid y)$ for $j = 1, \ldots, p$ for every simulation − Full Enumeration.
`modDeco` Model Size and correctly included predictors at every simulation − DECO
`modRaw` Model Size and correctly included predictors at every simulation − Raw Data
`modSP` Model Size and correctly included predictors at every simulation − SPCA
`modLS` Model Size and correctly included predictors at every simulation − LSSPCA
`modms` Model Size and correctly included predictors at every simulation − MCMC
`modfull` Model Size and correctly included predictors at every simulation − Full Enumeration

3.3 Simulations

In this section, we run Algorithm 2 on 50 independent simulated datasets for every situation considered. While n is fixed to 100 throughout all the experiments, $p = (10\ ,\ 50\ ,\ 200)$ to provide evidences of how the framework behaves as the situation becomes more and more challenging.

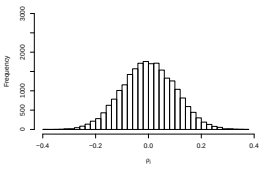
With similar purpose, different values of $\rho = (0.0\ ,\ 0.2\ ,\ 0.5\ ,\ 0.7)$ are considered for every value of p. Further, the true regression coefficients are set to be $\beta_j^* = 0$ for $j = 1, \ldots, p-3$, $\beta_{p-2}^* = 0.50$, $\beta_{p-1}^* = 0.75$, $\beta_p^* = 1$.

In order to give a better representation of what actual values of sample correlation $\hat{\rho}_{ij}$ our algorithm had to face in the orthogonalisation phase, the sampling distributions of each $\hat{\rho}$ are included. As of Figure 3.1, it is clear how such distributions tend to be symmetric and bell-shaped around the data-generating true value of ρ. Nonetheless, value as far as $\rho \pm 0.2$ are observed for all four sampling distributions.

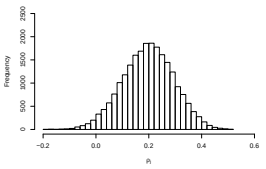
For all methods, we report the model mode $\widehat{\gamma}^*$ (i.e. the mean number of predictors selected), the connected mean number of True Positive (TP) and mean number of False Positive (FP), the average marginal inclusion probability $\widehat{p}(\gamma_j = 1 \mid y)$ for $j = p_1 : p-3, p-2, p-1, p$. Moreover, for every method the Mean Square Error (MSE) is obtained, and its ratio to the benchmark (Enumeration) MSE is reported, denoted as "MSE Ratio". Precisely:

$$MSERatio_j = \frac{\sum_{i=1}^{nsim}[\widehat{\mathbb{E}}_i(\beta \mid y) - \beta_j^*]^2}{\sum_{i=1}^{nsim}[\mathbb{E}_i(\beta \mid y) - \beta_j^*]^2} \tag{3.1}$$

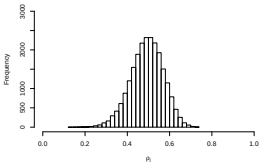
is reported for each group of coefficients $j = p_1 : p-3, p-2, p-1, p$. To provide intuitive and compact presentation of the results, several figures are presented in this section. Nevertheless, precise tables containing all the information are provided in Appendix B.



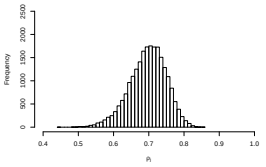
(A) $\rho = 0.0$



(B) $\rho = 0.2$



(C) $\rho = 0.5$



(D) $\rho = 0.7$

FIGURE 3.1: Sampling Distributions of $\hat{\rho}$ for several values of ρ

Firstly, as of Table 3.1 we comment on the posterior expected number of variables in the model $\widehat{\gamma}^*$ for all possible combinations of ρ and p. Notice how Enumeration is missing for $p = 200$: this is due to the very high computational cost of fully enumerating 2^{200} models. Instead, as it will come clear in a few lines, MCMC takes its place as benchmark, since it mirrors very closely the results of Enumeration.

		$\rho = 0.00$			$\rho = 0.20$			$\rho = 0.50$			$\rho = 0.70$		
p		10	50	200	10	50	200	10	50	200	10	50	200
Enumeration	$\widehat{\gamma}^*$	3.00	3.00		3.00	3.00		2.80	3.00		2.80	2.70	
	TP	3.00	3.00		3.00	3.00		2.80	2.90		2.70	2.60	
	FP	0.00	0.00		0.00	0.00		0.00	0.10		0.10	0.10	
MCMC	$\widehat{\gamma}^*$	3.00	3.30	3.00	3.00	3.20	4.00	2.80	3.30	4.00	2.80	2.80	3.66
	TP	3.00	3.00	3.00	3.00	3.00	2.66	2.80	2.80	2.66	2.70	2.60	2.00
	FP	0.00	0.30	0.00	0.00	0.20	1.44	0.00	0.50	1.44	0.10	0.20	1.66
Raw Data	$\widehat{\gamma}^*$	3.54	4.90	1.00	9.90	1.10	1.00	10.00	1.00	1.00	10.00	1.00	1.00
	TP	2.84	2.80	0.00	3.00	0.50	0.00	3.00	0.00	0.00	3.00	0.00	0.00
	FP	0.70	2.10	1.00	6.90	0.60	1.00	7.00	1.00	1.00	7.00	1.00	1.00
DECO	$\widehat{\gamma}^*$	2.46	2.60	1.00	2.60	2.60	1.00	2.10	1.30	1.00	2.10	1.20	1.00
	TP	2.32	2.50	1.00	2.50	2.60	0.33	2.00	1.20	0.00	1.80	0.90	0.00
	FP	0.14	0.10	0.00	0.10	0.00	0.77	0.10	0.10	1.00	0.30	0.30	1.00
SPCA	$\widehat{\gamma}^*$	5.68	7.88	0.00	9.90	3.50	0.00	10.00	50.00	0.00	10.00	50.00	0.00
	TP	2.84	2.62	0.00	3.00	0.98	0.00	3.00	3.00	0.00	3.00	3.00	0.00
	FP	2.84	5.6	0.00	6.90	2.52	0.00	7.00	47.00	0.00	7.00	47.00	0.00
LSSPCA	$\widehat{\gamma}^*$	5.36	6.44	0.00	9.70	7.98	0.00	9.06	3.66	4.66	9.26	3.28	8.00
	TP	2.80	2.52	0.00	3.00	1.30	0.00	2.98	0.68	0.33	3.00	0.46	0.66
	FP	2.56	3.92	0.00	6.70	6.68	0.00	6.08	2.98	4.33	6.26	2.82	7.33

TABLE 3.1: Model Size, True Positives and False Positives, nsim=50

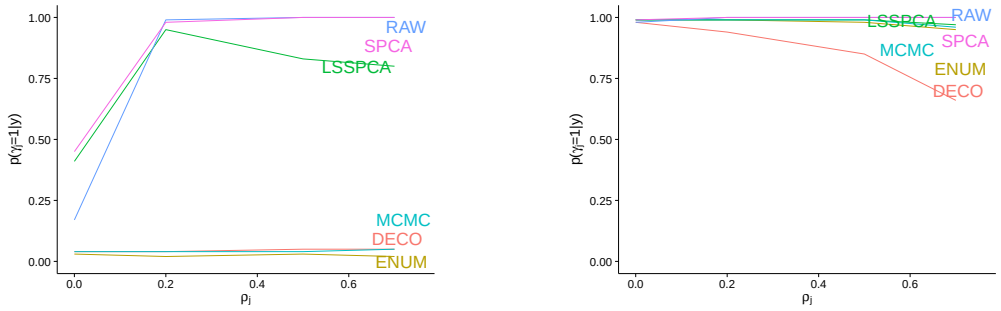
Commenting on the output contained in the Table, both Enumeration and MCMC manage to mostly select a very parsimonious model of size 3, in which the 3 included variables are in fact the right ones. This is true for every value of p and ρ. Regarding Raw Data, although it does quite well for $p = 10$, with at least selecting a superset of the right variables for increasing values of ρ, it selects a completely wrong model for $p = 50$,

by constantly selecting 1 spurious variable only; this behaviour is put to an extreme for $p = 200$. Interestingly, this situation arises for DECO as well in the $p = 200$ instance.

This is obviously suspicious, since it is not intuitive that by increasing p the model selected drastically goes from a superset of variables to something close to the Null model. Nevertheless, this problem is fully addressed Section 3.4, and it is hence not considered until then.

Finally, SPCA and LSSPCA tend to select supersets of the right model for $p = 10$ and $p = 50$, while it systematically selects the Null model for $p = 200$

We now analyse in depth what happens in terms of both marginal inclusion probabilities and MSE Ratio for every value of p, starting by considering the instance in which $p = 10$.



(A) $\beta_j^* = 0.00$ (B) $\beta_j^* = 0.50$

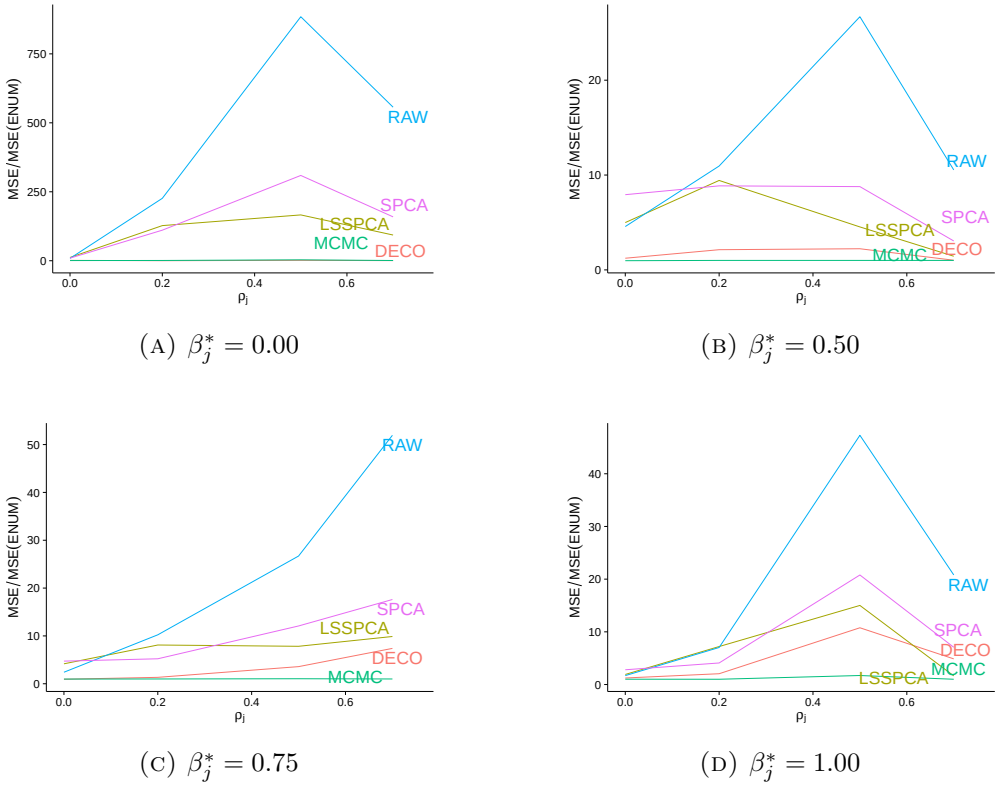
(C) $\beta^* = 0.75$ (D) $\beta_j^* = 1.00$

FIGURE 3.2: Average marginal inclusion probability vs ρ for each β_j^*, $p = 10$

As of Figure 3.2, in terms of $\widehat{p}(\gamma_j = 1 \mid y)$ DECO mirrors very well both Enumeration and MCMC for $\beta_j^* = 0$, even for very challenging values of ρ; on the other hand, SPCA and LSSPCA fail to avoid including irrelevant variables. This comes without surprise given the cardinality constraints one needs to fix for sparse PCA methods, as explained in Section 2.3.1. Particularly, since every column of the loadings matrix E is constrained to have $k = 2$ non-zero entries, selecting one component results in including two x_j's in the final model. This reasoning particularly makes sense when bonded with the results of Table 3.1, in which SPCA and LSSPCA tend to select models that include approximately 6 variables, of which 3 are the right ones. Finally, Raw Data also allows the inclusion of some spurious variables in the model.

The remaining three panels show that all techniques manage to recognise the truly significant three variables, although with lesser power as ρ increases.

This feature is particularly pronounced for DECO, with a minimum $\widehat{p}(\gamma_j = 1 \mid y)$ of 0.44 for $\beta_j^* = 0.50$. Nonetheless, this drawback of DECO is balanced by the MSE Ratios, as depicted in Figure 3.3, in which DECO has much lower MSE's, especially for high values of ρ. On the other hands, Raw Data deteriorates very quickly as ρ increases, with MSE values that are as high as 884 times higher than the Enumeration technique. Clearly, already for $p = 10$ it can be stated that pretending $X^T X$ to be orthogonal (i.e. Raw Data) is not a good idea, while DECO outperforms all other techniques, since it does not rely on loadings matrices that are constrained to have a certain cardinality in order to impose sparsity.



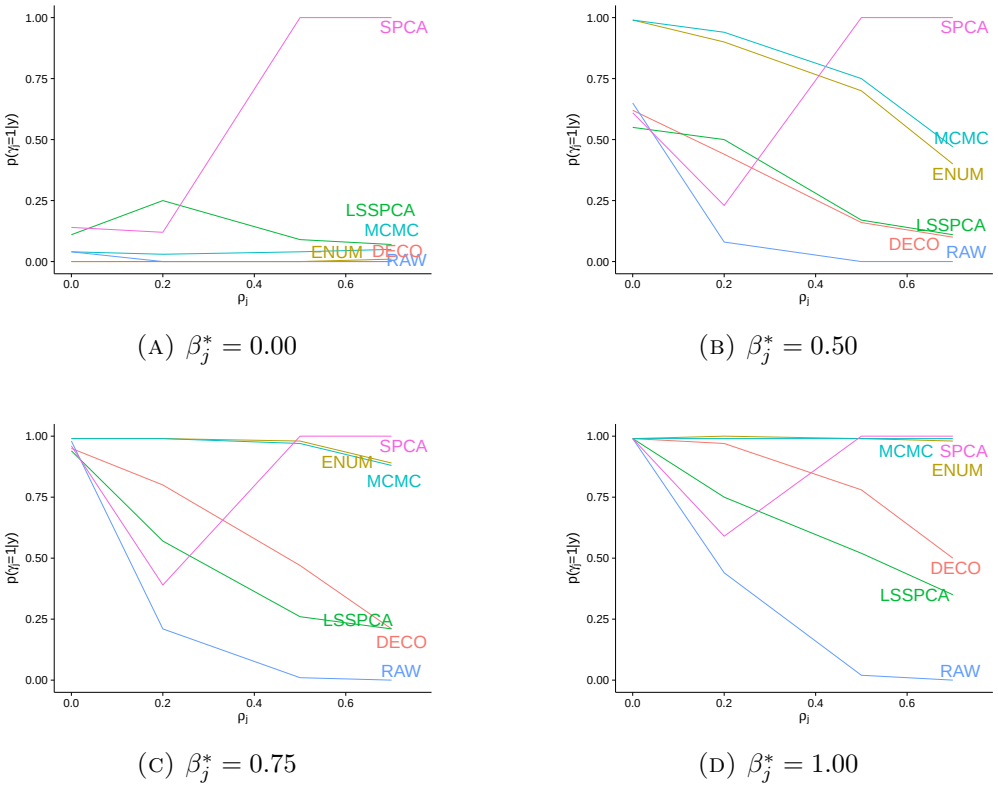
(A) $\beta_j^* = 0.00$ (B) $\beta_j^* = 0.50$

(C) $\beta_j^* = 0.75$ (D) $\beta_j^* = 1.00$

FIGURE 3.3: MSE Ratio vs ρ for each β_j^*, $p = 10$

As we now consider the case in which 40 spurious variables are added to the simulations (i.e. $p = 50$), Figure 3.4 shows how the marginal inclusion probabilities $\widehat{p}(\gamma_j = 1 \mid y)$ for all methods keep performing very well when $\beta_j^* = 0$, except for SPCA that tends to include on average spurious variables for values of $\rho = 0.5$ and higher; interestingly, SPCA shows a very similar pattern for all the three remaining panels.

Regarding all the other techniques, they all demonstrate the tendency of losing detection power for increasing values of ρ, which is expected. Unsurprisingly, both Enumeration and MCMC start to slightly suffer from this as well, which is a natural consequence of adding 40 extra spurious variables. Nonetheless, it is clear how DECO keeps outperforming LSSPCA and Raw Data for all values of β_j^*, confirming what has been said for $p = 10$ on DECO starting to be a clear winner.



(A) $\beta_j^* = 0.00$ (B) $\beta_j^* = 0.50$

(C) $\beta_j^* = 0.75$ (D) $\beta_j^* = 1.00$

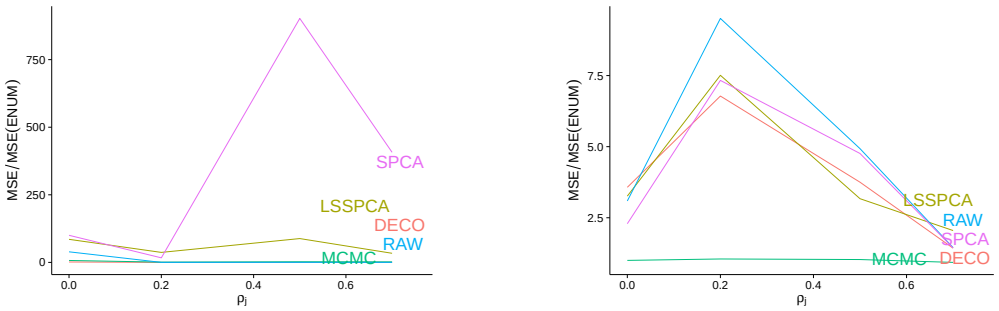
FIGURE 3.4: Average marginal inclusion probability vs ρ for each β_j^*, $p = 50$

As we take into consideration the MSE Ratio in the analysis, this statement becomes even clearer, as Figure 3.5 shows a maximal MSE ratio of 36.30 throughout all the four panels for DECO, while the remaining techniques reach much higher values, such as a maximum value of 902.43 for SPCA.

Finally, we consider the case $p = 200$. As of Figure 3.6, all methods succeed in avoiding the inclusion of spurious variables. However, as of panels (B), (C), and (D), all the considered methods also fail to include the truly relevant variables, even for contained values of ρ. This comes without surprise, given what has been said with regards to Table 3.1. Once again, this issue is going to be addressed in the next Section.

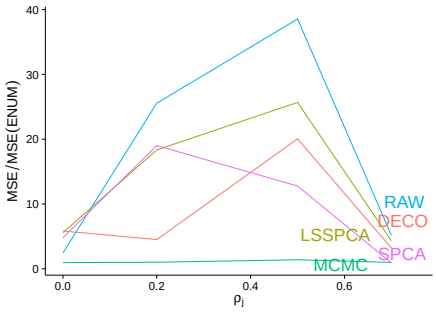
Finally, Figure 3.7 shows us the MSE Ratios, in which the benchmark is provided by MCMC for the computational reason explained previously. Unsurprisingly, such values are high for all values of β_j^* with respect to all techniques, DECO included. This is connected to what has been said with regard to Table 3.1, in which it has been detected that by increasing p the selected model jumps drastically from reasonable choices to the null one, and hence the output of this Figure shall also be reconsidered once the issue is addressed in Section 3.4.

In conclusion, we can summarise what has been done in this simulated examples' section as follows. MCMC has proved to be a valid benchmark, mirroring what happens by fully enumerating the models both in terms of marginal inclusion probabilities and MSE. It was hence reasonable to adopt MCMC as benchmark for the expensive $p = 200$ instance.

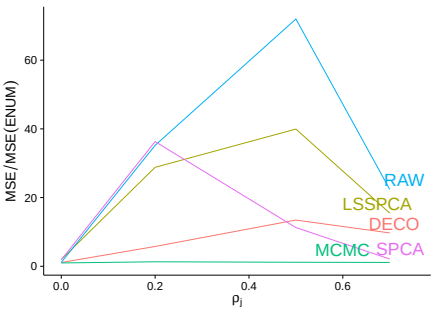


(A) $\beta_j^* = 0.00$

(B) $\beta_j^* = 0.50$

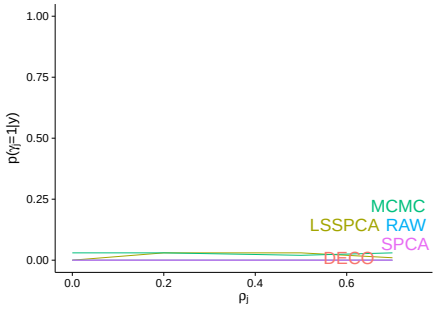


(C) $\beta_j^* = 0.75$

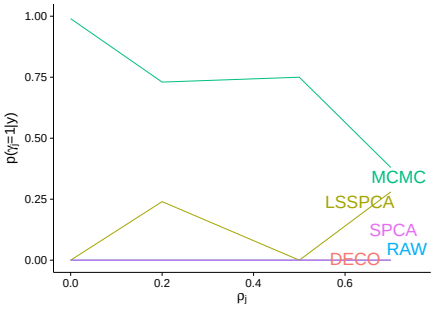


(D) $\beta_j^* = 1.00$

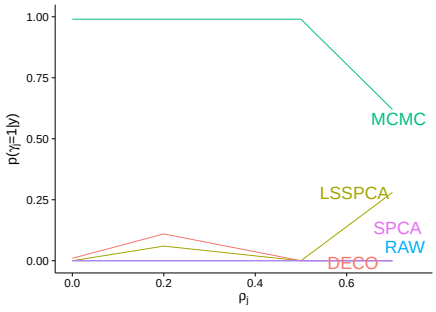
FIGURE 3.5: MSE Ratio vs ρ for each β_j^*, p=50



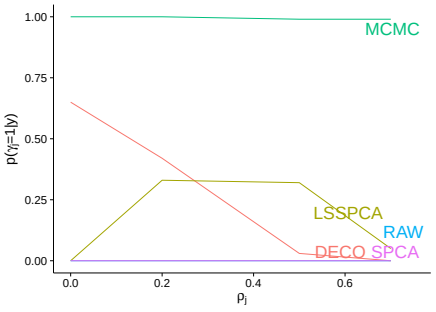
(A) $\beta_j^* = 0.00$



(B) $\beta_j^* = 0.50$

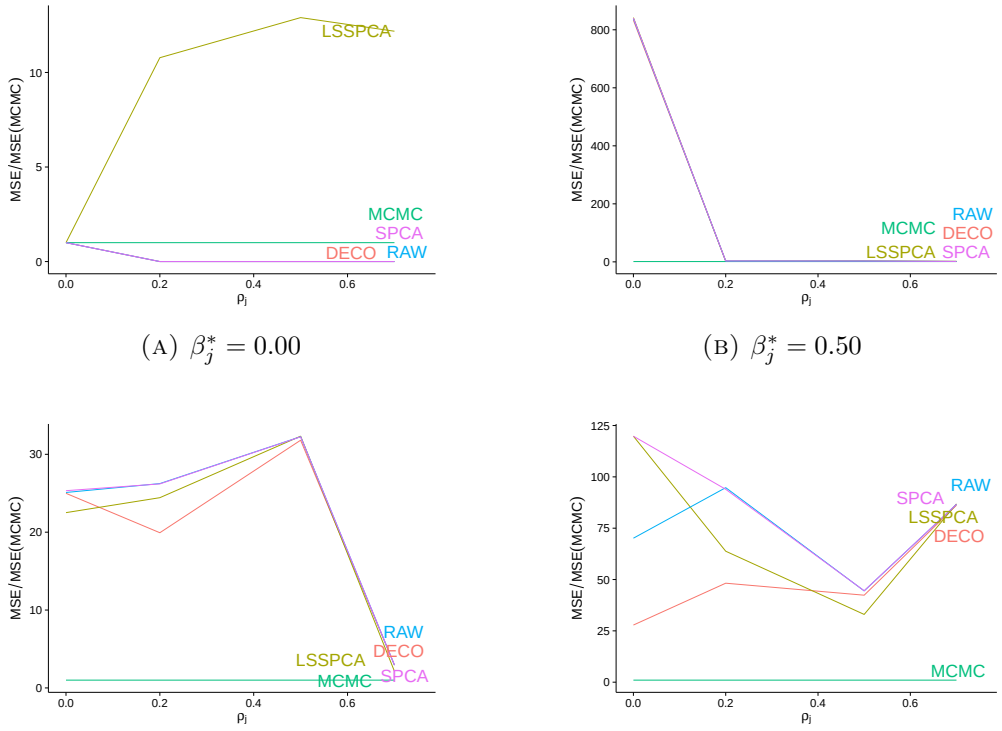


(C) $\beta_j^* = 0.75$



(D) $\beta_j^* = 1.00$

FIGURE 3.6: Average marginal inclusion probability vs ρ for each β_j^*, $p = 200$



(A) $\beta_j^* = 0.00$ (B) $\beta_j^* = 0.50$

(C) $\beta_j^* = 0.75$ (D) $\beta_j^* = 1.00$

FIGURE 3.7: MSE Ratio vs ρ for each β_j^*, p=200

Next, we have demonstrated how blindly running Bayesian model selection on the Raw Data is not a good idea, given the fast deterioration when $X^T X$ is even slightly far away from the truly orthogonal case. On the other hand, LSSPCA has returned better results, outperforming the more traditional SPCA, which is definitely justified by the attempt of the method to return something closer to orthogonal principal components. Nevertheless, both techniques managed to select a superset of the right variables in most cases, and it is hence reasonable to think of PCA related methods as a good pre-screening tool. That is, one could potentially use LSSPCA to pre-screen relevant variables and subsequently perform full enumeration of models that include only the pre-screened variables. However, DECO is the one to really shine, with not only very good marginal inclusion probabilities and MSE throughout all the simulations, but with very parsimonious models as shown with respect to Table 3.1.

In the next Section we study why the performance of some methods suffers such a significant drop for $p = 200$, and we attempt to provide a reasonable fix.

3.4 U-Score fix

As discussed in Section 3.3, there is a clear issue that can be appreciated in Table 3.1 where the selected model by Raw Data and DECO jumps from a a reasonable size to the drastic null model as p and ρ grow.

A connected behaviour that is particularly obvious for the Raw Data method is the one related to the marginal inclusion probabilities (Tables B.1, B.2, B.3 and B.4) in which $\widehat{p}(\gamma_j = 1 \mid y) = 1.00$ when $p = 10$ for all j's, while $\widehat{p}(\gamma_j = 1 \mid y) = 0.00$ when $p = 50$ for all j's. Although it is expected that Raw Data would not carry out a satisfying model selection, this situation is very bizarre and requires a better understanding of what is going on. Notice how this tendency is also present for SPCA and LSSPCA, although with less emphasis, which makes sense, as the resulting Gram matrix of PCA related methods tend to be closer to a diagonal matrix, especially for the LSSPCA technique.

In order to fully understand and solve this problem, it is worth diving a little deeper in how the framework of Papaspiliopoulos & Rossell selects the posterior mode and hence the best model. This is done taking advantage of the Bayes factor

$$\frac{p(y \mid \gamma, \tau, \omega)}{p(y \mid \emptyset, \tau, \omega)} = \left(\frac{l_\phi + y^T y}{l_\phi + y^T y - \frac{\tau}{\tau+1} u(y, \gamma)} \right)^{\frac{a_\phi + n}{2}} \frac{1}{(1+\tau)^{|\gamma|/2}} \tag{3.2}$$

of a given model γ under examination with respect to the null model $\emptyset$. However, since the methodology was designed to cope with orthogonal and block diagonal design, whenever this is not true the decomposition in (2.7) $u(y, \gamma) = \sum_{j=1}^{p} u(y, \gamma[j])$ can give rise to a negative denominator , which is clearly inadequate as it would result in negative posterior probabilities.

To avoid this limitation and other numerical overflows, the authors introduced a fix in how the u-scores are computed. Specifically, whenever $y^T y < \dfrac{\tau}{\tau+1} \sum_{j=1}^{p} u(y, \gamma[j])$ occurs, the u-scores are re-normalised as follows:

$$\widehat{u}(y, \gamma) = u(y, \gamma) \left(\frac{y^T y}{\sum_{j=1}^{p} u(y, \gamma[j])} \right) \leq y^T y \tag{3.3}$$

Although the fix makes sure that the algorithm doesn't return negative probabilities it has one big issue, namely that the values of the u-scores gets diluted, with a degree that increases with p. This causes the u-scores to be very small, and consequently the Bayesian model selection algorithm tends to select the Null model as the posterior mode, which explains the drop in sensitivity to detect truly active variables that we observed in our simulations.

As this fix can only be potentially triggered when the Gram matrix $X^T X$ is not diagonal, it affects Raw Data for any p and DECO for $p > n$.

In this work, we attempt to give an alternative fix: whenever the denominator of the Bayes Factor in Equation (3.2) is negative, the following is computed:

$$\widetilde{u}(y, \gamma) = min\left(u(y, \gamma), y^T y\right) \tag{3.4}$$

which implies that the first few u-scores (which should contain the truly active covariates) remain unchanged, until their cumulative sum multiplied by $\tau/(\tau+1)$ is equal to $y^T y$. The modified u-score for the remaining covariates is thus effectively set to 0, which intuitively means that adding more variables doesn't improve the model fit.

In order to fully appreciate the problem of Equation (3.3), and how the proposed fix can do better, we present the u-scores in a very simple example with both $p = 10$ and $p = 50$ when $\rho = 0.50$, for the Raw Data method. In order to guarantee homogeneous results, the $p = 10$ example uses a subset of the variables in the $p = 50$ and contains the 3 truly active predictors.

x_j	$u(y,\gamma)$	$\widehat{u}(y,\gamma)$	$\widetilde{u}(y,\gamma)$
1	114.61	33.94	0
2	133.55	39.55	0
3	124.84	36.97	0
4	96.47	28.57	0
5	133.37	39.50	0
6	157.15	46.54	0
7	97.08	28.75	0
8	114.18	33.82	0
9	192.29	56.95	192.29
10	269.21	79.73	269.21

TABLE 3.2: original u-scores, fix (3.3) and fix (3.4) for $p = 10$

x_j	$u(y,\gamma)$	$\widehat{u}(y,\gamma)$	$\widetilde{u}(y,\gamma)$
1	114.61	8.28	0
2	133.55	9.65	0
3	124.84	9.02	0
4	96.47	6.97	0
5	133.37	9.64	0
6	157.15	11.36	0
7	97.08	7.02	0
8	114.18	8.25	0
9	192.29	13.90	192.29
10	269.21	19.46	269.21

TABLE 3.3: original u-scores, fix (3.3) and fix (3.4) for $p = 50$

Tables 3.2 and 3.3 show the results for the $p = 10$ and $p = 50$ instances respectively, in which the true predictors are denoted in boldface. Column $u(y,\gamma)$ includes the values of the u-scores before applying any fix, column $\widehat{u}(y,\gamma)$ shows how the values change after re-normalising using fix (3.3), and column $\widetilde{u}(y,\gamma)$ includes the values of the u-scores adopting the proposed truncation fix (3.4), which reduces sensitivity to how many predictors are included in the analysis, since the ones exceeding $y^T y$ are to be truncated, and these are typically the truly inactive variables.

Given this result, it is worth going again over the simulated examples using $\widetilde{u}(y,\gamma)$ instead of $\widehat{u}(y,\gamma)$ as we did in Section 3.3. For the purpose of clarity, we will focus solely on the problematic $p = 200$ case, by re-running all the experiments for Raw Data and DECO. Table 3.4 includes all the results, in which both the old version of the algorithm adopting fix and the new version, nicknamed SST("Sum of Squares Truncation") adopting the fix $\widetilde{u}(y,\gamma)$ are reported.

As of Table 3.4, both the benchmark MCMC and the old faulty Raw Data and DECO are included, in order to provided a fast and coherent comparison with the novel Raw Data-SST and DECO-SST. Further, since we are now focusing solely on Raw Data and DECO, it is interesting to explore how the results change by modifying the ridge refinement parameter for DECO-SST. Hence, not only $r = 1$ is considered, but also $r = 10$.

Commenting on the results, the advantage of SST based methods is obvious, with MSE Ratios that are up to 10 times smaller for both Raw Data and DECO. Similarly, the marginal inclusion probabilities seem reliable, and manage to do a great job in mirroring

	β_j	MCMC	Raw Data	Raw Data-SST	DECO	DECO-SST($r=1$)	DECO-SST($r=10$)
				$\rho = 0.00$			
$p(\gamma_j = 1 \mid y)$	0.00	0.02	0.00	0.03	0.00	0.05	0.03
	0.50	0.93	0.00	0.80	0.02	0.92	0.84
	0.75	0.99	0.06	1.00	0.22	1.00	1.00
	1.00	1.00	0.51	1.00	0.60	1.00	1.00
MSE Ratio	0.00	1.00	0.00	7.22	0.00	7.77	7.66
	0.50	1.00	8.64	2.37	8.64	1.12	3.82
	0.75	1.00	50.56	2.22	35.03	1.95	2.11
	1.00	1.00	32.06	1.51	16.33	1.14	1.98
				$\rho = 0.20$			
$p(\gamma_j = 1 \mid y)$	0.00	0.02	0.00	0.00	0.00	0.05	0.01
	0.50	0.86	0.00	0.54	0.01	0.96	0.74
	0.75	0.99	0.00	0.94	0.16	1.00	1.00
	1.00	0.99	0.00	1.00	0.60	1.00	1.00
MSE Ratio	0.00	1.00	0.00	4.54	0.00	7.42	6.62
	0.50	1.00	7.27	6.35	7.18	0.93	5.43
	0.75	1.00	34.32	9.55	29.18	1.98	1.85
	1.00	1.00	55.03	4.03	18.53	1.74	1.50
				$\rho = 0.50$			
$p(\gamma_j = 1 \mid y)$	0.00	0.03	0.00	0.00	0.00	0.06	0.01
	0.50	0.70	0.00	0.10	0.00	0.84	0.64
	0.75	0.97	0.00	0.84	0.01	1.00	0.96
	1.00	0.99	0.00	0.94	0.29	1.00	1.00
MSE Ratio	0.00	1.00	0.00	1.06	0.00	8.01	5.64
	0.50	1.00	3.72	4.91	2.77	0.62	3.21
	0.75	1.00	13.68	16.36	18.79	1.37	4.87
	1.00	1.00	39.59	17.92	23.88	1.01	2.85
				$\rho = 0.70$			
$p(\gamma_j = 1 \mid y)$	0.00	0.02	0.00	0.00	0.00	0.06	0.01
	0.50	0.42	0.00	0.18	0.00	0.84	0.52
	0.75	0.78	0.00	0.48	0.00	1.00	0.88
	1.00	0.99	0.00	0.98	0.10	1.00	1.00
MSE Ratio	0.00	1.00	0.00	2.28	0.00	6.39	3.98
	0.50	1.00	1.55	3.22	1.56	0.49	1.92
	0.75	1.00	5.20	7.72	4.15	0.33	2.85
	1.00	1.00	42.34	30.29	28.56	1.10	1.96

TABLE 3.4: Comparing $\widehat{u}(y,\gamma)$ and $\widetilde{u}(y,\gamma)$ based methods, $p = 200$

the benchmark very closely for contained values of ρ. Unsurprisingly, DECO-SST constantly outperforms Raw Data-SST, although the quality of the output has increased substantially with respect to Raw Data.

On the other hand, for values of $\rho = 0.50$ and higher, it is worth observing how Raw Data-SST loses power in detecting the truly active variables, with diminished marginal inclusion probabilities, while DECO-SST has higher figures for the marginal probabilities of including the right variables. However, this is done at the expense of selecting more spurious variables than Raw Data-SST. This feature can be appreciated both in Table 3.4 higher MSE Ratios for spurious variables in DECO-SST, and in Table 3.5, in which the average model size denoted by $\widehat{\gamma}^*$ is considerably higher in DECO-SST than it is for Raw Data-SST. This drawback of DECO-SST is nonetheless mitigated by the lower MSE Ratios shown in Table 3.4 for the truly active covariates, which are way lower when compared to Raw Data-SST for high values of ρ.

Table 3.5 shows the average selected model size, true and false positives. Regarding the choice of ridge refinement, for this specific set of data $r = 1$ provides a less parsimonious model, with a larger model size compared to $r = 10$. The true positives are almost exactly the same in the two different choices of r, thus $r = 1$ is more aggressive in

		$\rho = 0.00$	$\rho = 0.20$	$\rho = 0.50$	$\rho = 0.70$
MCMC	$\widehat{\gamma}^*$	4.50	4.28	4.44	3.86
	TP	2.94	2.90	2.76	2.28
	FP	1.56	1.38	1.68	1.58
Raw Data	$\widehat{\gamma}^*$	1.00	1.00	1.00	1.00
	TP	0.46	0.00	0.00	0.00
	FP	0.64	1.00	1.00	1.00
Raw Data-SST	$\widehat{\gamma}^*$	9.44	3.44	2.02	2.00
	TP	2.80	2.48	1.88	1.64
	FP	6.65	0.96	0.14	0.36
DECO	$\widehat{\gamma}^*$	1.06	1.00	1.00	1.00
	TP	0.78	0.16	0.00	0.00
	FP	0.28	0.84	1.00	1.00
DECO-SST($r = 1$)	$\widehat{\gamma}^*$	13.34	13.00	15.10	16.36
	TP	2.90	2.96	2.84	2.84
	FP	10.44	10.04	12.26	13.52
DECO-SST($r = 10$)	$\widehat{\gamma}^*$	9.90	6.10	5.4	4.74
	TP	2.84	2.74	2.6	2.40
	FP	7.06	3.36	2.8	2.34

TABLE 3.5: Model Size $\widehat{\gamma}^*$, True Positives and False Positives, highlighting the benefits of SST ($p = 200$).

selecting variables and returns a solution that is a superset of the more conservative $r = 10$. Relative to the MCMC solution both $r = 1$ and $r = 10$ return slightly larger model sizes and false positive rates.

3.5 Gene expression data

We assess the performance of the proposed framework in gene expression data. [Calon *et al.* , 2012] used mice experiments to detect 172 genes potentially related to the gene family TGFB, which has been proved to be related to colon cancer progression in an independent data set of $n = 262$ human patients.

In this section we will hence apply the best performing SST methods to the dataset, and try to identify which of these $p = 172$ genes are related to TGFB, as the correct detection of such genes would improve the understanding of colon cancer progression. Recent work [Rossell & Telesca, 2015] has been done on this dataset, in which genes related to various cancer types (ESM1, GAS1, HIC1, CILP, ARL4C, PCGF2), TGFB regulators (FAM89B) or AOC3 which is used to alleviate certain cancer symptoms were detected as important in predicting TGFB.

The authors used MCMC to explore the model space, which we will use as a benchmark for comparison with our orthogonalisation-based methods. As an exploratory analysis, Figure 3.8 includes a histogram of the the off-diagonal elements of the correlation matrix $\widehat{\Sigma}$ of the predictors. Although the histogram seems to be approximately centered around zero, $\rho \in (-0.44, 0.84)$, providing a challenging environment for our algorithm.

We run the algorithm using the MCMC to provide a baseline, Raw Data, Raw Data-SST, and DECO for different values of ridge refinement $r = (1, 10, 100)$. However, since $r = 1$ and Raw Data selected the null model, their results are going to be omitted, and

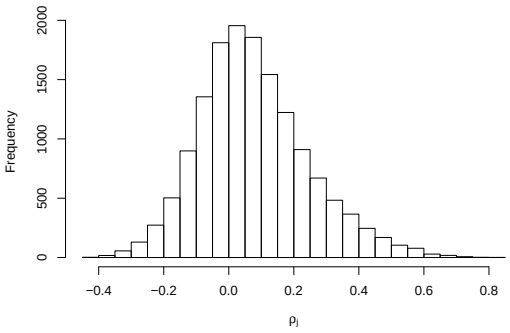


FIGURE 3.8: Distribution of ρ_j in the Gene expression data

instead we will focus on $r = 10$ and $r = 100$, and on Raw Data-SST. All the following results employ a Zellner Prior for the coefficients with parameter $\tau = n$, a Binomial$(1/p)$ Prior on the model indicator space, and a IG(0.01,0.01) Prior for the residual variance. Note that DECO-SST is not taken into consideration for this specific example, since as $p < n$ it is equivalent to regular DECO.

We start by presenting Table 3.6, in which all the reported methods tend to select a model with model size $\widehat{\gamma}^* \in 4, 6$, except for DECO ($r = 10$) that selects a bigger model of size 15. Nevertheless, all the four methods include very similar amounts of genes related to cancer. We have checked this by searching on specialised websites analysing gene expression for the selected ones, and checking whether they are related to colon cancer in some way, or more generally to tumor cell proliferation. This is important, in order to give a solid check from a biological point of view, augmenting the quality of our results.

	$\widehat{\gamma}^*$	Cancer Related
MCMC	6	5
Raw Data-SST	3	3
DECO($r = 10$)	15	6
DECO($r = 100$)	4	4

TABLE 3.6: Amount of variables selected, and how many of them are related to cancer.

Further, it is interesting to check how many of the selected genes are in common throughout the different techniques and with which marginal inclusion probability. In order to do so, Table 3.7 includes all genes that are selected at least in one method, sorted from the more to the less important according to the MCMC solution. For each method, the marginal inclusion probabilities are denoted in boldface if the connected gene has been included in the model.

Commenting on the results of Table 3.7, the first thing to mention is that out of the 6 genes selected by MCMC, only 3 are selected by some of the other methods. Next, it is worth observing how DECO ($r = 10$) selects a less parsimonious, yet a superset of DECO ($r = 100$), confirming the idea introduced on ridge refinement in the previous section. Further, all the extra genes in DECO ($r = 10$) are from a biological point of view not known to be related to cancer according to CCIB Gene Enrichment Profiler, suggesting that they may not be related to TGFB.

Gene	MCMC	Raw Data-SST	DECO ($r = 10$)	DECO ($r = 100$)
GAS1	**0.99**	**1.00**	**1.00**	0.00
CILP	**0.98**	0.00	0.00	0.00
HIC1	**0.92**	0.00	**1.00**	**1.00**
ESM1	**0.87**	0.00	0.00	0.00
IGFBP3	**0.71**	0.00	0.00	0.00
HG-U133B	0.40	1.00	0.00	0.00
SEMA7A	**0.28**	0.00	0.00	0.00
HG-U133A	0.17	0.00	**1.00**	0.00
EGR2	0.04	0.00	**1.00**	0.00
SPSB1	0.04	0.00	**1.00**	0.00
LOC100507263	0.04	0.00	**1.00**	0.00
RASL12	0.02	0.00	**1.00**	0.00
NUAK1	0.02	**1.00**	**1.00**	**1.00**
TIMP3	0.01	**1.00**	**1.00**	**1.00**
INHBA	0.00	0.00	**1.00**	0.00
TNFAIP6	0.00	0.00	**1.00**	0.00
COL10A1	0.00	0.00	**1.00**	0.00

TABLE 3.7: Genes selected as predictors for TGFB using several methods

Further, gene GAS1 is selected in all 4 considered methods, making it the strongest candidate in our analysis. Similarly, NUAK1 and TIMP3 are selected by all the methods except MCMC; both genes have a high expression in tumoral tissue always according to CCIB Gene Enrichment Profiler. Specifically, NUAK1 shows high expression in tumors such as the salivary gland cancer, colon cancer, kidney cancer, and ovary cancer while TIMP3 shows high expression in glioblastoma, melanoma, and breast cancer ER-. Except for Raw Data-SST (which is meant to be the less reliable anyway) all the remaining techniques also select HIC1: this gene works as growth regulator and tumor repressor, which is definitely appealing for our purpose.

Finally, SEMA7A gene is selected by MCMC method only, with a lower marginal inclusion probability of 0.28 compared to the remaining genes selected by the method. It is hence reasonable to see this gene not being selected by the other methods, that are trying to approximate the best solution we can obtain (i.e. the MCMC solution). More generally, as the marginal inclusion probability of a gene is higher in MCMC, more methods tend to detect it as relevant during model selection.

4. Conclusion

We showed how combining orthogonalisation methods with the framework of Papaspiliopoulos & Rossell returns a powerful tool for scalable Bayesian model selection, providing a potential basis upon which to expand the framework for a more general input X^TX matrix, applicable also in the $p > n$ context.

Additionally to exploring several orthogonalisation methods, as a novel contribution we proposed a fix for the u-scores dilution problem that significantly improved the performance of the framework when the Gram matrix X^TX is far from orthogonal.

When combined with DECO, the resulting method delivered promising results even when $p > n$ in simulations and in the gene expression data. The developed framework in this work has proved to select parsimonious models at a low computational complexity (all examples run in a few seconds using a purely R implementation), without requiring pre-screening of covariates, albeit the selected model does not always coincide and often is slightly larger than the exact Bayesian solution.

Future interesting research work include exploring more advanced strategies to approximate integrated likelihoods via diagonal and block-diagonal approximations, obvious venues for the latter are to consider alternative fixes to the u-scores or clustering techniques to group blocks of highly-correlated variables. Such clustering strategies can also be potentially used in combination with DECO when $p > n$, as then the correlation between transformed variables $\widetilde{X}$ are no longer exactly 0. Finally, here we have focused on studying computational approximations to a fully Bayesian solution based on a convenient prior formulation. It would be hence interesting to compare various other Bayesian and frequentist methods in terms of their model selection abilities.

We also remark that in this thesis we assume that the immediate output of our framework is the final selected model, alternatively one might easily extend it as a tool to pre-screen variables or guide model search. For example, considering the full sequence of models returned by the Coolblock algorithm of Papaspiliopoulos & Rossell (2016) instead of simply focusing on its posterior mode.

A. Appendix: R Code

The R code for the "truebma" function is given below.

```
truebma<-function(p=p,pm,rot){
  require(mombf)
  phi=pm$phi #phi and phi_pp
  margpp <- double(p) #initialize vector of margpp
  pm$bma[,1] #p(gammatilde=1|y)
  for (i in 1:p) {
    df=as.data.frame(which(rot[i,]!=0,arr.ind = F))
    df[,1]
    tilde=prod(1-pm$bma[c(df[,1]),1])
    tr<-tilde* phi[,2]
    margpp[i] <- int.simpson2(phi[,1],  tr, equi=FALSE, method="CSR")
  }
  margpp=1-margpp #P(gamma=1|y)
  beta=rot%*%as.matrix(pm$bma[,2]) #beta
  bma=as.data.frame(cbind(margpp,beta)) ; colnames(bma) <- c("margpp", "beta")
  return(bma)
}
```

The R code for the "simOrthofit" function is given below.

```
simOrthofit<-function(rho,p,n,ridge=1, nsim, enum=F, nvars, phi=1, nonzeros=3,
    coef=c(.5,.75,1),priorCoef=zellnerprior(tau=n),priorDelta=modelbinomprior(p
    =1/p),priorVar=igprior(0.01,0.01)){
  require(nsprcomp)
  require(mvtnorm)
  require(spca)
  require(tcltk)
  require(mombf)
  require(Metrics)
  require(stringi)
  p2=p-2
  p1=p-1
  pb <- txtProgressBar(min = 0, max = nsim, style = 3)
  ans <- list(NULL)
  ans$decot <- data.frame(matrix(NA, nrow = p, ncol = nsim))
  ans$decotpp <- data.frame(matrix(NA, nrow = p, ncol = nsim))
  ans$SPt <- data.frame(matrix(NA, nrow = p, ncol = nsim))
  ans$LSt <- data.frame(matrix(NA, nrow = p, ncol = nsim))
  ans$SPtpp <- data.frame(matrix(NA, nrow = p, ncol = nsim))
  ans$LStpp <- data.frame(matrix(NA, nrow = p, ncol = nsim))
  ans$Rawtpp <- data.frame(matrix(NA, nrow = p, ncol = nsim))
  ans$Rawt <- data.frame(matrix(NA, nrow = p, ncol = nsim))
  ans$mstpp <- data.frame(matrix(NA, nrow = p, ncol = nsim))
  ans$mst <- data.frame(matrix(NA, nrow = p, ncol = nsim))
  ans$fulltpp <- data.frame(matrix(NA, nrow = p, ncol = nsim))
  ans$fullt <- data.frame(matrix(NA, nrow = p, ncol = nsim))
  ans$modDeco<- data.frame(matrix(NA, nrow = nsim, ncol = 2))
  colnames(ans$modDeco) <- c("mod.size","correct")
  ans$modRaw <- data.frame(matrix(NA, nrow = nsim, ncol = 2))
  colnames(ans$modRaw) <- c("mod.size","correct")
  ans$modSP <- data.frame(matrix(NA, nrow = nsim, ncol = 2))
  colnames(ans$modSP) <- c("mod.size","correct")
```

```
ans$modLS <- data.frame(matrix(NA, nrow = nsim, ncol = 2))
colnames(ans$modLS) <- c("mod.size","correct")
ans$modSP <- data.frame(matrix(NA, nrow = nsim, ncol = 2))
colnames(ans$modSP) <- c("mod.size","correct")
ans$modms <- data.frame(matrix(NA, nrow = nsim, ncol = 2))
colnames(ans$modms) <- c("mod.size","correct")
ans$modfull <- data.frame(matrix(NA, nrow = nsim, ncol = 2))
colnames(ans$modfull ) <- c("mod.size","correct")
nonzeros=p-nonzeros
th<-c(rep(0,nonzeros),coef)
TH=do.call("cbind", replicate(2, th, simplify = FALSE))
wrap.mse <- function(i, x_hat, x) {
  mse(x_hat[i, ], x[i, ])
}
for (t in 1:nsim) {
Sys.sleep(0.1); setTxtProgressBar(pb, t)
R<-diag(p)
R[upper.tri(R)]<-R[lower.tri(R)]<-rho
x<-scale(rmvnorm(n,mean = rep(0, nrow(R)), sigma = R),center=T,scale=T)
th<-th
y<-x%*% matrix(th,ncol=1)+rnorm(n,sd=sqrt(phi))
#SPCA
comps=if(p>100){100}else{p}
e2<-nsprcomp(x, k = 2, scale. = TRUE,ncomp=comps)
x2<-x%*%e2$rotation
#DECO
  G=x%*%t(x)/p+ridge*diag(n)
  S=svd(G) ; uu<-S$u ; dd=S$d  ; vv<-S$v
  d1<-diag(1/sqrt(dd))
  tx=uu%*%d1%*%t(uu)%*%x
  ty=uu%*%d1%*%t(uu)%*%y
  deco<-postModeOrtho(y=ty,x=tx,priorCoef=priorCoef,priorDelta=priorDelta,
  priorVar=priorVar,bma=TRUE)
  ans$modDeco[t,1]=length(as.numeric(unlist(stri_split(deco$models$modelid
  [1],fixed=',',))))
  v1=length(which(as.numeric(unlist(stri_split(deco$models$modelid[1],fixed
  =',')))==p2))
  v2=length(which(as.numeric(unlist(stri_split(deco$models$modelid[1],fixed
  =',')))==p1))
  v3=length(which(as.numeric(unlist(stri_split(deco$models$modelid[1],fixed
  =',')))==p))
  ans$modDeco[t,2]=sum(v1,v2,v3)
#LSSPCA
blocks=rep(1:comps,1)
bbe1 <- spcabe(t(x)%*%x, nd = comps, thresh = 0.35, unc = F)
x3<-x%*%bbe1$loadings
SP<-postModeBlockDiag(y=y,x=x2,blocks=blocks,priorCoef=priorCoef,priorDelta=
  priorDelta,priorVar=priorVar,bma=TRUE)
    er=list()
    f=(as.numeric(unlist(stri_split((SP$models[which.max(SP$models$pp)
  ,][1,1]),fixed=',',))))
    for(i in 1:length(f)){
      er[i]=list(which(e2$rotation[,f[i]]!=0))}
    ner <- max(sapply(er, length))
    ll <- lapply(er, function(X) {
      c(as.character(X), rep("", times = ner - length(X)))
    })
```

```
    out <- do.call(cbind, ll)
    out=as.data.frame(out)
    ans$modSP[t,1]=sum((unique(unlist(out))) != "")
    cc1=length(which(as.numeric(as.character(unique(unlist(out))))==p)) #
  right ones
    cc2=length(which(as.numeric(as.character(unique(unlist(out))))==p1))
    cc3=length(which(as.numeric(as.character(unique(unlist(out))))==p2))
    ans$modSP[t,2]=sum(cc1,cc2,cc3)
 LS<-postModeBlockDiag(y=y,x=x3,blocks=blocks,priorCoef=priorCoef,priorDelta=
  priorDelta,priorVar=priorVar,bma=TRUE)
     er2=list()
     f2=(as.numeric(unlist(stri_split((LS$models[which.max(LS$models$pp)
  ,][1,1]),fixed=',',))))
     for(i in 1:length(f2)){
       er2[i]=list(which(bbe1$loadings[,f2[i]]!=0))}
     ner2 <- max(sapply(er2, length))
     ll2 <- lapply(er2, function(X) {
       c(as.character(X), rep("", times = ner2 - length(X)))
     })
     out2 <- do.call(cbind, ll2)
     out2=as.data.frame(out2)
     ans$modLS[t,1]=sum((unique(unlist(out2))) != "")
     c1=length(which(as.numeric(as.character(unique(unlist(out2))))==p)) #
  right ones
     c2=length(which(as.numeric(as.character(unique(unlist(out2))))==p1))
     c3=length(which(as.numeric(as.character(unique(unlist(out2))))==p2))
     ans$modLS[t,2]=sum(c1,c2,c3)
XX<-postModeOrtho2(y=y,x=x,priorCoef=priorCoef,priorDelta=priorDelta,priorVar
  =priorVar,bma=TRUE)
  ans$modRaw[t,1]=length(as.numeric(unlist(stri_split(XX$models$modelid[1],
  fixed=',',))))
  vv1=length(which(as.numeric(unlist(stri_split(XX$models$modelid[1],fixed
  =',')))==p2))
  vv2=length(which(as.numeric(unlist(stri_split(XX$models$modelid[1],fixed
  =',')))==p1))
  vv3=length(which(as.numeric(unlist(stri_split(XX$models$modelid[1],fixed
  =',')))==p))
  ans$modRaw[t,2]=sum(vv1,vv2,vv3)
mcmc<-modelSelection(y=y, x=x, enumerate=F,niter=10^2,center=F, scale=F,
  priorCoef=priorCoef, priorDelta=priorDelta, priorVar=priorVar)
  ans$modms[t,1]=length(which(mcmc$coef!=0))
  vvv1=length(which(which(mcmc$coef!=0)==p1))
  vvv2=length(which(which(mcmc$coef!=0)==p2))
  vvv3=length(which(which(mcmc$coef!=0)==p))
  ans$modms[t,2]=sum(vvv1,vvv2,vvv3)
ans$mstpp[t] <-  data.frame(mcmc$margpp)
ans$mst[t] <-  data.frame(mcmc$coef)
ans$decot[t] <- data.frame(deco$bma[,2])
ans$decotpp[t] <-  data.frame(deco$bma[,1])
ans$SPt[t] <- data.frame(truebma(p=p,pm=SP,rot=e2$rotation)[,2])
ans$LSt[t] <- data.frame(truebma(p=p,pm=LS,rot=bbe1$loadings)[,2])
ans$SPtpp[t] <- data.frame(truebma(p=p,pm=SP,rot=e2$rotation)[,1])
ans$LStpp[t] <- data.frame(truebma(p=p,pm=LS,rot=bbe1$loadings)[,1])
ans$Rawtpp[t] <-  data.frame(XX$bma[,1])
ans$Rawt[t] <-  data.frame(XX$bma[,2])
if(enum==TRUE){
```

```
    fully<-modelSelection(y=y, x=x, enumerate=T,maxvars=nvars,center=F, scale=F
    ,priorCoef=priorCoef, priorDelta=priorDelta, priorVar=priorVar)
    ans$fulltpp[t] <-  data.frame(fully$margpp)
    ans$fullt[t] <-  data.frame(fully$coef)
      ans$modfull[t,1]=length(which(fully$coef!=0))
      oo1=length(which(which(fully$coef!=0)==p1))
      oo2=length(which(which(fully$coef!=0)==p2))
      oo3=length(which(which(fully$coef!=0)==p))
      ans$modfull[t,2]=sum(oo1,oo2,oo3)
    }
  }
  ans$RawMSE=lapply(seq_len(nrow(TH)), wrap.mse, x_hat = ans$Rawt, x = TH) ;
    ans$RawMSE=as.data.frame(as.numeric(ans$RawMSE))
  ans$SPMSE=lapply(seq_len(nrow(TH)), wrap.mse, x_hat = ans$SPt, x = TH);
    ans$SPMSE=as.data.frame(as.numeric(ans$SPMSE))
  ans$LSMSE=lapply(seq_len(nrow(TH)), wrap.mse, x_hat = ans$LSt, x = TH) ;
    ans$LSMSE=as.data.frame(as.numeric(ans$LSMSE))
  ans$decoMSE=lapply(seq_len(nrow(TH)), wrap.mse, x_hat = ans$decot, x = TH) ;
    ans$decoMSE=as.data.frame(as.numeric(ans$decoMSE))
  ans$fullMSE=lapply(seq_len(nrow(TH)), wrap.mse, x_hat = ans$fullt, x = TH) ;
    ans$fullMSE=as.data.frame(as.numeric(ans$fullMSE))
  ans$msMSE=lapply(seq_len(nrow(TH)), wrap.mse, x_hat = ans$mst, x = TH);
    ans$msMSE=as.data.frame(as.numeric(ans$msMSE))
  return(ans)
}
```

B. Appendix: Tables

	β_j^*	Enumeration	MCMC	Raw Data	DECO	SPCA	LSSPCA
				p=10			
$p(\gamma_j = 1 \mid y)$	0.00	0.03	0.04	0.17	0.04	0.45	0.41
	0.50	0.99	0.99	0.88	0.90	0.90	0.89
	0.75	0.99	0.99	0.98	0.98	0.99	0.99
	1.00	1.00	1.00	1.00	0.99	1.00	1.00
MSE Ratio	0.00	1.00	0.64	8.62	1.31	9.49	11.85
	0.50	1.00	0.97	4.57	1.23	7.93	4.99
	0.75	1.00	0.98	2.41	0.94	4.73	4.17
	1.00	1.00	1.01	1.69	1.24	2.80	1.91
				p=50			
$p(\gamma_j = 1 \mid y)$	0.00	0.00	0.04	0.04	0.00	0.14	0.11
	0.50	0.99	0.99	0.65	0.62	0.61	0.55
	0.75	0.99	0.99	0.98	0.95	0.96	0.94
	1.00	0.99	0.99	0.99	0.99	0.99	0.99
MSE Ratio	0.00	1.00	7.11	39.11	1.48	100.20	85.28
	0.50	1.00	1.00	3.09	3.57	2.29	3.26
	0.75	1.00	0.95	2.45	5.83	4.80	5.57
	1.00	1.00	0.97	1.13	1.10	1.91	1.99
				p=200			
$p(\gamma_j = 1 \mid y)$	0.00		0.03	0.00	0.00	0.00	0.00
	0.50		0.99	0.00	0.00	0.00	0.00
	0.75		0.99	0.00	0.01	0.00	0.00
	1.00		1.00	0.00	0.65	0.00	0.00
MSE Ratio	0.00		1.00	0.00	0.00	1.03	0.18
	0.50		1.00	55.47	55.19	55.71	55.67
	0.75		1.00	185.52	173.46	159.54	189.79
	1.00		1.00	13.00	10.67	12.71	36.16

TABLE B.1: Algorithm 1 Output for all methods and various values of p, with $n = 100$ and $\rho = 0.00$

	β_j^*	Enumeration	MCMC	Raw Data	DECO	SPCA	LSSPCA
				p=10			
$p(\gamma_j = 1 \mid y)$	0.00	0.02	0.04	0.99	0.04	0.98	0.95
	0.50	0.98	0.95	1.00	0.87	1.00	0.97
	0.75	0.99	0.99	1.00	0.94	1.00	0.99
	1.00	1.00	1.00	1.00	0.98	1.00	1.00
MSE Ratio	0.00	1.00	1.00	226.30	0.16	110.81	127.39
	0.50	1.00	1.00	10.95	2.13	8.86	9.43
	0.75	1.00	1.00	10.23	1.34	5.22	8.10
	1.00	1.00	1.00	7.03	2.05	4.09	7.21
				p=50			
$p(\gamma_j = 1 \mid y)$	0.00	0.03	0.00	0.00	0.12	0.25	0.11
	0.90	0.94	0.08	0.44	0.23	0.50	0.55
	0.99	0.99	0.21	0.80	0.39	0.57	0.94
	1.00	0.99	0.44	0.97	0.59	0.75	0.99
MSE Ratio	1.00	0.96	0.10	0.01	16.83	36.94	85.28
	1.00	1.05	9.51	6.78	7.33	7.51	3.26
	1.00	1.01	25.55	4.51	19.01	18.34	5.57
	1.00	1.31	35.21	5.77	36.30	28.81	1.99
				p=200			
$p(\gamma_j = 1 \mid y)$	0.00		0.03	0.00	0.00	0.00	0.03
	0.50		0.73	0.00	0.00	0.00	0.24
	0.75		0.99	0.00	0.11	0.00	0.06
	1.00		1.00	0.00	0.42	0.00	0.33
MSE Ratio	0.00		1.00	0.00	0.00	0.00	10.78
	0.50		1.00	2.97	2.95	2.98	2.02
	0.75		1.00	26.24	19.93	26.20	24.43
	1.00		1.00	94.74	48.21	93.95	63.79

TABLE B.2: Algorithm 1 Output for all methods and various values of p, with $n = 100$ and $\rho = 0.20$

	β_j^*	Enumeration	MCMC	Raw Data	DECO	SPCA	LSSPCA
				p=10			
$p(\gamma_j = 1 \mid y)$	0.00	0.03	0.04	1.00	0.05	1.00	0.83
	0.50	0.85	0.85	1.00	0.68	1.00	0.99
	0.75	0.98	0.99	1.00	0.85	1.00	0.99
	1.00	0.99	0.99	1.00	0.90	1.00	1.00
MSE Ratio	0.00	1.00	3.25	884.61	1.48	309.21	166.14
	0.50	1.00	1.00	26.69	2.23	8.78	4.53
	0.75	1.00	1.05	26.70	3.58	12.08	7.83
	1.00	1.00	1.71	47.30	10.77	20.79	15.01
				p=50			
$p(\gamma_j = 1 \mid y)$	0.00	0.00	0.04	0.00	0.00	1.00	0.09
	0.50	0.70	0.75	0.00	0.16	1.00	0.17
	0.75	0.98	0.97	0.01	0.47	1.00	0.26
	1.00	0.99	0.99	0.02	0.78	1.00	0.52
MSE Ratio	0.00	1.00	2.31	0.07	0.04	902.43	88.12
	0.50	1.00	1.03	4.93	3.75	4.76	3.17
	0.75	1.00	1.39	38.55	20.05	12.77	25.68
	1.00	1.00	1.17	71.99	13.46	11.27	39.94
				p=200			
$p(\gamma_j = 1 \mid y)$	0.00		0.02	0.00	0.00	0.00	0.03
	0.50		0.75	0.00	0.00	0.00	0.00
	0.75		0.99	0.00	0.00	0.00	0.00
	1.00		0.99	0.00	0.03	0.00	0.32
MSE Ratio	0.00		1.00	0.00	0.00	0.00	12.90
	0.50		1.00	2.73	2.73	2.73	2.74
	0.75		1.00	32.26	31.81	32.25	32.33
	1.00		1.00	44.47	42.37	44.52	33.00

TABLE B.3: Algorithm 1 Output for all methods and various values of p, with $n = 100$ and $\rho = 0.50$

	β_j^*	Enumeration	MCMC	Raw Data	DECO	SPCA	LSSPCA
				p=10			
$p(\gamma_j = 1 \mid y)$	0.00	0.02	0.05	1.00	0.05	1.00	0.80
	0.50	0.59	0.60	1.00	0.44	1.00	0.92
	0.75	0.95	0.96	1.00	0.66	1.00	0.97
	1.00	0.99	0.99	1.00	0.81	1.00	0.99
MSE Ratio	0.00	1.00	0.85	556.71	0.45	159.08	93.23
	0.50	1.00	1.00	10.54	1.02	3.05	1.44
	0.75	1.00	1.00	51.94	7.39	17.61	9.88
	1.00	1.00	1.00	20.80	4.87	7.07	1.64
				p=50			
$p(\gamma_j = 1 \mid y)$	0.00	0.01	0.05	0.00	0.00	1.00	0.07
	0.50	0.40	0.47	0.00	0.10	1.00	0.11
	0.75	0.89	0.88	0.00	0.21	1.00	0.21
	1.00	0.98	0.99	0.00	0.50	1.00	0.35
MSE Ratio	0.00	1.00	1.87	0.03	0.20	407.15	33.65
	0.50	1.00	0.93	1.45	1.43	1.44	2.05
	0.75	1.00	0.98	5.15	3.20	0.90	4.32
	1.00	1.00	1.12	22.38	9.71	2.12	15.48
				p=200			
$p(\gamma_j = 1 \mid y)$	0.00		0.03	0.00	0.00	0.00	0.01
	0.50		0.38	0.00	0.00	0.00	0.28
	0.75		0.62	0.00	0.00	0.00	0.28
	1.00		0.99	0.00	0.00	0.00	0.05
MSE Ratio	0.00		1.00	0.00	0.00	0.00	12.18
	0.50		1.00	1.46	1.46	1.46	1.73
	0.75		1.00	2.93	2.93	2.94	2.13
	1.00		1.00	86.62	86.19	86.65	86.85

TABLE B.4: Algorithm 1 Output for all methods and various values of p, with $n = 100$ and $\rho = 0.70$

Bibliography

Benidis, Konstantinos, Sun, Ying, Babu, Prabhu, & Palomar, Daniel P. 2016. Orthogonal sparse pca and covariance estimation via procrustes reformulation. *arxiv preprint arxiv:1602.03992*.

Bertsimas, Dimitris, King, Angela, Mazumder, Rahul, *et al.* . 2016. Best subset selection via a modern optimization lens. *The annals of statistics*, **44**(2), 813–852.

Calon, Alexandre, Espinet, Elisa, Palomo-Ponce, Sergio, Tauriello, Daniele VF, Iglesias, Mar, Céspedes, María Virtudes, Sevillano, Marta, Nadal, Cristina, Jung, Peter, Zhang, Xiang H-F, *et al.* . 2012. Dependency of colorectal cancer on a tgf-beta-driven program in stromal cells for metastasis initiation. *Cancer cell*, **22**(5), 571–584.

Carvalho, Carlos M, Polson, Nicholas G, & Scott, James G. 2009. Handling sparsity via the horseshoe. *Pages 73–80 of: Aistats*, vol. 5.

Chen, Scott Shaobing, Donoho, David L, & Saunders, Michael A. 2001. Atomic decomposition by basis pursuit. *Siam review*, **43**(1), 129–159.

Chipman, Hugh A, Kolaczyk, Eric D, & McCulloch, Robert E. 1997. Adaptive bayesian wavelet shrinkage. *Journal of the american statistical association*, **92**(440), 1413–1421.

Fan, Jianqing, & Li, Runze. 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the american statistical association*, **96**(456), 1348–1360.

Furnival, George M, & Wilson, Robert W. 2000. Regressions by leaps and bounds. *Technometrics*, **42**(1), 69–79.

Gelfand, Alan E, Hills, Susan E, Racine-Poon, Amy, & Smith, Adrian FM. 1990. Illustration of bayesian inference in normal data models using gibbs sampling. *Journal of the american statistical association*, **85**(412), 972–985.

Geman, Stuart, & Geman, Donald. 1984. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *Ieee transactions on pattern analysis and machine intelligence*, 721–741.

George, Edward I, & McCulloch, Robert E. 1993. Variable selection via gibbs sampling. *Journal of the american statistical association*, **88**(423), 881–889.

Green, Peter J. 1995. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, **82**(4), 711–732.

Hans, Chris, Dobra, Adrian, & West, Mike. 2007. Shotgun stochastic search for large p regression. *Journal of the american statistical association*, **102**(478), 507–516.

Hastings, W Keith. 1970. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, **57**(1), 97–109.

Hotelling, Harold. 1933. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, **24**(6), 417.

Jolliffe, Ian T, Trendafilov, Nickolay T, & Uddin, Mudassir. 2003. A modified principal component technique based on the lasso. *Journal of computational and graphical statistics*, **12**(3), 531–547.

Madigan, David, York, Jeremy, & Allard, Denis. 1995. Bayesian graphical models for discrete data. *International statistical review/revue internationale de statistique*, 215–232.

Mazumder, Rahul, Friedman, Jerome H, & Hastie, Trevor. 2012. Sparsenet: Coordinate descent with nonconvex penalties. *Journal of the american statistical association*.

Merola, Giovanni Maria. 2014. Sparse principal component analysis: a least squares approximation approach. *arxiv preprint arxiv:1406.1381*.

Metropolis, Nicholas, Rosenbluth, Arianna W, Rosenbluth, Marshall N, Teller, Augusta H, & Teller, Edward. 1953. Equation of state calculations by fast computing machines. *The journal of chemical physics*, **21**(6), 1087–1092.

Miller, Alan. 2002. *Subset selection in regression*. CRC Press.

Mitchell, Toby J, & Beauchamp, John J. 1988. Bayesian variable selection in linear regression. *Journal of the american statistical association*, **83**(404), 1023–1032.

Moghaddam, Baback, Weiss, Yair, & Avidan, Shai. 2005. Spectral bounds for sparse pca: Exact and greedy algorithms. *Pages 915–922 of: Advances in neural information processing systems*.

Natarajan, Balas Kausik. 1995. Sparse approximate solutions to linear systems. *Siam journal on computing*, **24**(2), 227–234.

Papaspiliopoulos, Omiros, & Rossell, David. 2016. Scalable bayesian variable selection and model averaging under block orthogonal design. *arxiv preprint arxiv:1606.03749*.

Pearson, Karl. 1901. Liii. on lines and planes of closest fit to systems of points in space. *The london, edinburgh, and dublin philosophical magazine and journal of science*, **2**(11), 559–572.

Rossell, David, & Telesca, Donatello. 2015. Non-local priors for high-dimensional estimation. *Journal of the american statistical association*, 1–33.

Tibshirani, Robert. 1996. Regression shrinkage and selection via the lasso. *Journal of the royal statistical society. series b (methodological)*, 267–288.

Tipping, Michael E. 2001. Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, **1**(Jun), 211–244.

Wang, Xiangyu, Dunson, David, & Leng, Chenlei. 2016. Decorrelated feature space partitioning for distributed sparse regression. *arxiv preprint arxiv:1602.02575*.

Zhang, Cun-Hui. 2010. Nearly unbiased variable selection under minimax concave penalty. *The annals of statistics*, 894–942.

Zou, Hui, Hastie, Trevor, & Tibshirani, Robert. 2006. Sparse principal component analysis. *Journal of computational and graphical statistics*, **15**(2), 265–286.