Accelerating Bayesian Synthetic Likelihood with the Graphical Lasso

Ziwen An*,†, David J. Nott‡, and Christopher C. Drovandi*,†

*School of Mathematical Sciences, Queensland University of Technology, Australia

†Australian Research Council Centre of Excellence for Mathematical and Statistics Frontiers

‡Department of Statistics and Applied Probability, National University of Singapore

ziwen.an@hdr.qut.edu.au

May 8, 2017

Abstract

Simulation-based Bayesian inference methods are useful when the statistical model of interest does not possess a computationally tractable likelihood function. One such likelihood-free method is approximate Bayesian computation (ABC), which approximates the likelihood of a carefully chosen summary statistic via model simulation and non-parametric density estimation. ABC is known to suffer a curse of dimensionality with the size of the summary statistic. When the model summary statistic is roughly normally distributed in regions of the parameter space of interest, Bayesian synthetic likelihood (BSL), which uses a normal likelihood approximation for a summary statistic, is a useful method known to be more computationally efficient than ABC. However, BSL requires estimation of the covariance matrix of the summary statistic for each proposed parameter, which requires a large number of simulations to estimate precisely using the sample covariance matrix when the summary statistic is high dimensional. In this paper we propose to use the graphical lasso to provide a sparse estimate of the precision matrix. This approach can estimate the covariance matrix accurately with significantly fewer model simulations. We discuss the non-trivial issue of tuning parameter choice in the context of BSL and demonstrate on several complex applications that our method, which we call BSLasso, provides significant improvements in computational efficiency whilst maintaining the ability to produce similar posterior distributions to BSL.

Keywords: approximate Bayesian computation (ABC), covariance matrix estimation, Markov chain Monte Carlo (MCMC), pseudo-marginal methods, shrinkage estimators

1 Introduction

Across many different disciplines such as genetics, ecology, biology and finance, the growth of datasets and computing power has led to more complex and realistic statistical models being proposed. A critical step in testing and developing these models is to estimate the parameters based on the data collected from the true underlying process. It is important also to quantify the uncertainty in the parameter estimates. The Bayesian framework provides a principled framework to perform this task. However, for many complex models of interest, the likelihood function is computationally intractable, preventing the routine use of standard Bayesian computational methods.

However, a collection of likelihood-free methods has been developed for applications where model simulation is computationally cheap in comparison to likelihood evaluation. The most well-known likelihood-free Bayesian method is approximate Bayesian computation (ABC, see for example Beaumont et al. (2002), Beaumont (2010) and Marin et al. (2012)). For each proposed value of the parameter, ABC may be thought of as approximating the density of a summary statistic believed to be informative about the model parameter using non-parametric density estimation (Blum, 2010). The method requires specification of a distance function that compares the observed and simulated summary statistics, a kernel weighting function and its bandwidth (often referred to as the ABC tolerance). Due to the non-parametric density estimation, ABC scales poorly with the dimension of the summary statistic. Thus practitioners must often resort to dimension reduction methods (Blum et al., 2013), where ultimately information from the full dataset is lost. Further, ABC can involve significant tuning, especially in trying to select a suitable distance function and value for the ABC tolerance.

It is of interest, then, to seek methods that scale better to an increase in the dimension of the summary statistic. When the selected summary statistic has roughly a multivariate normal distribution under the model in parameter regions of interest, the intractable summary statistic likelihood may be replaced with a multivariate normal density (Wood, 2010). When implemented in a Bayesian framework, Price et al. (2017) refer to this as Bayesian synthetic likelihood (BSL). Price et al. (2017) demonstrate that BSL is more computationally efficient and requires significantly less tuning than ABC, at the expense of making a parametric approximation to the summary statistic likelihood. Further, Price et al. (2017) demonstrate empirically that BSL shows robustness to some departure away from normality. For each proposed parameter value in a Bayesian algorithm, such as Markov chain Monte Carlo (MCMC), BSL requires estimating the covariance matrix of the summary statistic. This is done by performing n independent simulations in parallel and computing the sample covariance matrix. It is well known that the sample covariance matrix is unbiased but for small to moderate n it performs poorly as an estimator with respect to a variety of loss functions (see, for example, Ledoit and Wolf (2004)). Thus, when the summary statistic is high dimensional, the value of n must be set large to achieve reasonable performance with BSL, thus making the overall procedure computationally intensive despite the efficiency improvements over ABC.

In this paper we propose to use the graphical lasso (glasso, Friedman et al. (2008)) as an alternative estimator of the covariance matrix required in BSL. The glasso assumes that there is sparsity in the inverse covariance matrix, which implies conditional independence amongst some of the selected summary statistics. Intuitively this seems to be a reasonable assumption for many complex likelihood-free applications. Although the resulting estimator of the covariance matrix is no longer unbiased, it can have a significantly smaller risk for small to moderate values of n. This is important in the context of complex likelihood-free applications, as fewer model simulations may be required to approximate the posterior. The amount of sparsity in the glasso estimator is controlled by a penalty parameter. We demonstrate an approach for selecting this tuning parameter so that reasonable mixing can be ensured in the MCMC algorithm. This is an important and non-trivial part of the contribution of the present work, since the considerations involved in tuning parameter choice are quite different in BSL applications compared to the usual ones when applying the glasso.

This paper is organised as follows. In Section 2 we cover the BSL method. Details of the glasso and how we can use this to accelerate BSL are provided in Section 3. The improvements afforded by our method are demonstrated on several examples in Section 4. We close the paper with a summary and discussion in Section 5.

2 Bayesian Synthetic Likelihood

Assume that the problem of interest is estimating the parameter $\boldsymbol{\theta} \in \Theta \subseteq \mathbb{R}^p$ of a statistical model with a likelihood function $p(\boldsymbol{y}|\boldsymbol{\theta})$ where $\boldsymbol{y} \in \mathsf{Y}$ is the observed data and p is the number of parameters. We assume that $p(\boldsymbol{y}|\boldsymbol{\theta})$ is not tractable to compute pointwise as a function of $\boldsymbol{\theta}$ with $\boldsymbol{y}$ fixed. However, we assume that it is comparatively straightforward to simulate data, $\boldsymbol{x} \sim p(\cdot|\boldsymbol{\theta})$, for a range of $\boldsymbol{\theta}$ that is supported by the data, and we aim to use simulation as a surrogate for likelihood evaluation. In such 'likelihood-free' settings, it is common practice to reduce the full dataset down to a set of summary statistics, $\boldsymbol{s_y} \in \mathsf{S} \subseteq \mathbb{R}^d$, where d is the number of summary statistics and $d \geq p$ (see Blum et al. (2013) for a review of dimension reduction methods in likelihood-free applications). The smaller the value of d, the less computation required to estimate $\boldsymbol{\theta}$. However, information in the full data may be lost by reducing d.

In a Bayesian framework, a prior distribution, $p(\boldsymbol{\theta})$, is placed on $\boldsymbol{\theta}$ and interest is in sampling from the posterior distribution

$$p(\boldsymbol{\theta}|\boldsymbol{s_y}) \propto p(\boldsymbol{s_y}|\boldsymbol{\theta})p(\boldsymbol{\theta}),$$

so that expectations of the form $\mathsf{E}[f(\boldsymbol{\theta})|\boldsymbol{s_y}]$ can be estimated for some integrable function $f(\cdot)$.

Wood (2010) propose to approximate $p(\boldsymbol{s_y}|\boldsymbol{\theta})$ with a multivariate normal density, with the mean, $\boldsymbol{\mu}$, and covariance matrix, $\boldsymbol{\Sigma}$, depending on $\boldsymbol{\theta}$, leading to the synthetic likelihood (SL) approximation

$$p(\boldsymbol{s_y}|\boldsymbol{\theta}) \approx p_A(\boldsymbol{s_y}|\boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{s_y}; \boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{\Sigma}(\boldsymbol{\theta})). \quad (1)$$

There may be several reasons to appeal to the normal distribution. Firstly, the central limit theorem may justify a normal approximation for some summary statistics. Secondly, various one-to-one transformation can be applied to improve the normality assumption. Thirdly, some summary statistics may be chosen on the basis of indirect inference (II). In II (Smith, 1993) parameter estimates or the score of a tractable alternative model that still provides a reasonable description of the data might be considered. Under some mild conditions there is theory that such summaries are asymptotically normal. Of course, the relationship between $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\boldsymbol{\theta}$ will generally be unknown. However, they can be estimated by simulation. If we draw $\boldsymbol{x}_{1:n} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)^\top$, where $\boldsymbol{x}_i \overset{\text{iid}}{\sim} p(\cdot|\boldsymbol{\theta})$ for $i = 1, \ldots, n$, we can calculate the summary statistic for each dataset, $\boldsymbol{s}_{1:n} = (\boldsymbol{s}_1, \ldots, \boldsymbol{s}_n)^\top$, where $\boldsymbol{s}_i$ is the summary statistic for $\boldsymbol{x}_i$, $i = 1, \ldots, n$. Unbiased estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be obtained using the sample of summary statistics

$$\begin{aligned} \boldsymbol{\mu}_n(\boldsymbol{\theta}) &= \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{s}_i, \\ \boldsymbol{\Sigma}_n(\boldsymbol{\theta}) &= \frac{1}{n-1}\sum_{i=1}^{n}(\boldsymbol{s}_i - \boldsymbol{\mu}_n(\boldsymbol{\theta}))(\boldsymbol{s}_i - \boldsymbol{\mu}_n(\boldsymbol{\theta}))^\top. \end{aligned} \tag{2}$$

The estimates in (2) can be plugged into the SL into (1) to estimate the SL as $\mathcal{N}(\boldsymbol{s_y}; \boldsymbol{\mu}_n(\boldsymbol{\theta}), \boldsymbol{\Sigma}_n(\boldsymbol{\theta}))$. Together with the prior, $p(\boldsymbol{\theta})$, this approximate likelihood can be used as a replacement to the intractable $p(\boldsymbol{s_y}|\boldsymbol{\theta})$ in a Bayesian algorithm such as MCMC (see Algorithm 2 in Appendix A). We denote the implied posterior distribution of this method as

$$p_{A,n}(\boldsymbol{\theta}|\boldsymbol{s_y}) \propto p_{A,n}(\boldsymbol{s_y}|\boldsymbol{\theta})p(\boldsymbol{\theta}),$$

where

$$p_{A,n}(\boldsymbol{s_y}|\boldsymbol{\theta}) = \int_{\mathsf{S}^n} \mathcal{N}(\boldsymbol{s_y}; \boldsymbol{\mu}_n(\boldsymbol{\theta}), \boldsymbol{\Sigma}_n(\boldsymbol{\theta})) \left\{ \prod_{i=1}^{n} p(\boldsymbol{s}_i|\boldsymbol{\theta}) \right\} d\boldsymbol{s}_i.$$

As $n \to \infty$ we obtain $p_{A,n}(\boldsymbol{\theta}|\boldsymbol{s_y}) = p_A(\boldsymbol{\theta}|\boldsymbol{s_y}) \propto p_A(\boldsymbol{s_y}|\boldsymbol{\theta})p(\boldsymbol{\theta})$.

Despite the fact that $\boldsymbol{\mu}_n(\boldsymbol{\theta})$ and $\boldsymbol{\Sigma}_n(\boldsymbol{\theta})$ are unbiased estimators, $\mathcal{N}(\boldsymbol{s_y}; \boldsymbol{\mu}_n(\boldsymbol{\theta}), \boldsymbol{\Sigma}_n(\boldsymbol{\theta}))$ is not an unbiased estimator of $\mathcal{N}(\boldsymbol{s_y}; \boldsymbol{\mu}(\boldsymbol{\theta}), \boldsymbol{\Sigma}(\boldsymbol{\theta}))$. Andrieu and Roberts (2009) demonstrate that if a non-negative and unbiased estimator of a likelihood is used in an MCMC algorithm the target distribution is remarkably unchanged. Such approaches are referred to as pseudo-marginal methods. This highlights that the target distribution $p_{A,n}(\boldsymbol{\theta}|\boldsymbol{s_y})$ does indeed depend on n. Price et al. (2017) consider using an unbiased estimator of the multivariate normal density under the assumption that the model summary statistic is exactly normal. They call this method uBSL, which is theoretically unaffected by the value of n. Here we use the vanilla BSL method as in the next subsection we consider an alternative estimator of the covariance matrix, for which it would appear infeasible to construct an unbiased estimator of the multivariate normal density. We do not see this as a major drawback, since Price et al. (2017) demonstrate with substantial empirical evidence that the target distribution of MCMC BSL is remarkably insensitive to n.

Given the insensitivity of the BSL target to n, it is of interest to choose n to maximise computational efficiency. The larger the value of n, the more accurately the SL is estimated, which increases the acceptance rate of the MCMC. However, more computation time is required per iteration. Price et al. (2017) demonstrate that there is a wide range of n values that lead to relatively efficient performance, but values of n too large or too small lead to poor results, due to large computing times per iteration and too small acceptance rate, respectively. Further, it is not surprising that a larger value of n is required as the dimension of the summary statistic is increased. Thus, BSL still involves many model simulations and is computationally very intensive in complex applications with high dimensional summary statistics.

Ong et al. (2016) propose to use a variational Bayes (VB) implementation of BSL (VBSL), which reduces the computing time significantly. However, VB requires the pre-specification of a parametric form for the posterior distribution that depends on several hyperparameters (Ong et al. (2016) consider a multivariate normal approximation of the posterior). Such a parametric approximation may not be reasonable for some applications and furthermore the number of hyperparameters to estimate grows with an increase in the number of parameters, p. In this paper, we consider a different approach to accelerate BSL that does not resort to parametric approximations of the posterior. This involves replacing the unbiased sample covariance matrix with one formed by the glasso, which we describe in the following subsection. We note, however, that our approach could also be used to increase the speed of VBSL even further. We plan to investigate this elsewhere.

3 BSL with the Graphical Lasso

3.1 Graphical Lasso

In this subsection we discuss different estimators for the covariance matrix of the multivariate normal distribution with a mean vector $\boldsymbol{\mu}$ and a covariance matrix of $\boldsymbol{\Sigma}$. For this subsection, we drop the dependence of $\boldsymbol{\Sigma}$ on $\boldsymbol{\theta}$ for notational convenience. Without loss of generality, we may assume that the mean $\boldsymbol{\mu}$ is given by a vector of zeros.

Assume that $\boldsymbol{s}_{1:n}$ is an iid sample of size n from a normal distribution with covariance matrix $\boldsymbol{\Sigma}$. The log-likelihood function for fixed $\boldsymbol{s}_{1:n}$ is given by

$$\log p(\boldsymbol{s}_{1:n}|\boldsymbol{\Sigma}) = K + \log|\boldsymbol{\Sigma}^{-1}| - \operatorname{tr}(\boldsymbol{\Sigma}^{-1}\boldsymbol{S}),$$

where K is a constant independent of $\boldsymbol{\Sigma}$ and $\boldsymbol{S}$ is given by

$$\boldsymbol{S} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{s}_i\boldsymbol{s}_i^\top.$$

The maximum likelihood estimate (MLE) for $\boldsymbol{\Sigma}$ is $\hat{\boldsymbol{\Sigma}} = \boldsymbol{S}$. The MLE is close to being unbiased for moderate n. However, when n is small, the MLE can perform poorly according to a variety of loss functions. In the context of BSL, an estimator of the covariance matrix with high variability may lead also to an SL estimator with high variability, negatively impacting the mixing of the MCMC.

Fortunately, there has been and still is a significant amount of research performed on estimating covariance matrices from small sample sizes. We do not provide a full literature review here but refer to the reader to a review paper by Fan et al. (2016) for a more detailed discussion. In this paper we investigate the computational gains that can be achieved by assuming that the inverse covariance matrix, or precision matrix, $\boldsymbol{\Omega} = \boldsymbol{\Sigma}^{-1}$, is sparse. Denote the (i, j)th element of $\boldsymbol{\Omega}$ as ω_{ij}. From a graphical modelling perspective, if $\omega_{ij} = 0$ for $i \neq j$, it implies that the ith and jth variables are conditionally independent given all the other variables. By making this assumption we can obtain, for relatively small values of n, estimates of the precision matrix, and also the covariance matrix, with much better risk behaviour compared with the MLE. The main idea of this paper is that for a given value of n, the number of model simulations, the SL can be estimated with a lower variance. Hence a smaller value of n can be used without sacrificing the mixing properties of the MCMC.

A popular approach for estimating a sparse precision matrix is the glasso (Friedman et al., 2008). This involves maximising the following penalised log-likelihood

$$\log p(\boldsymbol{s}_{1:n}|\boldsymbol{\Omega}) = K + \log|\boldsymbol{\Omega}| - \operatorname{tr}(\boldsymbol{\Omega}\boldsymbol{S}) - \lambda||\boldsymbol{\Omega}||_1, \tag{3}$$

over the space of all positive-definite matrices. Here $||\boldsymbol{\Omega}||_1 = \sum_i \sum_j |\omega_{ij}|$ is the L_1 norm of $\boldsymbol{\Omega}$. There is not an analytical solution to the maximisation of (3) but it is a convex optimisation problem and we use the approach of Friedman et al. (2008) to determine a numerical solution. The glasso also has a Bayesian interpretation. It is the posterior mode when placing a prior on $\boldsymbol{\Omega}$ that has a double exponential prior with parameter λ on the upper triangular elements and an exponential prior on the diagonal components of $\boldsymbol{\Omega}$ (see for example Wang (2012)). The penalty parameter λ controls the sparsity of the estimated precision matrix, with increasing λ leading to more sparsity. For a given dataset $\boldsymbol{s}_{1:n}$, the penalty parameter λ may be chosen using various information criteria and cross validation (see Gao et al. (2012) for example). However, incorporating the glasso estimator in our MCMC BSL algorithm requires different considerations for choosing λ that we discuss in the next subsection.

In (3), we penalise the L_1 norm of the precision matrix, which means all elements are equally penalised. When the summary statistics have significantly different scales, it is natural to standardise before applying the glasso. Suppose s_{ij} is the (i, j)th element in dataset $\boldsymbol{s}_{1:n}$. We define the standardised value of s_{ij} as $\tilde{s}_{ij} = (s_{ij} - \bar{s}_j)/\eta_j$, where $\bar{s}_j$ and η_j are the mean and standard deviation of the jth column of $\boldsymbol{s}_{1:n}$, respectively. Note that the sample covariance matrix of the standardised statistics is simply the sample correlation matrix. Thus, when the glasso is applied in the standardised case, we do not penalise the diagonal elements so that the result of the glasso has all of its diagonal elements equal to one.

3.2 BSLasso

We refer to our BSL procedure of using the glasso to estimate the covariance matrix $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ from the $\boldsymbol{s}_{1:n}$ as BSLasso. Incorporating the corresponding estimate of the SL within an MCMC method creates the MCMC BSLasso algorithm.

The BSLasso approach requires the choice of an additional tuning parameter, λ. Ideally, we wish to choose λ as small as possible so that the results can be expected to be close to that obtained in the standard BSL. However, we also wish to limit the value n to achieve computational gains. We choose the value of λ on the basis that, for a particular value of n, we would like to achieve a reasonable acceptance rate in the MCMC.

In order to explain exactly how we select λ, it is instructive to first consider pseudo-marginal algorithms (Andrieu et al., 2010). In pseudo-marginal methods, an unbiased likelihood estimator is used as a replacement to the exact intractable likelihood in an MCMC algorithm. Remarkably, the MCMC procedure retains the exact posterior as its limiting distribution. However, the stochasticity of the estimator impacts negatively on the mixing relative to the corresponding ideal MCMC algorithm that uses the exact likelihood if it were available. The major issue is that the pseudo-marginal MCMC chain can exhibit significant stickiness when the log-likelihood is grossly overestimated. Under the assumption that the log-likelihood estimator has a normal distribution, Doucet et al. (2015) show that reasonable mixing of pseudo-marginal MCMC can be achieved if the standard deviation of the log-likelihood estimator at a value of $\boldsymbol{\theta}$ with reasonable posterior support is roughly 1.

Even though technically our approach is not a pseudo-marginal method, as the likelihood estimator we use is not an unbiased estimator of the SL if we knew $\boldsymbol{\mu}(\boldsymbol{\theta})$ and $\boldsymbol{\Sigma}(\boldsymbol{\theta})$, we can still use the results from pseudo-marginal methods as guidance for so-called noisy MCMC algorithms (Alquier et al., 2016) more generally. Interestingly, Price et al. (2017)

find that the value of n that leads to the best mixing relative to computing time per iteration is the one that produces an estimated log SL with a standard deviation of roughly 1.5-2. In our experience, it is also the distribution of the log-likelihood estimator, not just the standard deviation, that has an impact on the mixing. When the log-likelihood estimator is roughly normally distributed, we may follow the advice for pseudo-marginal methods and aim for a standard deviation closer to 1. However, we have also observed that the distribution of the log SL estimator can be skew left, which inflates the standard deviation. Underestimated log-likelihood estimates have a milder consequence on the mixing compared to overestimation. In these cases, a standard deviation closer to 2 could still be suitable.

We assume a reasonable point estimate of the parameter $\boldsymbol{\theta}$ has been found. This point estimate may by informed by experts, obtained from previous analyses or from some initial experimentation and pilot runs. Note that we do not necessarily see this as a drawback of our method relative to other similar likelihood-free methods. Both MCMC ABC (Marjoram et al., 2003) and MCMC BSL require the discovery of a suitable starting value as they suffer from very slow convergence when starting from a poor initial value (Lee and Łatuszyński, 2014; Price et al., 2017).

Our approach for selecting the penalty is shown in Algorithm 1. The user needs to select a value of n that will subsequently be used in the MCMC BSLasso algorithm. The value of n could be chosen based on computational considerations and relative to the dimension of the summary statistic, d. It is possible to test a collection of n values in an efficient manner. Denote the largest tested n value as $n_{\max}$. We generate simulations $\boldsymbol{s}_{1:n_{\max}} \overset{\text{iid}}{\sim} p(\boldsymbol{s}|\boldsymbol{\theta})$ and for any $n < n_{\max}$ we take a random sample of size n from $\boldsymbol{s}_{1:n_{\max}}$ without replacement. Then, the glasso method is used to estimate the covariance matrix across a path of penalty values $\lambda_1, \ldots, \lambda_K$ that is pre-specified. In practice, the user can firstly choose a coarse grid of penalty values to identify a particular region of penalty values to focus on. This process can be iterated a few times if necessary. It is important to note that the same set of simulations can be used for each λ value. This process is repeated M times to produce a set of log SL estimates $\{\log p_A^{n,\lambda_k}(\boldsymbol{s_y}|\boldsymbol{\theta})\}_{m=1}^{M}$ for each value of λ_k, $k = 1, \ldots, K$. By inspecting the distribution of log SL estimates a suitable value of the standard deviation σ is chosen that is expected to lead to reasonable mixing in MCMC BSLasso. Then we are required to find the corresponding λ that produces an estimated standard deviation close to σ. We use a value of $\sigma = 1.5$ in this paper as we find it works well in the examples, see Section 5 for more discussion on selecting λ. If a lower value of λ is desired, then a larger value of n is required. The approach is summarised in Algorithm 1.

There are other standard approaches to select the penalty in glasso when performing a data analysis, such as the Bayesian information criterion (BIC) and cross validation (CV). We use BIC and CV error (see Gao et al. (2012) for the relevant formulae) to determine the penalty values and compare them with those from Algorithm 1. We adopt the same number of iterations, $M = 300$, and average the BIC and CV error for different values of n and λ. Let $\text{BIC}(\lambda, n|\boldsymbol{\theta})$ and $\text{CV}(\lambda, n|\boldsymbol{\theta})$ be the averaged BIC and 10-fold CV error, where $\boldsymbol{\theta}$ is a point estimate with reasonable posterior support. The optimal values of λ from BIC and CV are chosen such that $\lambda_{\text{BIC}} = \operatorname*{argmin}_{\lambda} \text{BIC}(\lambda, n|\boldsymbol{\theta})$, $\lambda_{\text{CV}} = \operatorname*{argmin}_{\lambda} \text{CV}(\lambda, n|\boldsymbol{\theta})$.

```
Input  : Parameter value with reasonable posterior support, $\boldsymbol{\theta}_0$, the number of
         simulations that will be performed per iteration in the MCMC BSLasso,
         $n$, a sequence of potential penalty values, $\lambda_1, \ldots, \lambda_K$, and the number of
         log SL estimates obtained, $M$.
Output: A penalty parameter $\lambda$ to be used in the MCMC BSLasso algorithm.
for $m = 1$ to $M$ do
    Generate a collection of summary statistics $\boldsymbol{s}_{1:n} \overset{\text{iid}}{\sim} p(\boldsymbol{s}|\boldsymbol{\theta}_0)$
    Compute the sample mean $\boldsymbol{\mu}_n(\boldsymbol{\theta}_0) = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{s}_i$
    Use the glasso to obtain $\boldsymbol{\Sigma}_n^{\lambda_k}(\boldsymbol{\theta}_0)$ for each $k = 1, \ldots, K$ based on the same
      simulations $\{\boldsymbol{s}_i\}_{i=1}^{n}$
    Use the estimated mean $\boldsymbol{\mu}_n(\boldsymbol{\theta}_0)$ and the collection of covariance matrix
      estimates $\{\boldsymbol{\Sigma}_n^{\lambda_k}(\boldsymbol{\theta}_0)\}_{k=1}^{K}$ to estimate the log SL $\log\{p_A^{n,\lambda_k}(\boldsymbol{s}_{\boldsymbol{y}}|\boldsymbol{\theta}_0)\}_{k=1}^{K}$
end
By inspecting the distribution of log SL estimates choose a suitable value of the
  standard deviation $\sigma$ that will achieve reasonable mixing in the MCMC BSLasso
  algorithm and return the corresponding $\lambda_k$ that produces an estimated standard
  deviation close to $\sigma$
```

Algorithm 1: Procedure to select the penalty value λ for use within MCMC BSLasso

4 Examples

Below we consider applications of varying complexity. The examples shed light on the considerations that need to be made when using BSLasso. Further, the examples investigate the quality of approximation and the computational gains that can be achieved by BSLasso relative to standard BSL.

We attempt to avoid the impact of the MCMC proposal distribution on the comparisons. This is quite difficult to do since the approximate posterior obtained can depend on the penalty parameter λ and also potentially on the choice of n. Therefore, for each individual approximation to the posterior, we perform pilot runs to obtain what we believe to be an efficient proposal distribution. The examples that we consider are low dimensional in terms of the number of the parameters so we use a multivariate normal random walk proposal with a covariance matrix that is estimated from the pilot runs. To measure the computational efficiency we empirically compute the effective sample size (ESS) from the MCMC output, divide it by the total number of model simulations used and then multiply the result by a constant large scalar to increase the magnitude of the numbers to facilitate comparison.

We run the MCMC algorithms for a sufficiently large number of iterations T so that the results are not dominated by Monte Carlo error. Since we assume a reasonable point estimate of the parameter has been found, we do not use any burn-in.

We acknowledge that even though we find the glasso method to be fast, it is still slower than computing the sample covariance matrix. However, in complex applications where the computation is dominated by model simulations, for example when the summary statistic is high-dimensional and/or model simulation is even moderately expensive, the additional time introduced by glasso will be small. According to Friedman et al. (2008), the speed of the glasso decreases with an increase in the dimension of the covariance matrix and also with a decrease in the amount of sparsity that is being assumed. However, we find that the glasso method runs quickly in the examples of this paper.

Below we consider two simulated examples and a real data example. For illustration purposes, the proportion of partial correlations below particular thresholds in the 'true' covariance matrix by performing many model simulations at either the true parameter value and one with high posterior support are shown in Table 1. The second example has a very high degree of potential sparsity. The first example also has an inverse covariance matrix that may be approximated well with a sparse version. The third example only has some potential sparsity.

Table 1: The proportion of partial correlations below certain thresholds in the true covariance matrices for the three examples in this paper.

Example/Threshold	0.01	0.02	0.05	0.10	0.20	0.50	0.75
Example 1: MA(2) model	0.81	0.81	0.85	0.92	0.96	0.96	0.96
Example 2: Cell biolgy model	0.83	0.94	0.98	$\approx$1	1	1	1
Example 3: Multivariate g-and-k model	0.03	0.06	0.17	0.38	0.61	0.83	0.89

We display the posteriors with different combinations of n and λ in the three individual examples. We suggest that the posterior distributions are mainly impacted by the λ value, however we cannot eliminate the possibility of n influencing the results. Therefore, in Appendix B, we show that BSLasso is very insensitive to n by testing different n values with fixed λ in the three examples.

4.1 MA(2) Example

Firstly we consider a simple example that has a tractable likelihood function. The model in question is a standard MA(2) time series model, and has been considered in other likelihood-free research (see for example Marin et al. (2012)). The process evolves according to the following

$$y_t = z_t + \theta_1 z_{t-1} + \theta_2 z_{t-2}, \text{ for } t = 1, \ldots, L,$$

where $z_t \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ for $t = -1, 0, \ldots, L$. The parameter of interest is $\boldsymbol{\theta} = (\theta_1, \theta_2)^\top$ and the parameter space is restricted to $\Theta \equiv \{\mathbb{R}^2 : -2 < \theta_1 < 2, \theta_1 + \theta_2 > -1, \theta_1 - \theta_2 < 1\}$ to ensure that the time series process is invertible.

We take as the 'observed' data, $\boldsymbol{y} = (y_1, y_2, \ldots, y_L)^\top$, a simulated dataset with $\theta_1 = 0.6$, $\theta_2 = 0.2$ and $L = 50$. We use as the summary statistic the full dataset so that the dimension of the summary statistic is also $L = 50$. Notice that marginally $y_t \sim \mathcal{N}(0, \theta_1^2 + \theta_2^2)$ for $t = 1, \ldots, L$, so that the summary statistics are already on the same scale.

For standard BSL we trial n values of $n = 200$, 250, 300, 500 and 750. Consistent with the extensive empirical evidence provided in Price et al. (2017), we find that the approximate posterior is very insensitive to n. We find that choosing $n = 500$ gives the highest efficiency out of the trialled values of n. Thus, we do our efficiency comparisons of BSLasso with BSL based on $n = 500$. Unsurprisingly, the BSL posterior is very close to the true posterior as the full dataset does indeed have a multivariate normal distribution (results not shown).

We trial BSLasso with $n = 50$, 150, 300 and 500. We first need to determine a suitable penalty parameter λ to use for each value of n. For Algorithm 1 we use $M = 300$ and the true value of $\boldsymbol{\theta}$. For illustration, we use a sequence of log-uniformly distributed values of $e^{-8}, e^{-7.9}, \dots, e^{0.5}$ for λ within Algorithm 1. In practice, a more coarse grid of penalty values can initially be chosen to find a region of λ values to focus on. We find that the distributions of the log-likelihood estimated are roughly symmetric but that there is a slight tendency for outliers in the left tail that may inflate the standard deviation of the log SL (see Section 5 for a discussion). We choose to select λ values that lead to $\sigma \approx 1.5$ for each value of n, noting that the desired value of λ will be smaller for increasing n. The exact values of λ that we use in MCMC BSLasso are shown in Table 2. It is evident that we are able to select a lower value of λ as the value of n is increased. Figure 1 shows the comparison of Algorithm 1 , BIC and CV methods for selecting the penalty value λ. Algorithm 1 and CV give similar values of λ for $n = 50, 150, 300, 500$, while the λ values chosen by the BIC method are slightly larger.

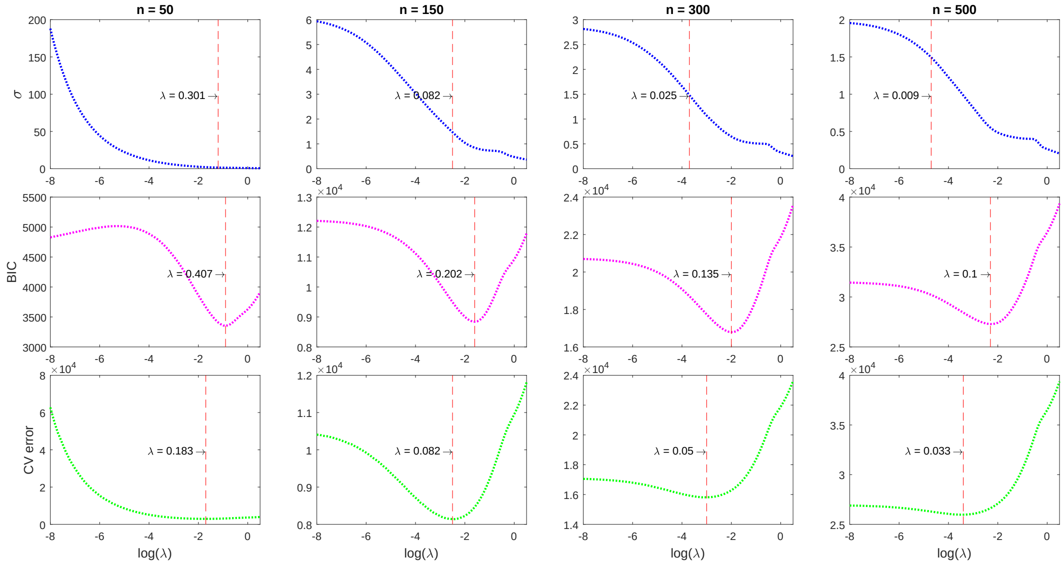

Figure 1: Comparing the methods of selecting the penalty for the MA(2) example. The first row is the standard deviation of the log SL, the second row is the average BIC and the third row is the average CV error.

We first focus on the results for computational efficiency. The normalised ESS values for BSLasso and standard BSL are shown in Table 2. It is clear that introducing the glasso with a suitably large penalty allows us to obtain quite high acceptance rates even for small n. This results in much larger normalised ESS values. Furthermore, it is evident that our strategy for selecting λ seems to be effective as the acceptance rate is similar regardless of the choice of n.

We have demonstrated the relative computational efficiency of BSLasso but have not considered the accuracy. As is evident from Table 2, we are able to decrease the penalty value without sacrificing on the acceptance rate by increasing the value of n. Thus we expect the accuracy (in the sense of closeness to the standard BSL posterior) to improve as n is increased. However, the significant computational gains come from small n, so there is interest in how small we can take n and not lose much accuracy. Figure 2 shows BSLasso posteriors for a select few n values and compares them with the standard BSL posteriors. It is clear when we increase the value of n (hence decrease the value of λ), that the BSLasso posterior gets closer to the standard BSL posterior. The effect of the approximation for larger λ appears to be a posterior with an inflated variance, i.e. conservative. Although the greatest computational gain is obtained with $n = 50$, the quality of the posterior

Table 2: Normalised ESS values and MCMC acceptance rates for standard BSL and BSLasso for the MA(2) example. Also shown are the different combinations of n and λ trialled for BSLasso. The first row corresponds to standard BSL, which does not require a λ value.

n	λ	acc. rate (%)	ESS θ_1	ESS θ_2
500	-	15	39	38
50	0.300	31	243	265
150	0.080	29	162	123
300	0.025	27	89	79
500	0.009	26	67	66

approximation may not be considered reasonable depending on the decisions that might be made on the basis of these results. However, $n = 300$ provides a much more reasonable approximation and the computational gains are still substantial.

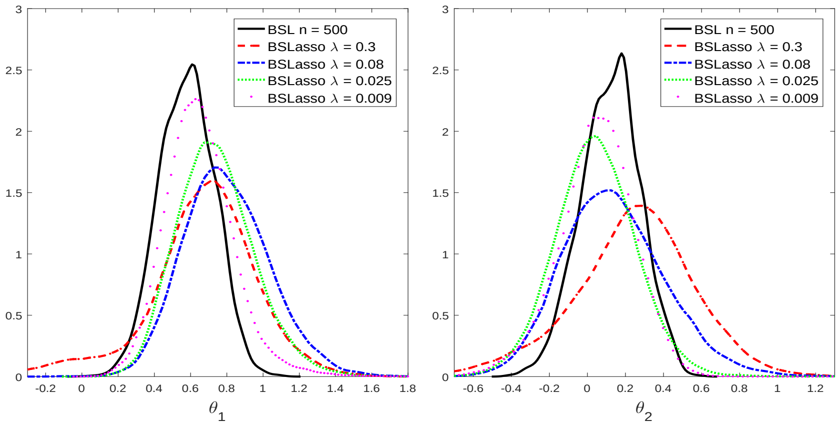

Figure 2: Posteriors for the MA(2) example with standard BSL and BSLasso with various values for λ (based on the values in Table 2).

4.2 Cell Biology Example

We consider an example in collective cell spreading, which has important biological applications in wound healing and skin cancer growth (Swanson et al., 2003; Dale et al., 1994). We follow the study in Price et al. (2017), who developed a simulated version of the real data analysis conducted in Johnston et al. (2014). We briefly describe the experimental set-up in Johnston et al. (2014), but refer the interested reader to their paper for more details. Initially, a population of cells is placed in a dish and a scratch is made down the middle. Images of the cell population are taken every 5 minutes until the cells have re-filled the hole made by the scratch (last image at 12 hours).

Johnston et al. (2014) develop a discrete-time stochastic model to try to explain the evolution of the cell population. A square lattice is constructed where the size of each lattice site is approximately equal to the average diameter of a cell. In each small time step of the simulator, each cell is given an opportunity to move to a neighbouring lattice site (chosen randomly) with probability P_m, with the move aborted if a cell is already present at the proposed lattice site. Similarly, during the small time step, each cell is given an opportunity to give birth and deposit a daughter cell at a neighbouring lattice site (chosen randomly) with probability P_p, where again the proposal is rejected if there is already a cell present at the desired location. The parameters P_m and P_p can be converted into biologically relevant parameters, the cell diffusivity and the cell proliferation rate, via suitable transformations (see Johnston et al. (2014) for more details). We denote the parameter as $\boldsymbol{\theta} = (P_m, P_p)^\top$.

Johnston et al. (2014) consider only 3 of the 144 images (excluding image at time 0) for ABC parameter estimation. The reason for this is two-fold: (1) the image analysis for the observations relies on some manual processing to map the cells onto a square lattice so it is comparable with the simulation model and (2) the dimension of the data is substantially reduced facilitating an ABC analysis with less computational burden. Price et al. (2017) consider a simulation experiment that attempts to utilise the information from all images to see if more precise estimates of $\boldsymbol{\theta}$ can be obtained. The summary statistics developed are the collection of Manhattan distances between adjacent binary matrices and the number of cells present at the end of the experiment (145 statistics), with the former designed to be informative about cell motility and the latter being informative about cell proliferation. Price et al. (2017) demonstrate that BSL with a suitably large value of n is able to accommodate this high-dimensional summary statistic. The aim of our analysis is to see if we can produce similar inferences for $\boldsymbol{\theta}$ with a much smaller value of n. The simulated dataset of Price et al. (2017) uses $P_m = 0.35$, $P_p = 0.001$ and 100 cells at time 0.

Price et al. (2017) find that $n = 5000$ produces the most efficient BSL results (out of the values $n = 2500$, 3750, 5000, 7500 and 10000). We use Algorithm 1 with $M = 300$ to determine the potential penalty values for n values of $500, 1000, 1500$ and 2000. We determine the λ values from $e^{-1.5}, e^{-1.4}, \ldots, e^{3}$ for MCMC BSLasso to correspond with $\sigma \approx 1.5$. These λ values are shown in Table 3. Plots of σ, the BIC and the 10-fold CV error against $\log \lambda$ are presented in Figure 3. It can be seen that BIC produces a much larger value of λ than our approach. This suggests that the BIC gives values of λ that are too conservative, in the sense that MCMC mixing should not be an issue but the larger λ value will lead to a worse BSLasso approximation. The CV approach produces values of λ that are also larger than that of Algorithm 2 and are thus also too conservative.

From Table 3 it is evident that, again, our approach to determine λ can result in a consistent acceptance rate for different n values. Significant efficiency gains can be achieved since a reasonably high acceptance rate can be attained for relatively small n values.

Table 3: Normalised ESS values and MCMC acceptance rates for standard BSL and BSLasso for the cell biology example. Also shown are the different combinations of n and λ trialled for BSLasso. The first row corresponds to standard BSL, which does not require a λ value.

n	λ	acc. rate (%)	ESS P_m	ESS P_p
5000	-	21	8	8
500	4.50	16	55	63
1000	2.20	17	36	22
1500	1.10	16	16	19
2000	0.74	16	14	16

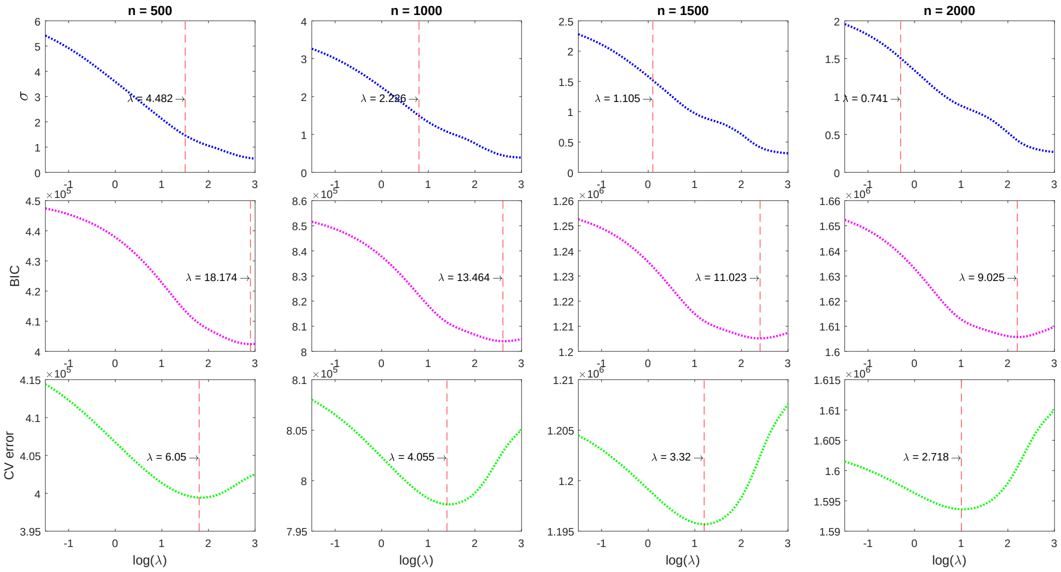

Figure 3: Comparing the methods of selecting the penalty for the cell biology example. The first row is the standard deviation of the log SL, the second row is the average BIC and the third row is the average CV error.

The posterior distributions for most combinations of n and λ in Table 3, in comparison with standard BSL, are shown in Figure 4. In this example it is apparent that very little accuracy is lost (relative to standard BSL) even for relatively large λ values. This is not surprising given the high amount of potential sparsity shown in Table 1 for this example.

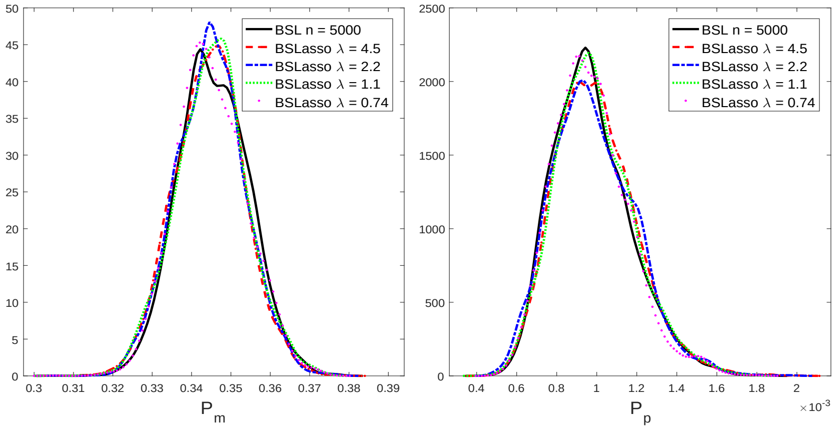

Figure 4: Posteriors for the cell biology example with standard BSL and BSLasso with various values for λ (based on the values in Table 3).

4.3 Multivariate g-and-k Example

Finally, we move on to fitting a multivariate g-and-k model to currency exchange data. The marginal distribution of a multivariate g-and-k distribution is a univariate g-and-k distribution. Following Drovandi and Pettitt (2011), we model the dependency of the variables with a Gaussian copula, see also Ong et al. (2016).

A g-and-k distribution (Rayner and MacGillivray (2002)) has 5 parameters a, b, c, g and k, where c is fixed at 0.8 in this paper, so that $\boldsymbol{\theta} = (a, b, g, k)^\top$. These four parameters grant the g-and-k distribution with considerable flexibility, controlling location, scale, skewness and kurtosis respectively. However, the distribution does not possess a closed form expression for the probability density function. Simulation can be achieved easily by plugging a uniform random number between 0 and 1 into the quantile function, $Q(p),\ p \in (0, 1)$. The quantile function of a g-and-k distribution is given by

$$Q(p|\boldsymbol{\theta}) = a + b\left(1 + c\frac{1 - \exp(-gz(p))}{1 + \exp(-gz(p))}\right)(1 + z(p)^2)^k z(p),$$

where $a \in \mathbb{R},\ b > 0,\ g \in \mathbb{R},\ k > -0.5$ and $z(p) = \Phi^{-1}(p)$ is the standard normal quantile function.

Suppose $\boldsymbol{y}$ is a $n \times q$ observation matrix from n observations, i.e. $\boldsymbol{y} = (\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n)^\top$. The dependency structure between marginals is modelled by a Gaussian copula with a $q \times q$ correlation matrix $\boldsymbol{\Delta}$. The only free parameters in $\boldsymbol{\Delta}$ are the non-diagonal elements in the upper triangular matrix of $\boldsymbol{\Delta}$. Let $\delta_{i,j}$ be the (i, j)th element of $\boldsymbol{\Delta}$. We define the following non-standard vectorisation operator (column-stacked) accordingly

$$\text{vec}(\boldsymbol{\Delta}) = (\delta_{1,2}, \delta_{1,3}, \delta_{2,3}, \delta_{1,4}, \ldots, \delta_{3,4}, \ldots, \delta_{1,q}, \ldots, \delta_{q-1,q})^\top.$$

The parameter of the multivariate g-and-k distribution is defined as $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^\top, \ldots, \boldsymbol{\theta}_q^\top, \text{vec}(\boldsymbol{\Delta})^\top)^\top$, where $\boldsymbol{\theta}_j = (a_j, b_j, g_j, k_j)^\top,\ j = 1, \ldots, q$ is the parameter for the jth marginal. The density function for one observation can be written as

$$f(\boldsymbol{y}_i|\boldsymbol{\theta}) = |\boldsymbol{\Delta}|^{-1/2} \exp(\boldsymbol{z}_i^\top(\mathbf{I}_q - \boldsymbol{\Delta}^{-1})\boldsymbol{z}_i) \prod_{j=1}^{q} f(y_{i,j}|\boldsymbol{\theta}_j),\ i = 1, \ldots, n,$$

where $\mathbf{I}_q$ is the $q \times q$ identity matrix, $\boldsymbol{z}_i = (z(F(y_{i,1}|\boldsymbol{\theta_1})), \ldots, z(F(y_{i,q}|\boldsymbol{\theta_q})))^\top$ and $F(\cdot|\boldsymbol{\theta})$ is the cumulative distribution function of the univariate g-and-k distribution given $\boldsymbol{\theta}$.

We consider a trivariate dataset, i.e. $q = 3$, of foreign currency daily exchange rates between June 1, 2007 and 31 December, 2013 (1652 trading days) from US dollar, Euro and Japanese Yen to Australian dollar (dataset obtained from `http://www.rba.gov.au/statistics/historical-data.html`). Let $\boldsymbol{x}$ be the 1652×3 observation matrix with row being the time index and column representing each currency. Assume that the log daily return $d_{t,i} = \log(x_{t+1,j}/x_{t,j}),\ t = 1, \ldots, 1651,\ j = 1, 2, 3$ follows a trivariate g-and-k distribution. There are 15 parameters to estimate, $\boldsymbol{\theta} = (a_1, b_1, g_1, k_1, a_2, b_2, g_2, k_2, a_3, b_3, g_3, k_3, \delta_{12}, \delta_{13}, \delta_{23})^\top$.

We consider a re-parametrisation so that we can sample over an unrestricted parameter space. Pinheiro and Bates (1996) introduce a spherical parametrisation that always produces a positive definite covariance matrix upon back-transformation, also see Ong et al. (2016) for another description. For simplicity, we only consider the $q = 3$ case. Let $\boldsymbol{w} = (w_1, w_2, w_3)^\top$ be the unconstrained parameter and $\boldsymbol{\Delta} = \boldsymbol{L}\boldsymbol{L}^\top$ be the proposed correlation matrix. Let $\gamma_j = \pi/ (1 + \exp(-w_j))$ for $j = 1, 2, 3$ and

$$\boldsymbol{L} = \begin{bmatrix} 1 & 0 & 0 \\ \cos(\gamma_1) & \sin(\gamma_1) & 0 \\ \cos(\gamma_2) & \sin(\gamma_2)\cos(\gamma_3) & \sin(\gamma_2)\sin(\gamma_3) \end{bmatrix}.$$

We also follow Ong et al. (2016) and re-parametrise the marginal parameters $\boldsymbol{\theta}_j \to \tilde{\boldsymbol{\theta}}_j = (\tilde{a}_j, \tilde{b}_j, \tilde{g}_j, \tilde{k}_j)^\top$. By using the following transformations

$$\tilde{a}_j = \log\left(\frac{a_j + 0.1}{0.1 - a_j}\right),\ \tilde{b}_j = \log\left(\frac{b_j}{0.05 - b_j}\right),\ \tilde{g}_j = \log\left(\frac{g_j + 1}{1 - g_j}\right) \text{ and } \tilde{k}_j = \log\left(\frac{k_j + 0.2}{0.5 - k_j}\right),$$

we are able to sample over an unconstrained parameter space. The vector of parameters after transformation is $\tilde{\boldsymbol{\theta}} = (\tilde{\boldsymbol{\theta}}_1^\top, \tilde{\boldsymbol{\theta}}_2^\top, \tilde{\boldsymbol{\theta}}_3^\top, \boldsymbol{w}^\top)^\top$. The prior distribution for $\tilde{\boldsymbol{\theta}}_j, j = 1, 2, 3$ is $\mathcal{N}(\mathbf{0}_4, 2^2\mathbf{I}_4)$ and the prior distribution for $\boldsymbol{w}$ is $\mathcal{N}(\mathbf{0}_3, 1.75^\mathbf{2}\mathbf{I}_3)$, respectively, where $\mathbf{0}_q$ is a vector of q zeros.

We use a 15-dimensional summary statistic for $q = 3$, $\boldsymbol{s} = (\boldsymbol{s}_1^\top, \boldsymbol{s}_2^\top, \boldsymbol{s}_3^\top, \boldsymbol{s}_{cor}^\top)^\top$, where $\boldsymbol{s}_j = (s_{a,j}, s_{b,j}, s_{g,j}, s_{k,j})^\top$ for $j = 1, 2, 3$ is the vector of robust summary statistics (Drovandi and Pettitt (2011)) from the jth marginal distribution. The four components in the robust summary statistic are given by

$$s_{a,j} = L_{2,j},\ s_{b,j} = L_{3,j} - L_{1,j},\ s_{g,j} = \frac{L_{3,j} + L_{1,j} - 2L_{2,j}}{s_{b,j}} \text{ and } s_{k,j} = \frac{E_{7,j} - E_{5,j} + E_{3,j} - E_{1,j}}{s_{b,j}}.$$

In the above equations, $L_{i,j}$ and $E_{i,j}$ are the ith quantile and octile from the jth marginal, respectively.

Correlation between variates is summarised by the Gaussian rank correlation (GRCor) or normal score, see Boudt et al. (2012) for details. The GRCor has a range between -1 and 1, so when there exists a strong correlation between variables, the normality assumption might be violated. Thus, we adopt the Fisher transformation (Fisher (1915)) to transform the estimated correlations to approximate normality,

$$\tilde{\rho}_g = \frac{1}{2}\log\left(\frac{\rho_g + 1}{1 - \rho_g}\right),$$

where ρ_g is the estimated GRCor.

In this example, the components of the summary statistic are not on a similar scale. The diagonal elements of the precision matrix range from 10^3 to above 10^8. For instance, the diagonal elements corresponding to $s_{a,1}$ and $s_{b,1}$ are both of the order 10^8, whilst the diagonal elements corresponding to $s_{g,1}$ and $s_{k,1}$ are of the order 10^3. Thus, we perform glasso on the standardised summary statistic. For a summary statistic of 15 dimensions, we choose $n = 15, 20, 30, 50$ for comparison of the BSLasso posteriors. We use $M = 300$ iterations to determine λ via Algorithm 1, as well as to compare with the BIC and CV methods. The results for the three approaches are shown in Figure 5. Note, with respect to the CV method, we use 7 folds for $n = 15$ and 10 folds for all other values of n. The candidates for λ are $e^{-6}, e^{-5.8}, \ldots, e^0$. It is evident that the BIC is minimised near zero and thus will produce penalty values that lead to a log SL with a very large variance and hence poor MCMC mixing. Penalties by CV are very similar to Algorithm 1. Table 4 shows our choice of the λ values, and we only show the ESS values for a subset of the parameters for brevity. The dimension of the summary statistic is relatively large in this example, however, the acceptance rate is still very high. The n used in BSL is 60, which appears to be the most efficient choice out of the tested values $n = 20, 30, 40, 50, 60, 75, 100, 125, 150$. Posteriors by BSLasso are not substantially different from those by BSL considering there is not a very high level of sparsity in this example. BSLasso generally gives more conservative posteriors comparing to BSL, in the sense that the posterior variances are inflated. The smaller the penalty value the closer the BSLasso approximation is to the BSL posteriors, as expected.

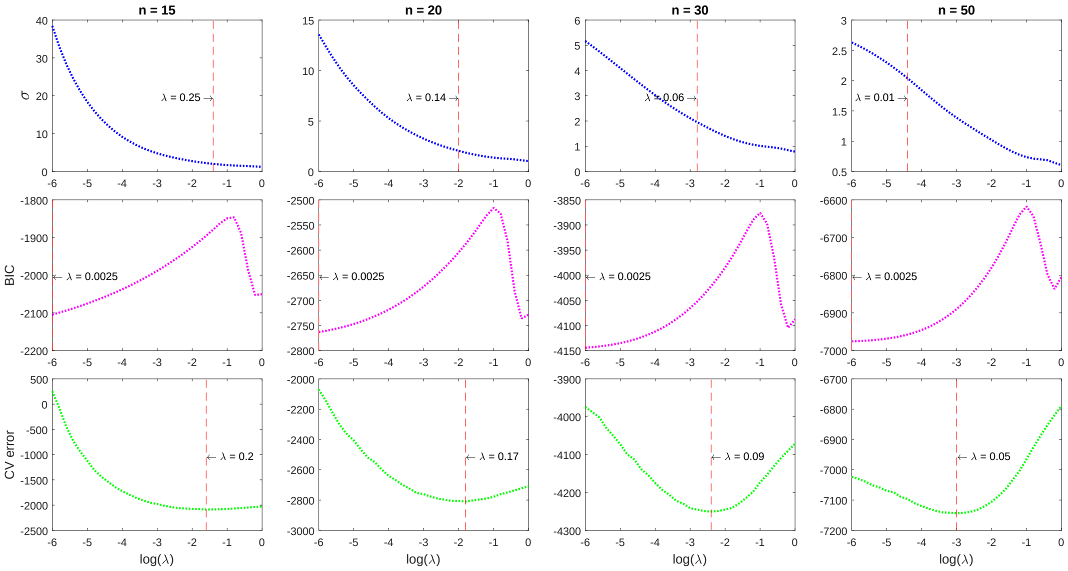

Figure 5: Comparing the methods of selecting the penalty for the multivariate g-and-k example. The first row is the standard deviation of the log SL, the second row is the average BIC and the third row is the average CV error.

Table 4: Normalised ESS values and MCMC acceptance rates for standard BSL and BSLasso for the multivariate g-and-k example using $\sigma = 1.5$. Also shown are the different combinations of n and λ trialled for BSLasso. The first row corresponds to standard BSL, which does not require a λ value.

n	λ	acc. rate (%)	ESS a_1	ESS b_1	ESS g_1	...	ESS δ_{12}	ESS δ_{13}	ESS δ_{23}
60	-	23	545	601	522	⋯	626	653	622
15	0.55	36	1615	1917	2222	⋯	2540	1861	2604
20	0.30	37	1526	1597	1774	⋯	1859	1640	1910
30	0.11	34	1225	1235	1258	⋯	1372	1221	1309
50	0.04	34	812	873	808	⋯	947	893	930

5 Discussion

In this paper we have presented an approach to accelerate the BSL method of Price et al. (2017) by using the glasso estimator for the covariance matrix of a set of summary statistics rather than using the sample covariance matrix. We have demonstrated that significantly less model simulations are required to form an approximate posterior. The accuracy of the resulting posterior relative to standard BSL will be application dependent. We expect our approach to be most effective in applications where the computation time is dominated by model simulation (in the presence of a high dimensional summary statistic

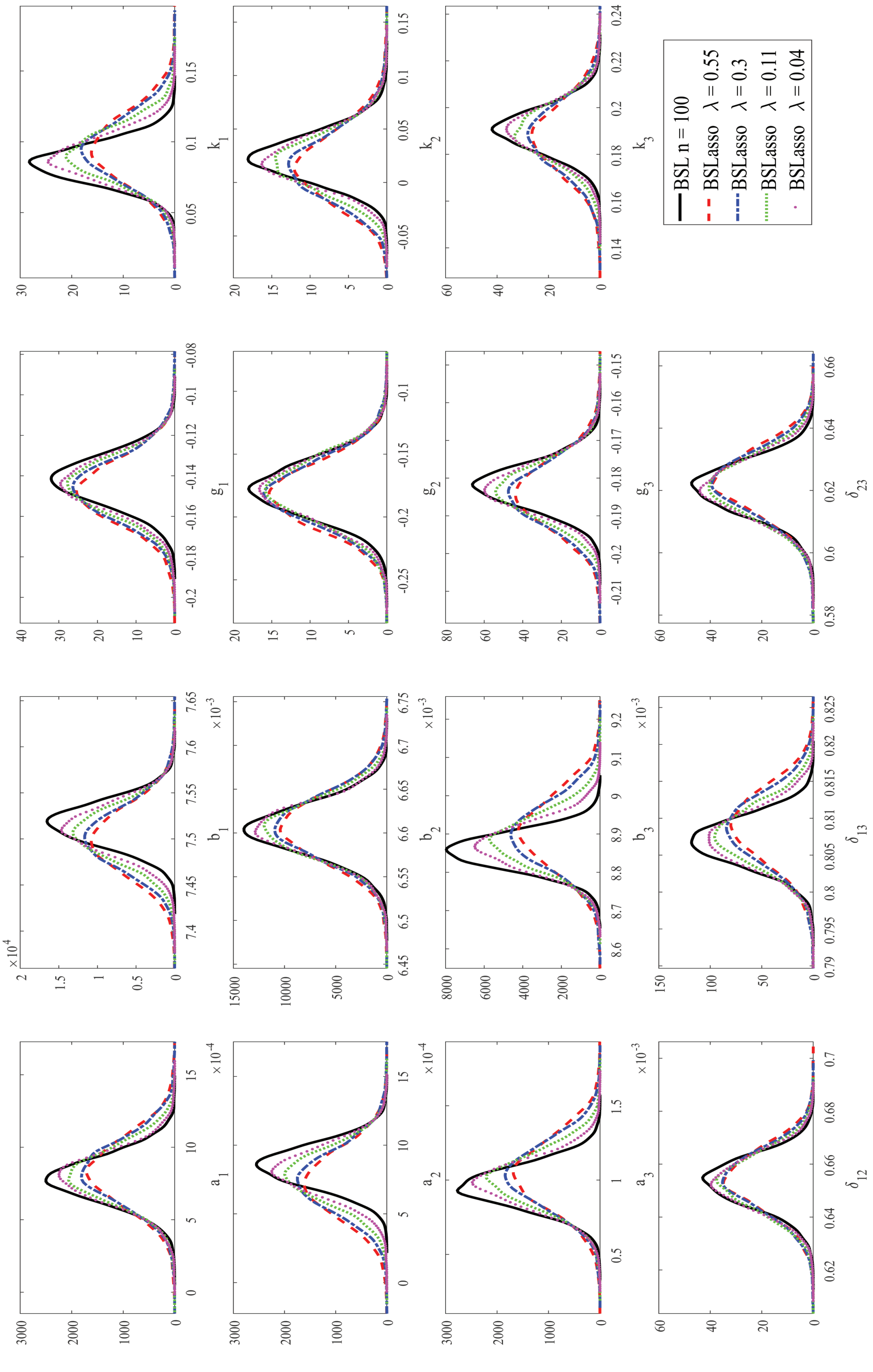

Figure 6: Posteriors for the multivariate g-and-k example using $\sigma = 1.5$ with standard BSL and BSLasso with various values for λ (based on the values in Table 4).

and/or an expensive model simulator) and where the joint distribution of the summary statistic is regular enough in non-negligible posteriors regions so that it can be reasonably approximated with a multivariate normal distribution.

It is clear that the major drawback of SL methods is the multivariate normal assumption, despite the fact that Price et al. (2017) provide empirical evidence to show that BSL is partially robust to this assumption. One approach that we may consider to help verify if the normal assumption is reasonable is the marginal adjustment method of Nott et al. (2014). The approach of Nott et al. (2014) partially involves determining accurate estimates for each of the posterior marginals by focussing on statistics relevant only for individual parameters, which is likely to be low-dimensional. The marginal posteriors from BSL could then be compared to the ABC marginal adjustments. In future research, we plan to develop methods to relax the multivariate normal assumption whilst maintaining our capability of using various shrinkage estimators of the covariance matrix.

In this paper we have determined an off-line approach for choosing an appropriate penalty value λ to use in MCMC BSLasso that is tied in with a particular choice of n. However, an adaptive MCMC strategy that could select this penalty on-the-fly would be useful. We leave that for future research.

Our main conclusion is that standard approaches for selecting λ in a typical data analysis are not appropriate for use within an MCMC algorithm as we do. We suggest that the value of λ should be chosen as small as possible such that reasonable MCMC acceptance rates can still be expected assuming that a reasonable proposal distribution is selected. We use $\sigma = 1.5$ throughout the three examples, however, larger σ is also worth considering if the distribution of the estimated log SL has noticeable skewness. In particular, we are more concerned about left skewness rather than right skewness as $\boldsymbol{\theta}$ values that produce small log SL are likely to be rejected so that the Markov Chain will not get stuck. We use boxplots (see Appendix C) to investigate the behaviour of the distribution of the estimated synthetic log-likelihood when the value of σ is between 1 and 3, which might be considered reasonable values for σ. For visualisation purposes, we only consider a small number of penalty values in the boxplots. For instance, in Figure 10 of the web appendix C, the boxplots of the estimated log SL of the MA(2) example show approximate symmetry for the chosen n and λ values. Boxplots for the cell biology example are roughly symmetric as well, we omit the figure for brevity. Boxplots for the multivariate g-and-k example in Figure 11 of the web appendix show a tendency for left skewness in the distribution of the log SLs. The standard deviation of the log SL in this example could be inflated. In this example we also expect reasonable posterior results with slightly larger σ values. Results regarding $\sigma = 2$ are shown in Appendix D. The BSLasso posteriors using $\sigma = 2$ are more accurate than those using $\sigma = 1.5$ at the expense of losing a small amount of efficiency.

Acknowledgements

CCD was supported by an Australian Research Council's Discovery Early Career Researcher Award funding scheme (DE160100741). ZA was supported by a scholarship under CCDs grant DE160100741 and a top-up scholarship from the Australian Research Council Centre of Excellence for Mathematical and Statistics Frontiers (ACEMS). DJN was supported by a Singapore Ministry of Education Academic Research Fund Tier 2 grant (R-155-000-143-112). Computational resources and services used in this work were provided by the HPC and Research Support Group, Queensland University of Technology, Brisbane, Australia.

Supplementary Material

Appendices Further discussions regarding (A) sensitivity to n and (B) multivariate g-and-k example with $\sigma = 2$ (.pdf file)

Matlab Code Matlab implementation of the BSLasso method for the MA(2) example. Please see the README.txt file for details on how to run the code (.zip file)

References

Alquier, P., Friel, N., Everitt, R., and Boland, A. (2016). Noisy Monte Carlo: Convergence of Markov chains with approximate transition kernels. *Statistics and Computing*, 26(1-2):29–47.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269–342.

Andrieu, C. and Roberts, G. O. (2009). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics*, 37(2):697–725.

Beaumont, M. A. (2010). Approximate Bayesian computation in evolution and ecology. *Annual Review of Ecology, Evolution, and Systematics*, 41:379–406.

Beaumont, M. A., Zhang, W., and Balding, D. J. (2002). Approximate Bayesian computation in population genetics. *Genetics*, 162(4):2025–2035.

Blum, M. G. B. (2010). Approximate Bayesian computation: a non-parametric perspective. *Journal of the American Statistical Association*, 105(491):1178–1187.

Blum, M. G. B., Nunes, M. A., Prangle, D., and Sisson, S. A. (2013). A comparative review of dimension reduction methods in approximate Bayesian computation. *Statistical Science*, 28(2):189–208.

Boudt, K., Cornelissen, J., and Croux, C. (2012). The Gaussian rank correlation estimator: robustness properties. *Statistics and Computing*, 22(2):471–483.

Dale, P. D., Maini, P. K., and Sherratt, J. A. (1994). Mathematical modeling of corneal epithelial wound healing. *Mathematical Biosciences*, 124:127–147.

Doucet, A., Pitt, M. K., Deligiannidis, G., and Kohn, R. (2015). Efficient implementation of Markov chain Monte Carlo when using an unbiased likelihood estimator. *Biometrika*, 102(2):295–313.

Drovandi, C. C. and Pettitt, A. N. (2011). Likelihood-free Bayesian estimation of multivariate quantile distributions. *Computational Statistics and Data Analysis*, 55(9):2541–2556.

Fan, J., Liao, Y., and Liu, H. (2016). An overview of the estimation of large covariance and precision matrices. *The Econometrics Journal*, 19(1):C1–C32.

Fisher, R. A. (1915). Frequency distribution of the values of the correlation coefficient in samples from an indefinitely large population. *Biometrika*, 10(4):507–521.

Friedman, J., Hastie, T., and Tibshirani, R. (2008). Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441.

Gao, X., Pu, D. Q., Wu, Y., and Xu, H. (2012). Tuning parameter selection for penalized likelihood estimation of Gaussian graphical model. *Statistica Sinica*, 22:1123–1146.

Johnston, S., Simpson, M. J., McElwain, D. L. S., Binder, B. J., and Ross, J. V. (2014). Interpreting scratch assays using pair density dynamic and approximate Bayesian computation. *Open Biology*, 4(9):140097.

Ledoit, O. and Wolf, M. (2004). A well-conditioned estimator for large-dimensional covariance matrices. *Journal of Multivariate Analysis*, 88(2):365–411.

Lee, A. and Łatuszyński, K. (2014). Variance bounding and geometric ergodicity of Markov chain Monte Carlo kernels for approximate Bayesian computation. *Biometrika*, 101(3):655–671.

Marin, J.-M., Pudlo, P., Robert, C. P., and Ryder, R. J. (2012). Approximate Bayesian computation methods. *Statistics and Computing*, 22(6):1167–1180.

Marjoram, P., Molitor, J., Plagnol, V., and Tavaré, S. (2003). Markov chain Monte Carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26):15324–15328.

Nott, D. J., Fan, Y., Marshall, L., and Sisson, S. (2014). Approximate Bayesian computation and Bayes linear analysis: toward high-dimensional ABC. *Journal of Computational and Graphical Statistics*, 23(1):65–86.

Ong, V. M., Nott, D. J., Tran, M.-N., Sisson, S. A., and Drovandi, C. C. (2016). Variational Bayes with synthetic likelihood. *arXiv preprint arXiv:1608.03069*.

Pinheiro, J. C. and Bates, D. M. (1996). Unconstrained parametrizations for variance-covariance matrices. *Statistics and Computing*, 6(3):289–296.

Price, L. F., Drovandi, C. C., Lee, A., and Nott, D. J. (2017). Bayesian synthetic likelihood. *Journal of Computational and Graphical Statistics*.

Rayner, G. and MacGillivray, H. (2002). Weighted quantile-based estimation for a class of transformation distributions. *Computational statistics & data analysis*, 39(4):401–433.

Smith, Jr., A. A. (1993). Estimating nonlinear time-series models using simulated vector autoregressions. *Journal of Applied Econometrics*, 8(S1):S63–S84.

Swanson, K. R., Bridge, C., Murray, J. D., and Jr, E. C. A. (2003). Virtual and real brain tumor: using mathematical modeling to quantify glioma growth and invasion. *Journal of the Neurological Sciences*, 216:1–10.

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

Wood, S. N. (2010). Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466:1102–1107.

Web Appendices for Accelerating Bayesian Synthetic Likelihood with the Graphical Lasso by An et al 2017

A MCMC BSL Algorithm

```
Input : Summary statistic of the data, $\boldsymbol{s_y}$, the prior distribution, $p(\boldsymbol{\theta})$, the
        proposal distribution $q$, the number of iterations, $T$, and the initial value
        of the chain $\boldsymbol{\theta}^0$.
Output: MCMC sample $(\boldsymbol{\theta}^0, \boldsymbol{\theta}^1, \ldots, \boldsymbol{\theta}^T)$ from the BSL posterior, $p_{A,n}(\boldsymbol{\theta}|\boldsymbol{s_y})$. Some
        samples can be discarded as burn-in if required.
Simulate $\boldsymbol{x}_{1:n} \overset{\text{iid}}{\sim} p(\cdot|\boldsymbol{\theta}^0)$ and compute $\boldsymbol{s}_{1:n}$
Compute $\boldsymbol{\phi}^0 = (\boldsymbol{\mu}_n(\boldsymbol{\theta}^0), \boldsymbol{\Sigma}_n(\boldsymbol{\theta}^0))$ using (2)
for $i = 1$ to $T$ do
    Draw $\boldsymbol{\theta}^* \sim q(\cdot|\boldsymbol{\theta}^{i-1})$
    Simulate $\boldsymbol{x}^*_{1:n} \overset{\text{iid}}{\sim} p(\cdot|\boldsymbol{\theta}^*)$ and compute $\boldsymbol{s}^*_{1:n}$
    Compute $\boldsymbol{\phi}^* = (\boldsymbol{\mu}_n(\boldsymbol{\theta}^*), \boldsymbol{\Sigma}_n(\boldsymbol{\theta}^*))$ using (2)
    Compute $r = \min\left(1, \frac{\mathcal{N}(\boldsymbol{s_y};\boldsymbol{\mu}_n(\boldsymbol{\theta}^*),\boldsymbol{\Sigma}_n(\boldsymbol{\theta}^*))p(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}^{i-1}|\boldsymbol{\theta}^*)}{\mathcal{N}(\boldsymbol{s_y};\boldsymbol{\mu}_n(\boldsymbol{\theta}^{i-1}),\boldsymbol{\Sigma}_n(\boldsymbol{\theta}^{i-1}))p(\boldsymbol{\theta}^{i-1})q(\boldsymbol{\theta}^*|\boldsymbol{\theta}^{i-1})}\right)$
    if $\mathcal{U}(0,1) < r$ then
        Set $\boldsymbol{\theta}^i = \boldsymbol{\theta}^*$ and $\boldsymbol{\phi}^i = \boldsymbol{\phi}^*$
    else
        Set $\boldsymbol{\theta}^i = \boldsymbol{\theta}^{i-1}$ and $\boldsymbol{\phi}^i = \boldsymbol{\phi}^{i-1}$
    end
end
```

Algorithm 2: MCMC BSL algorithm.

B Sensitivity to n

In the main paper, we show that the accuracy of the BSLasso posterior distributions is affected by the choice of penalty value. However, it is unclear if the BSLasso target posterior distribution is also sensitive to n. It is of interest to see if the property of insensitivity to n of BSL (Price et al. (2017)) carries over to BSLasso. For this purpose, we trialled different values of n with penalty value fixed at our choice in each example. Figures 7, 8 and 9 show the comparison of posteriors at each λ chosen for the MA(2), cell biology and multivariate g-and-k examples, respectively. The number of MCMC iterations in every trial is taken to be large enough so that the Monte Carlo error does not dominate.

Figures below imply that BSLasso is also insensitive to n. Thus we can conclude that the estimated BSLasso posterior distributions shown in Figures 2, 4 and 6 of the main paper are mainly influenced by λ not n.

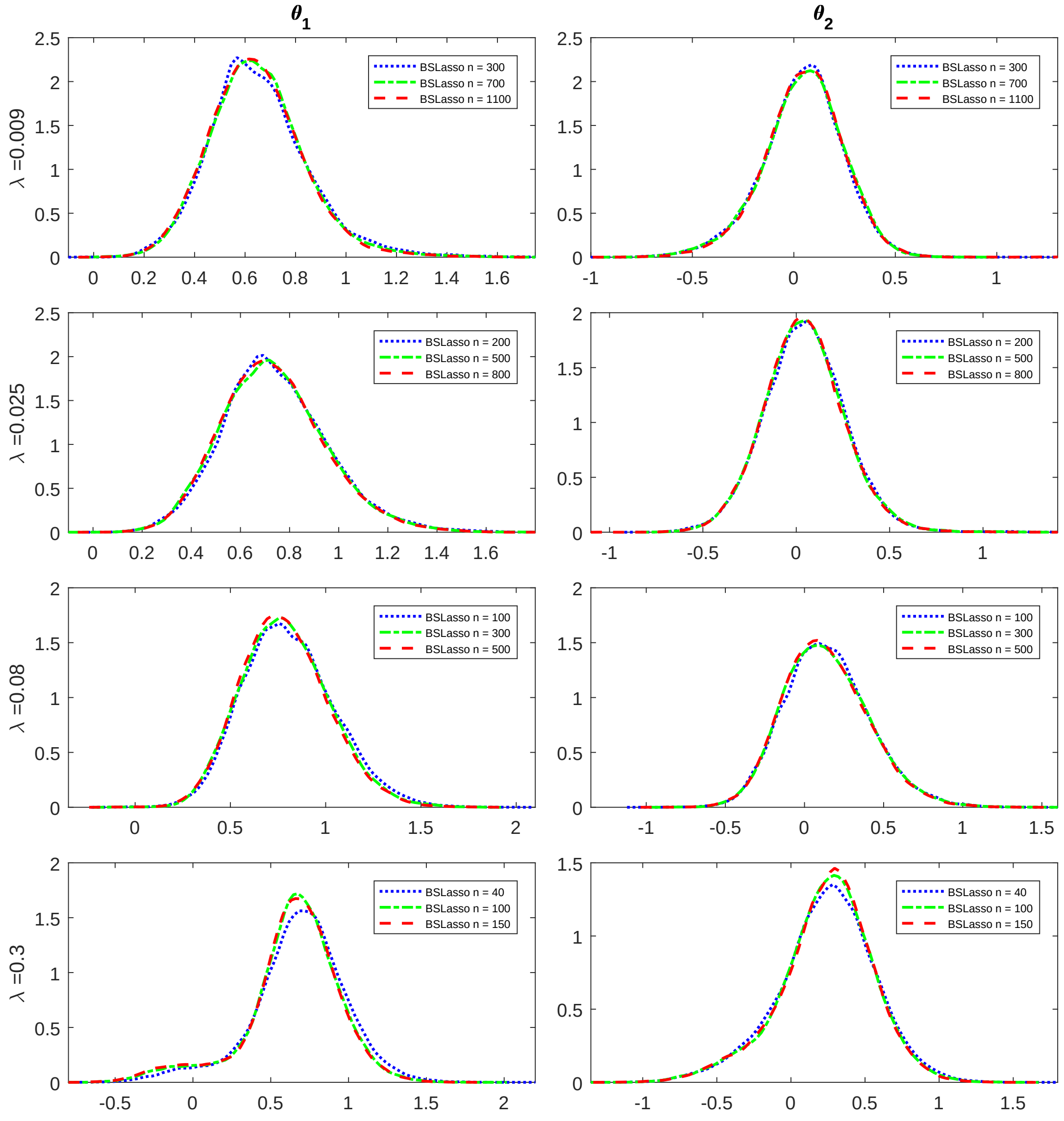

Figure 7: Posteriors for the MA(2) example with standard BSL and BSLasso with different values of n and λ. Results show that the BSLasso posterior is not sensitive to n.

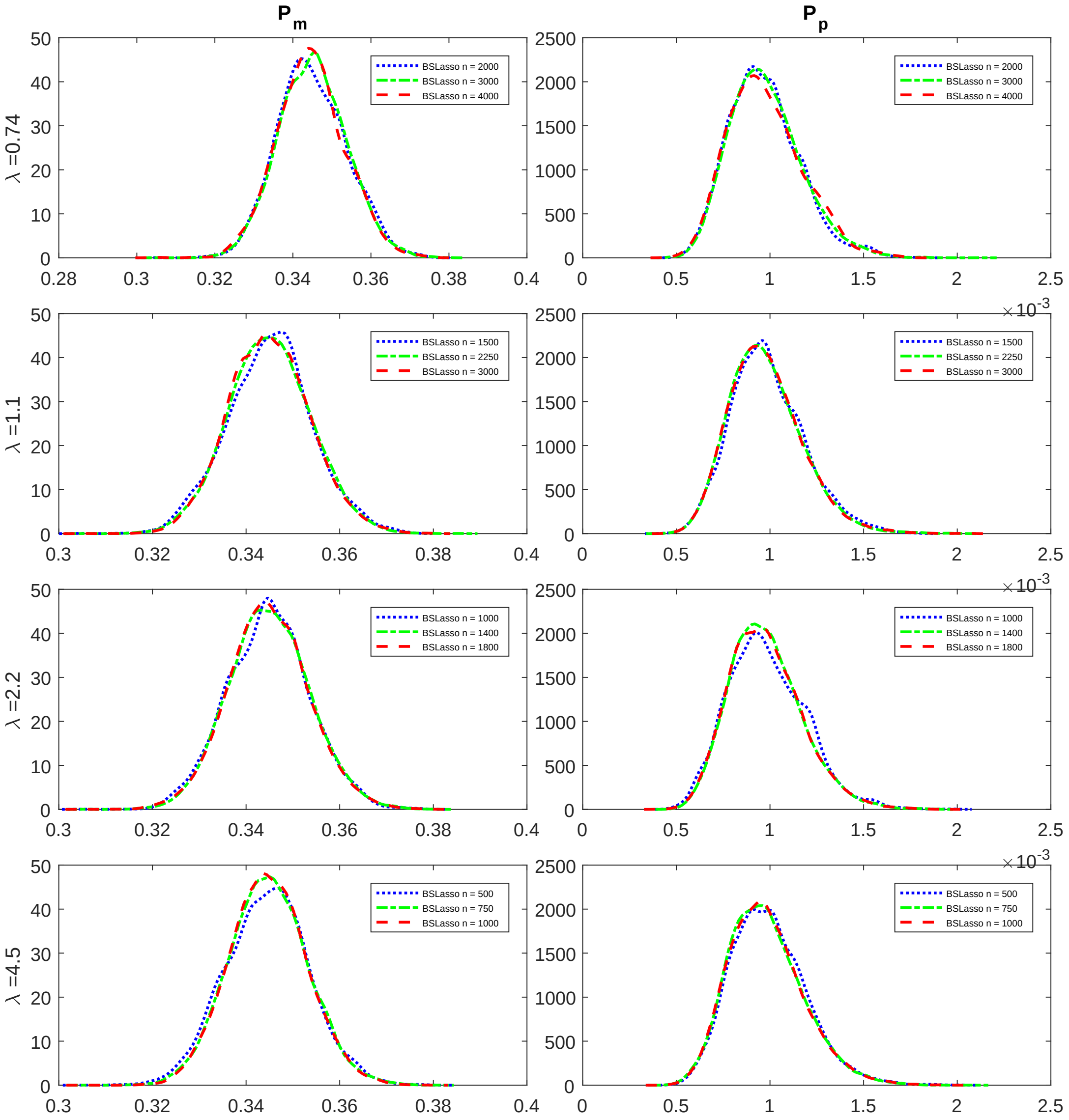

Figure 8: Posteriors for the cell biology example with standard BSL and BSLasso with different values of n and λ. Results show that BSLasso posterior is not sensitive to n.

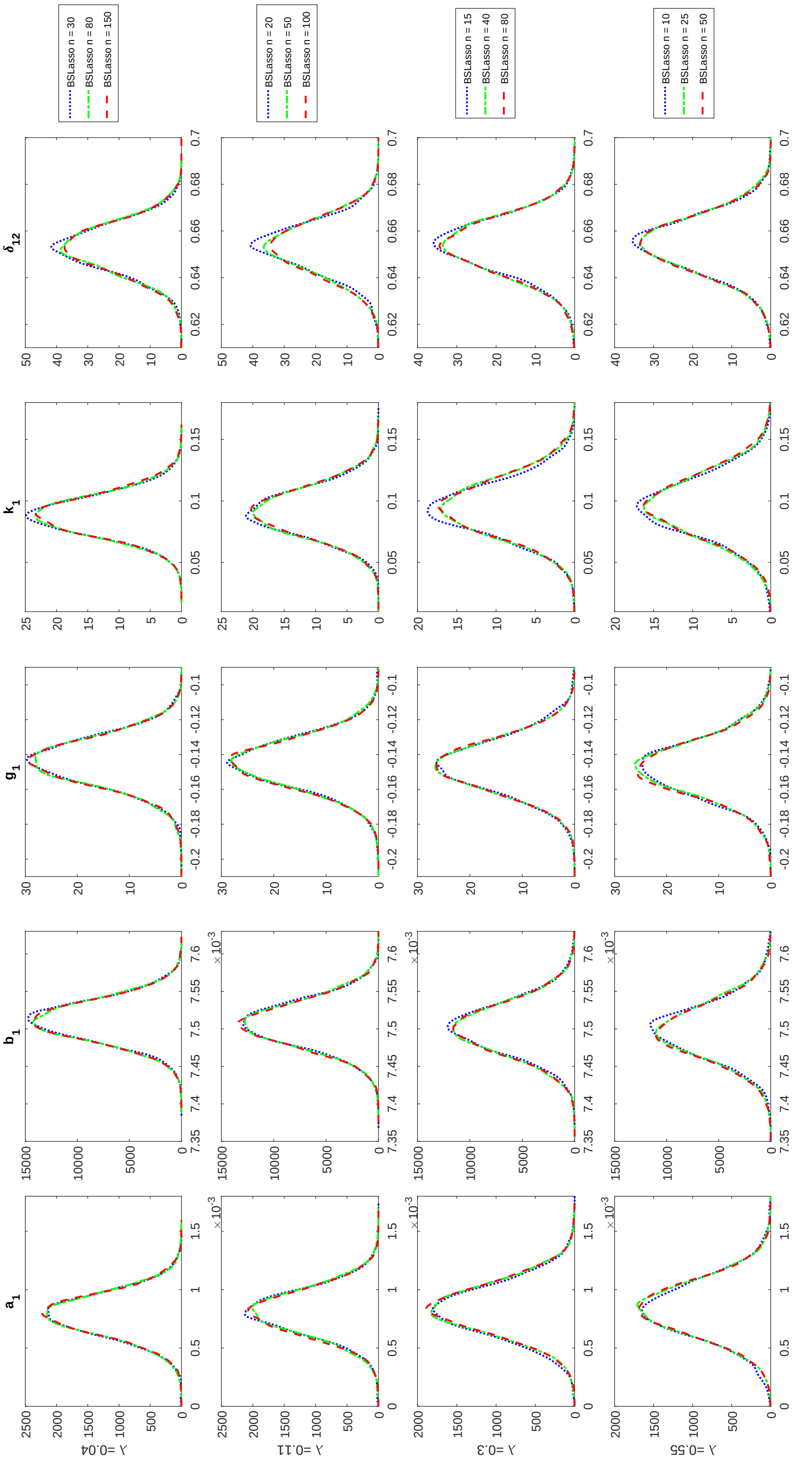

Figure 9: Posteriors for the multivariate g-and-k example with standard BSL and BSLasso with different values of n and λ. Results show that BSLasso posterior is not sensitive to n.

C Boxplots of Log Synthetic Likelihoods

Boxplots of the log SL estimates for the MA(2) and multivariate g-and-k examples are shown in Figures 10 and 11 respectively.

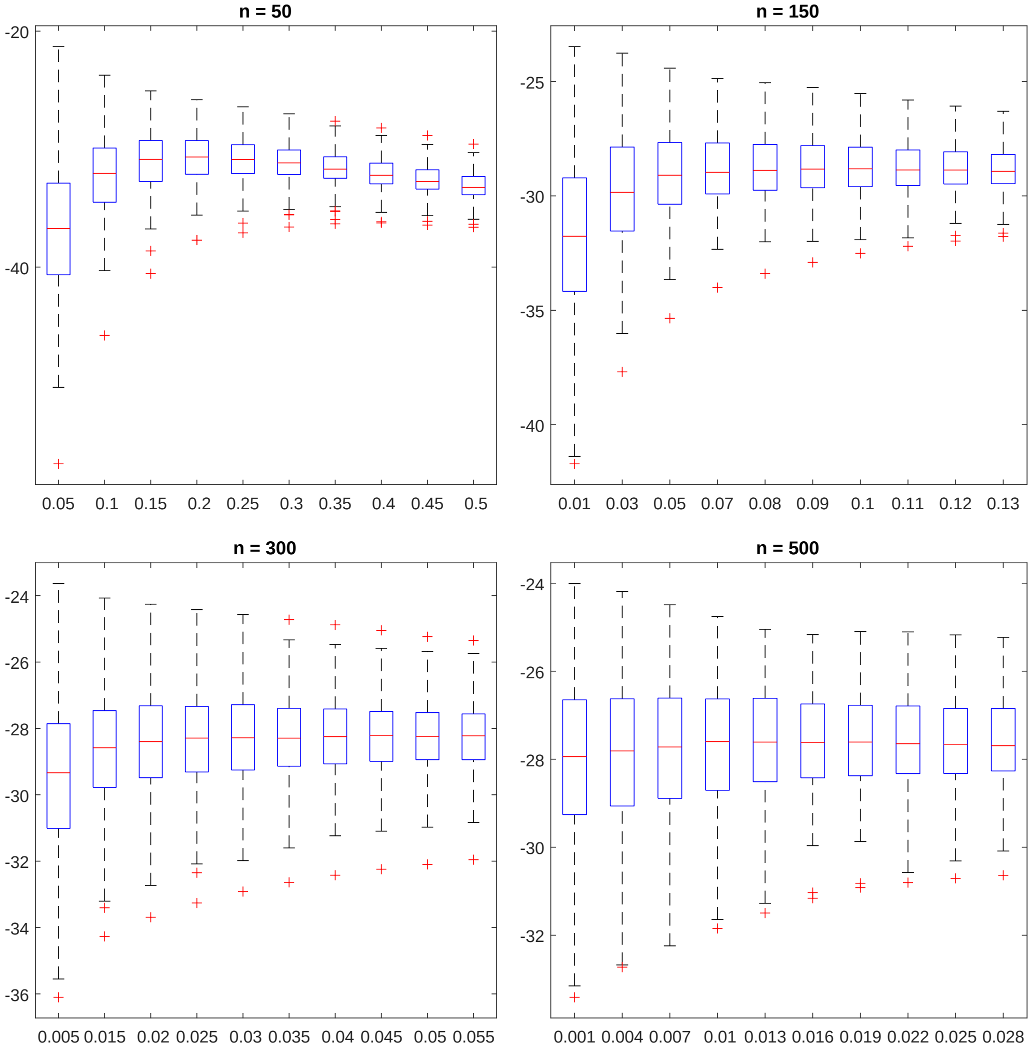

Figure 10: Boxplots of the estimated log SL estimates obtained for different combinations of n and λ for the the MA(2) example based on $\boldsymbol{\theta} = (0.6, 0.2)^\top$. The x-axis of each plots shows the λ value and the corresponding y-axis is the estimated log SL.

D Multivariate g-and-k Example with $\sigma = 2$

In this section, we re-run the multivariate g-and-k example using $\sigma = 2$. The data and methods are the same as Section 4.3 of the main paper. Figure 12 shows the posterior distributions with λ chosen on the basis of $\sigma = 2$. The selected penalty values are shown in Table 5 together with the ESS values for a subset of the parameters. Given that a σ value of 2 leads to smaller values of λ compared with $\sigma = 1.5$, the BSLasso posteriors are

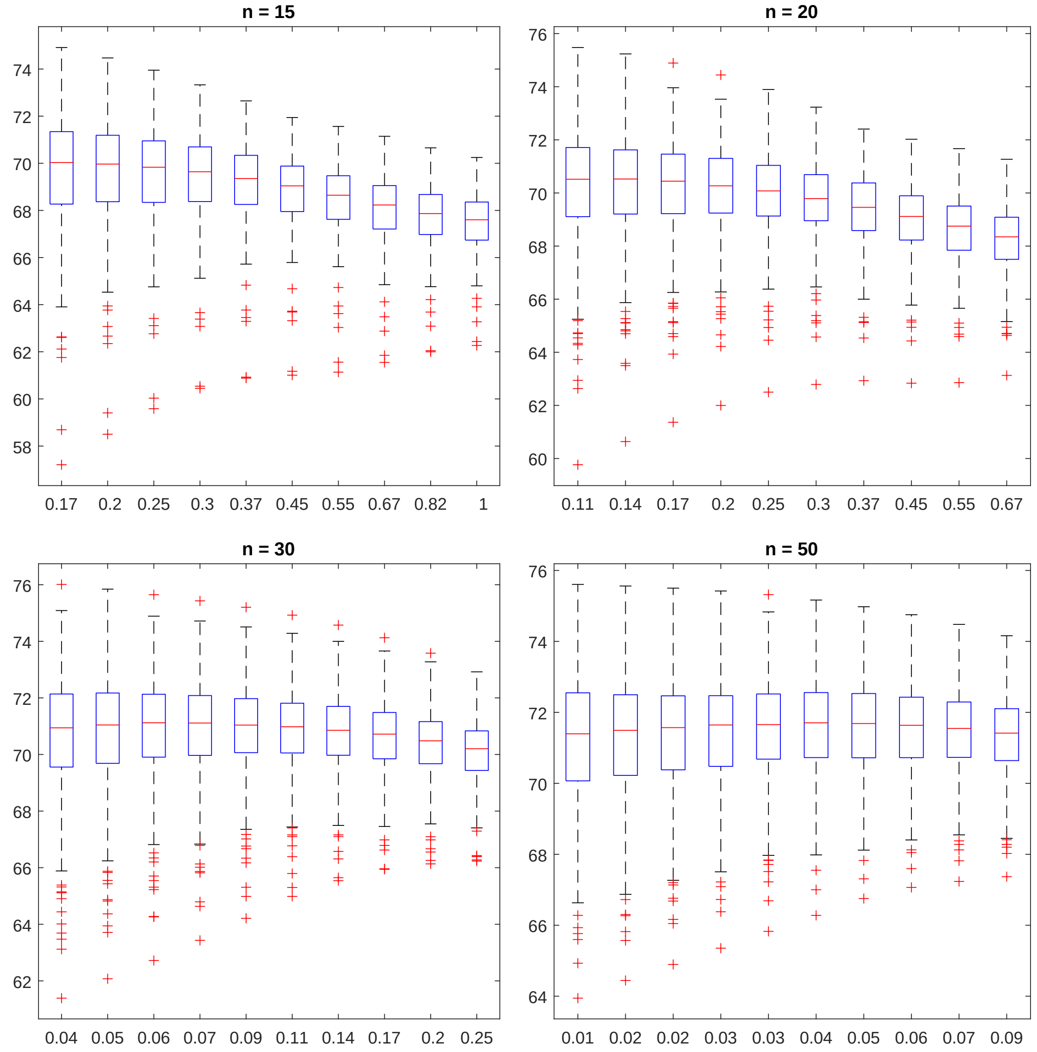

Figure 11: Boxplots of the estimated log SL estimates obtained for different combinations of n and λ for the multivariate g-and-k example based on the currency exchange data. The x-axis of each plots shows the λ value and the corresponding y-axis is the estimated log SL.

more accurate in the sense that plots in Figure 12 are closer to the BSL posteriors than Figure 6. As expected, there is a decrease in the acceptance rate and ESS values for the larger σ, however the efficiency gains of the BSLasso approach remain clear.

Table 5: Normalised ESS values and MCMC acceptance rates for standard BSL and BSLasso for the multivariate g-and-k example using $\sigma = 2$. Also shown are the different combinations of n and λ trialled for BSLasso. The first row corresponds to standard BSL, which does not require a λ value.

n	λ	acc. rate (%)	ESS a_1	ESS b_1	ESS g_1	$\cdots$	ESS δ_{12}	ESS δ_{13}	ESS δ_{23}
60	-	23	545	601	522	$\cdots$	626	653	622
15	0.25	29	1692	1813	1748	$\cdots$	2000	1781	2031
20	0.14	28	1406	1467	1442	$\cdots$	1708	1604	1704
30	0.06	27	1034	1093	1061	$\cdots$	1145	1064	1143
50	0.01	25	644	685	615	$\cdots$	797	768	788

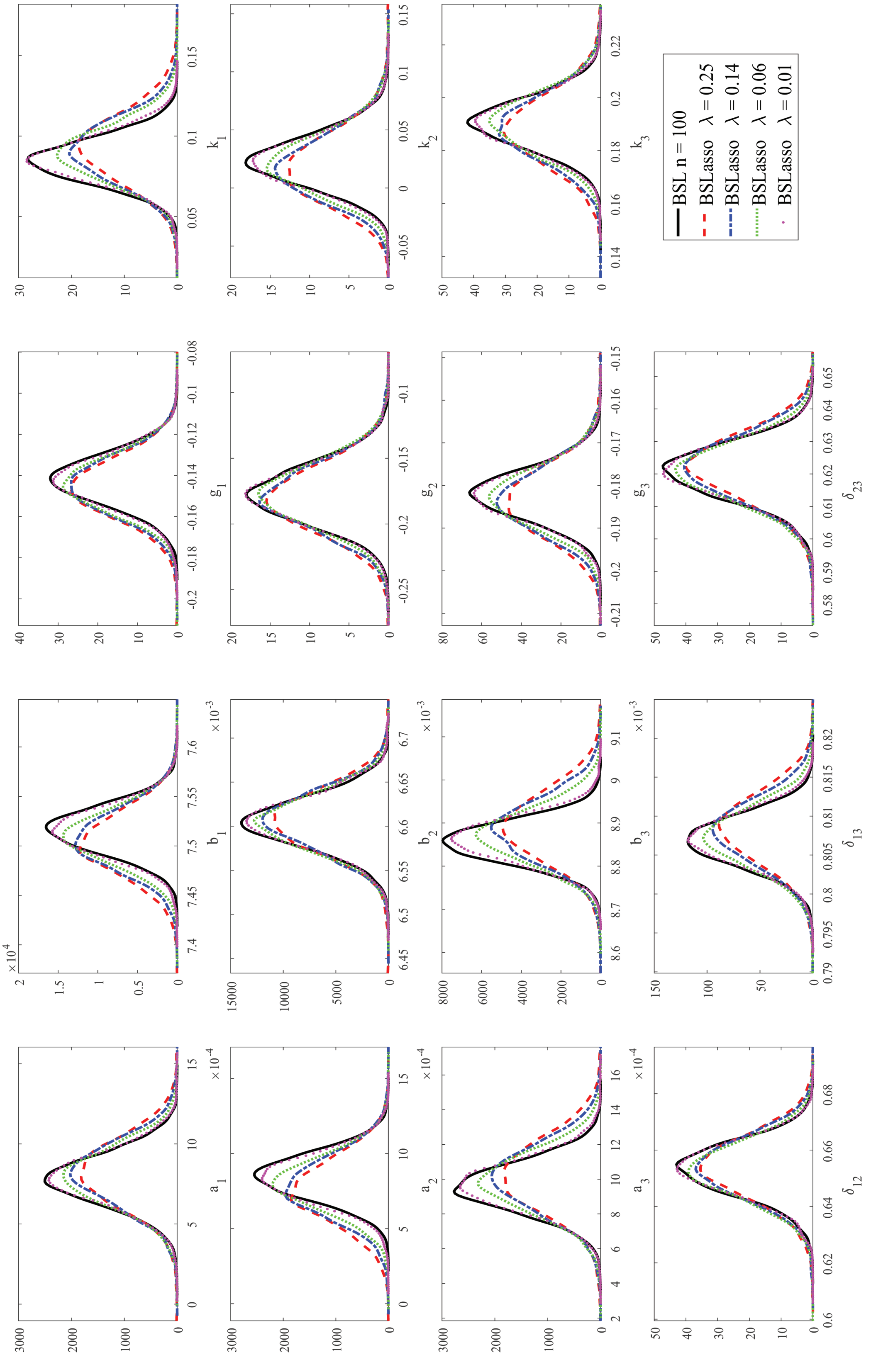

Figure 12: Posteriors for the multivariate g-and-k example using $\sigma = 2$ with standard BSL and BSLasso with various values for λ (based on the values in Table 5).