Technometrics

ISSN: (Print) (Online) Journal homepage: https://www.tandfonline.com/loi/utch20

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

Rene Gutierrez & Rajarshi Guhaniyogi

To cite this article: Rene Gutierrez & Rajarshi Guhaniyogi (2022) Bayesian Dynamic Feature Partitioning in High-Dimensional Regression With Big Data, Technometrics, 64:2, 224-240, DOI: 10.1080/00401706.2021.1952899

To link to this article: https://doi.org/10.1080/00401706.2021.1952899

TECHNOMETRICS
2022, VOL. 64, NO. 2, 224–240
https://doi.org/10.1080/00401706.2021.1952899

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

Rene Gutierrez*[a] and Rajarshi Guhaniyogi*[b]

[a]Department of Statistics, UC Santa Cruz, CA; [b]Department of Statistics, SOE2, UC Santa Cruz, Santa Cruz, CA

ABSTRACT

Bayesian computation of high-dimensional linear regression models using Markov chain Monte Carlo (MCMC) or its variants can be extremely slow or completely prohibitive since these methods perform costly computations at each iteration of the sampling chain. Furthermore, this computational cost cannot usually be efficiently divided across a parallel architecture. These problems are aggravated if the data size is large or data arrive sequentially over time (streaming or online settings). This article proposes a novel dynamic feature partitioned regression (DFP) for efficient online inference for high-dimensional linear regressions with large or streaming data. DFP constructs a *pseudo posterior density* of the parameters at every time point, and quickly updates the pseudo posterior when a new block of data (data shard) arrives. DFP updates the pseudo posterior at every time point suitably and partitions the set of parameters to exploit parallelization for efficient posterior computation. The proposed approach is applied to high-dimensional linear regression models with Gaussian scale mixture priors and spike-and-slab priors on large parameter spaces, along with large data, and is found to yield state-of-the-art inferential performance. The algorithm enjoys theoretical support with pseudoposterior densities over time being arbitrarily close to the full posterior as the data size grows, as shown in the supplementary material. Supplementary material also contains details of the DFP algorithm applied to different priors. Package to implement DFP is available in *https://github.com/Rene-Gutierrez/DynParRegReg*. The dataset is available in *https://github.com/Rene-Gutierrez/DynParRegReg_Implementation*.

ARTICLE HISTORY
Received February 2019
Accepted May 2021

KEYWORDS
Bayesian statistics; Data shards; High-dimensional regression; Shrinkage prior; Streaming data; Sufficient statistics

1. Introduction

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

$$y = \boldsymbol{x}'\boldsymbol{\beta} + \epsilon, \quad \epsilon \sim N(0, \sigma^2), \tag{1}$$

where $\boldsymbol{x}$ is a $p \times 1$ predictor, $\boldsymbol{\beta}$ is the corresponding $p \times 1$ coefficient, y is the continuous response and σ^2 is the error variance. Bayesian methods for estimating $\boldsymbol{\beta}$ provide a natural probabilistic characterization of uncertainty in the parameters and in predictions. Fitting Bayesian linear regression models in the presence of very high-dimensional predictors presents onerous computational burdens either due to decomposition of large matrices or due to poor convergence and inferential issues caused by the high correlations among the parameters. This article develops a dynamic approach, called dynamic feature partitioning (DFP), for boosting the scalability of high-dimensional Bayesian linear models for large/streaming data.

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

CONTACT Rajarshi Guhaniyogi *rguhaniy@ucsc.edu* Rajarshi Guhaniyogi Department of Statistics, SOE2, UC Santa Cruz, 1156 High Street, Santa Cruz, CA 95064.

*These authors contributed equally

Supplementary materials for this article are available online. Please go to *www.tandfonline.com/r/TECH*.

enable efficient computations in high-dimensional regressions involving small n and large p, though it is less clear as to how these approaches can be adapted when the number of samples is massive to start with, or data is observed in a stream. Besides, having small sample size may limit the inferential accuracy for large p.

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

We propose a novel online Bayesian sampling algorithm, referred to as dynamic feature partitioning (DFP) that enables efficient computation of high-dimensional regression in the presence of a large number of parameters and a large sample size. DFP works with data shards that are sequentially fed to the model. The DFP framework *dynamically* partitions the set of parameters into disjoint subsets with the onset of a new data shard and obtains posterior samples for each subset of the parameters by sampling from a distribution that conditions on functions of the point estimates of the remaining parameters and sufficient statistics from the data observed so far, instead of sampling from the full conditional distribution. While the ordinary un-approximated full conditional posterior distributions of these parameter subsets would have been updated sequentially at each iteration of the Markov Chain, DFP constructs approximations of the conditional posterior distributions of each parameter subset, allowing posterior updates of these parameter subsets at different processors in parallel. This leads to a significant gain in computational efficiency over the sequential updating of parameter subsets in the ordinary MCMC. Additionally, the algorithm needs storing and propagating only a few lower dimensional sufficient statistics of the data over time, implying storage efficiency in the model fitting procedure. Moreover, we show that the DFP algorithm leads to approximations of the conditional distributions producing samples from the correct target posterior asymptotically. The DFP algorithm is demonstrated to be highly versatile and efficient across a variety of high-dimensional linear regression settings, enabling online sampling of parameters with dramatic reductions in the per-iteration computational requirement.

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

In the general Bayesian literature of streaming data, sequential Monte Carlo (SMC) (Lopes and Tsay 2011; Doucet, De Freitas, and Gordon 2001; Moral, Jasra, and Zhou 2017) is one of the most popular online methods that relies on resampling particles sequentially as data shards arrive over time. A naive implementation of SMC might be less efficient and less accurate involving large n and p due to the need to employ very large numbers of particles to obtain adequate approximations and prevent particle degeneracy. The latter is addressed through rejuvenation steps using all the data (or sufficient statistics), which may become expensive in an online setting (Snyder et al. 2008). There are approaches in the recent years to overcome the dimensionality issues in the SMC algorithm mainly in the context of fitting state-space models. To this end, carefully constructed SMC algorithms (Chopin et al. 2004; Beskos et al. 2014; Carvalho et al. 2010) show promise in terms of scaling in a polynomial complexity with the number of parameters, though the complexity as a function of the size of the dataset is either growing with time (e.g., for Chopin et al. 2004) or is not apparent from the context. Rebeschini and Handel (2015) developed a blocking strategy for high-dimensional particle learning (PL) where the error of approximation is free of the dimension of the parameter space. Unfortunately, the numerical examples for high dimensions provided by Rebeschini and Handel (2015) did not demonstrate satisfactory performance with large state-space models. Furthermore, the results rely on the decay of correlations for state-space varying parameters in the fitted model, which is suitable in the context of state-space models, but less satisfactory for our problem of interest. Wigren, Murray, and Lindsten (2018) proposed another approach for high-dimensional PL in state-space models, though the numerical illustration of the approach may struggle to comfortably scale beyond a few dozen dimensional state-space models. Lindsten et al. (2017) proposed a new SMC algorithm based on parameter partitioning in the high-dimensional space, though difficulties may arise when joining the partitions, which requires a careful resampling. In the same vein, Gunawan et al. (2018) proposed an approach that employs a sub-sampling technique to combat the problem of large data in the realm of high-dimensional problems. Arguably, there is a general lack of extensive empirical investigations of SMC or PL algorithms proposed for high-dimensional problems, and most of them do not come with any open-source code for implementation.

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

which can be easily accommodated by the DFP algorithm (see Section 4.3).

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

The rest of the article is organized as follows. Section 2 introduces a number of shrinkage priors and variable selection priors in high-dimensional regression and describes the computational challenges with big n and p. Section 3 introduces the assumptions, notations and then the description of the DFP algorithm. Section 4 demonstrates the performance of DFP for high-dimensional linear regression with (i) the Bayesian Lasso and (ii) the Horseshoe shrinkage prior distributions and (iii) the Spike and Lasso discrete mixture prior distribution for variable selection (described in Section 2.3). Further evidence on the empirical performance of DFP is provided in the analysis of a financial dataset consisting of the minute by minute average log-prices of the NASDAQ stock exchange from September 10, 2018 to November 13, 2018 during trading hours in Section 5. Finally, Section 6 concludes the article with discussions and possibilities of future directions. Theoretical insights into the convergence behavior of the DFP algorithm are provided in the supplementary material.

2. Computational Challenges in High-Dimensional Regression Models

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

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

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

2.1. Bayesian Lasso Shrinkage Prior

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

$$\boldsymbol{y}_s|\boldsymbol{X}_s, \boldsymbol{\beta}, \sigma^2 \sim N_n\left(\boldsymbol{X}_s\boldsymbol{\beta}, \sigma^2\boldsymbol{I}_n\right), \quad s = 1, ..., t$$

$$\boldsymbol{\beta}|\boldsymbol{\tau}^2, \sigma^2 \sim N_p\left(\boldsymbol{0}, \sigma^2\boldsymbol{M}_\tau\right), \quad \tau_j^2 \sim Exp\left(\frac{\lambda^2}{2}\right),$$

$$\pi\left(\sigma^2\right) \propto \frac{1}{\sigma^2}, \quad \lambda^2 \sim G(r, d), \quad j = 1, ..., p,$$

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

by

$$\boldsymbol{\beta}|\sigma^2,\boldsymbol{\tau}^2,\lambda^2,\boldsymbol{D}^{(t)} \sim N_p\left(\left(\boldsymbol{S}_1^{(t)}+\boldsymbol{M}_\tau^{-1}\right)^{-1}\boldsymbol{S}_2^{(t)},\sigma^2\left(\boldsymbol{S}_1^{(t)}+\boldsymbol{M}_\tau^{-1}\right)^{-1}\right),$$

$$\sigma^2|\boldsymbol{\beta},\boldsymbol{\tau}^2,\lambda^2,\boldsymbol{D}^{(t)} \sim \text{IG}\left(\frac{nt+p}{2},\frac{\left(\boldsymbol{S}_3^{(t)}+\boldsymbol{\beta}'\boldsymbol{S}_1^{(t)}\boldsymbol{\beta}-2\boldsymbol{\beta}'\boldsymbol{S}_2^{(t)}\right)+\boldsymbol{\beta}'\boldsymbol{M}_\tau^{-1}\boldsymbol{\beta}}{2}\right),$$

$$\frac{1}{\tau_j^2}|\boldsymbol{\beta},\sigma^2,\lambda^2,\boldsymbol{D}^{(t)} \sim \text{Inv}-\text{Gaussian}\left(\sqrt{\frac{\lambda^2\sigma^2}{\beta_j^2}},\lambda^2\right),$$

$$\lambda^2|\boldsymbol{\beta},\sigma^2,\boldsymbol{\tau}^2,\boldsymbol{D}^{(t)} \sim \text{IG}\left(p+r,\frac{\sum_{j=1}^p\tau_j^2}{2}+d\right). \quad (2)$$

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

2.2. Horseshoe Shrinkage Prior

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

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

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

where $\boldsymbol{\beta} = \left(\beta_1,\ldots,\beta_p\right)'$, $\boldsymbol{M}_\lambda = \text{diag}\left(\lambda_1^2,\ldots,\lambda_p^2\right)$, $\boldsymbol{\lambda} = \left(\lambda_1^2,\ldots,\lambda_p^2\right)'$ and $\boldsymbol{\nu} = \left(\nu_1,\ldots,\nu_p\right)'$. The data augmentation allows the batch MCMC procedure to draw MCMC samples at time t from the following full conditional distributions,

$$\boldsymbol{\beta}|\sigma^2,\boldsymbol{\tau}^2,\lambda^2,\boldsymbol{D}^{(t)} \sim N_p\left(\left(\boldsymbol{S}_1^{(t)}+\frac{\boldsymbol{M}_\lambda^{-1}}{\tau^2}\right)^{-1}\boldsymbol{S}_2^{(t)},\sigma^2\left(\boldsymbol{S}_1^{(t)}+\frac{\boldsymbol{M}_\lambda^{-1}}{\tau^2}\right)^{-1}\right),$$

$$\sigma^2|\boldsymbol{\beta},\boldsymbol{\tau}^2,\lambda^2,\boldsymbol{D}^{(t)} \sim \text{IG}\left(\frac{nt+p}{2},\frac{\boldsymbol{S}_3^{(t)}+\boldsymbol{\beta}'\boldsymbol{S}_1^{(t)}\boldsymbol{\beta}-2\boldsymbol{\beta}'\boldsymbol{S}_2^{(t)}}{2}+\frac{\boldsymbol{\beta}'\boldsymbol{M}_\lambda^{-1}\boldsymbol{\beta}}{2\tau^2}\right),$$

$$\lambda_j^2|\beta_j,\nu_j,\tau^2,\sigma^2,\boldsymbol{D}^{(t)} \sim \text{IG}\left(1,\left[\frac{1}{\nu_j}+\frac{\beta_j^2}{2\tau^2\sigma^2}\right]\right),$$

$$\nu_j|\lambda_j^2,\boldsymbol{D}^{(t)} \sim \text{IG}\left(1,\left(1+\frac{1}{\lambda_j^2}\right)\right),$$

$$\xi|\boldsymbol{\beta},\sigma^2,\boldsymbol{\tau}^2,\boldsymbol{D}^{(t)} \sim \text{IG}\left(1,1+\frac{1}{\tau^2}\right),\tau^2|\boldsymbol{\beta},\boldsymbol{\lambda},\sigma^2,$$

$$\boldsymbol{D}^{(t)} \sim \text{IG}\left(\frac{p+1}{2},\frac{1}{\xi}+\frac{\boldsymbol{\beta}'\boldsymbol{M}_\lambda^{-1}\boldsymbol{\beta}}{2\sigma^2}\right). \quad (3)$$

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

2.3. Spike-and-Lasso Variable Selection Prior

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

$$\begin{aligned}\beta_j|\sigma^2,\tau_j^2,\gamma_j &\sim \gamma_j N\left(0,\sigma^2\tau_j^2\right)+(1-\gamma_j)N\left(0,\sigma^2c^2\right),\\ \tau_j^2 &\sim \exp(\lambda^2/2),\ \gamma_j\sim\text{Ber}(\theta),\\ \lambda^2 &\sim \text{Ga}(r,d),\ \theta\sim\text{Beta}(a,b).\end{aligned}$$

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

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

$$\boldsymbol{\beta}|\sigma^2,\tau^2,\gamma,\boldsymbol{D}^{(t)} \sim N_p\left(\left(\boldsymbol{S}_1^{(t)}+\boldsymbol{M}^{-1}\right)^{-1}\boldsymbol{S}_2^{(t)},\sigma^2\left(\boldsymbol{S}_1^{(t)}+\boldsymbol{M}^{-1}\right)^{-1}\right),$$

$$\sigma^2|\boldsymbol{\beta},\boldsymbol{\tau}^2,\lambda^2,\boldsymbol{D}^{(t)} \sim \text{IG}\left(\frac{nt+p}{2},\frac{\left(\boldsymbol{S}_3^{(t)}+\boldsymbol{\beta}'\boldsymbol{S}_1^{(t)}\boldsymbol{\beta}-2\boldsymbol{\beta}'\boldsymbol{S}_2^{(t)}\right)+\boldsymbol{\beta}'\boldsymbol{M}^{-1}\boldsymbol{\beta}}{2}\right),$$

$$\lambda^2|\boldsymbol{\beta},\sigma^2,\tau^2,\boldsymbol{D}^{(t)} \sim \text{IG}\left(p+r,\frac{\sum_{j=1}^p\gamma_j\tau_j^2}{2}+d\right),$$

$$\theta \sim \text{Beta}\left(a+\sum_{j=1}^{p}\gamma_j, b+p-\sum_{j=1}^{p}\gamma_j\right),$$
$$\frac{1}{\tau_j^2}|\gamma_j=1,\boldsymbol{\beta},\sigma^2,\lambda^2,\boldsymbol{D}^{(t)} \sim \text{Inv}-\text{Gaussian}\left(\sqrt{\frac{\lambda^2\sigma^2}{\beta_j^2}},\lambda^2\right),$$
$$\tau_j^2|\gamma_j=0,\boldsymbol{\beta},\sigma^2,\lambda^2,\boldsymbol{D}^{(t)} \sim \exp(\lambda^2/2),$$
$$\gamma_j|\boldsymbol{\beta},\sigma^2,\tau^2,\theta,\boldsymbol{D}^{(t)} \sim \text{Ber}(\eta_j),$$
$$\eta_j = \frac{\theta\left(\sigma^2\tau_j^2\right)^{-\frac{1}{2}}\exp\left(-\frac{\beta_j^2}{2\sigma^2\tau_j^2}\right)}{\theta\left(\sigma^2\tau_j^2\right)^{-\frac{1}{2}}\exp\left(-\frac{\beta_j^2}{2\sigma^2\tau_j^2}\right)+(1-\theta)\left(c^2\right)^{-\frac{1}{2}}\exp\left(-\frac{\beta_j^2}{2c^2}\right)}. \quad (4)$$

where $\boldsymbol{M} = \text{diag}(w_1,\ldots,w_p)$ with $w_j = \tau_j^2$ if $\gamma_j = 1$; $w_j = c^2$ otherwise. The computational issue arises from the Gibbs sampling step of $\boldsymbol{\beta}$ that incurs a complexity of $O(p^3)$, as well as due to updating γ_j's, $j = 1,...,p$ resulting in high auto-correlation. Updating subsets of $\boldsymbol{\beta}$ parameters in smaller blocks may be an option. However, shrinkage or variable selection priors generally do not allow closed form marginal distributions for such blocks of regression parameters. Again, the sequential nature of Gibbs sampling prohibits updating blocks of parameter $\boldsymbol{\beta}$ in parallel. The dynamic feature partitioning strategy developed in the next section will provide a solution to this computational challenge by parallelizing the approximate Bayesian computation of blocks of parameters into different processors.

3. Dynamic Feature Partition in High-Dimensional Regression

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

3.1. Relevant Notations and Details of DFP

Let $\boldsymbol{\Theta} = \{\theta_1,\ldots,\theta_q\}$ represent the parameter space with q parameters, which is bigger than p (the no. of predictors), since the parameter space includes the error variance σ^2 as well as latent variables from the data augmentation procedures described in Section 2. We further assume

1. q is fixed over time, that is, the parameter space does not change with the arrival of new data shards.
2. At each time point, the posterior distribution of the parameters $\boldsymbol{\Theta}$ depends on the data only through lower dimensional functions of $\boldsymbol{D}^{(t)}$ which are referred to as sufficient statistics. More formally, $\boldsymbol{S}^{(t)}$ is a vector of sufficient statistics for $\boldsymbol{\Theta}$ if $\boldsymbol{\Theta}|\boldsymbol{D}^{(t)}$ has the same distribution as $\boldsymbol{\Theta}|\boldsymbol{S}^{(t)}$. Denoting $f(\boldsymbol{\Theta}|\boldsymbol{D}^{(t)})$ as the full posterior distribution of $\boldsymbol{\Theta}$, this assumption implies that $f(\boldsymbol{\Theta}|\boldsymbol{D}^{(t)}) = f(\boldsymbol{\Theta}|\boldsymbol{S}^{(t)})$.

Referring to Section 2, both 1 and 2 are valid for linear regression models with shrinkage prior distributions or discrete mixture variable selection priors on coefficients.

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

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

3.2. DFP Algorithm for Online Approximate MCMC Inference

The DFP algorithm provides an online approximate MCMC sampling based on dynamically adaptive parameter partitions and their point estimates constructed sequentially over time. The algorithm begins by initializing the point estimate of $\boldsymbol{\Theta}$ (call it $\widehat{\boldsymbol{\Theta}}^{(0)}$) at some default value and initializing sufficient statistics $\boldsymbol{S}^{(0)}$ at $\boldsymbol{0}$. When new data shard $\boldsymbol{D}_t$ arrives at time t $(t = 1,...,T)$, sufficient statistics $\boldsymbol{S}^{(t)}$ are updated as a function of $\boldsymbol{S}^{(t-1)}$ and $\boldsymbol{D}_t$, denoted as $\boldsymbol{S}^{(t)} = g(\boldsymbol{S}^{(t-1)},\boldsymbol{D}_t)$. In the examples of Section 2, $g(\cdot)$ is implicitly defined through the three equations, $\boldsymbol{S}_1^{(t)} = \boldsymbol{S}_1^{(t-1)} + \boldsymbol{X}_t'\boldsymbol{X}_t$, $\boldsymbol{S}_2^{(t)} = \boldsymbol{S}_2^{(t-1)} + \boldsymbol{X}_t'\boldsymbol{y}_t$ and $\boldsymbol{S}_3^{(t)} = \boldsymbol{S}_3^{(t-1)} + \boldsymbol{y}_t'\boldsymbol{y}_t$. The dynamic partitioning scheme (described later) then updates partitions of the set of parameters and creates new partitions $\mathcal{G}^{(t)}$ at time t. The DFP algorithm then proceeds by sampling from the DFP pseudo conditional posteriors at time t in parallel. If the DFP pseudo conditional posteriors are in closed form, one may consider block updating of $\boldsymbol{\Theta}_{G_l^t}$ from $f\left(\boldsymbol{\Theta}_{G_l^t}|\widehat{\boldsymbol{\Theta}}_{-G_l^t}^{(t-1)},\boldsymbol{S}^{(t)}\right)$. Otherwise, the sampling in each partition proceeds by employing a Gibbs sampler with smaller blocks of parameters in the lth partition. More specifically, $\theta_j \in \boldsymbol{\Theta}_{G_l^t}$ is

updated by drawing S (a moderately large number, taken to be 500 in Section 4) approximate MCMC samples $\tilde{\theta}_j^{(1,t)}, ..., \tilde{\theta}_j^{(S,t)}$ from $f\left(\theta_j | \boldsymbol{\Theta}_{G_l^t \setminus \{j\}}, \widehat{\boldsymbol{\Theta}}_{-G_l^t}^{(t-1)}, \boldsymbol{S}^{(t)}\right)$, where the tilde emphasizes the fact that we are sampling from an approximation to the full conditional distribution, instead of the full conditional distribution. Often this distribution depends on a lower dimensional function of $\boldsymbol{\Theta}_{G_l^t \setminus \{j\}}$, $\widehat{\boldsymbol{\Theta}}_{-G_l^t}^{(t-1)}$ and $\boldsymbol{S}^{(t)}$, as we will see in Sections 4.1–4.3. Once S approximate MCMC samples are drawn from DFP pseudo conditional posteriors fairly rapidly, we use these samples to construct the point estimates of parameters at time t, given by $\widehat{\boldsymbol{\Theta}}^{(t)}$. In our exposition, we use the mean of the S samples $\tilde{\theta}_j^{(1,t)}, ..., \tilde{\theta}_j^{(S,t)}$ to construct $\widehat{\theta}_j^{(t)}$. The theoretical results in the supplementary material prove desirable performance of the proposed algorithm when the sequence of estimators $\widehat{\boldsymbol{\Theta}}^{(t)}$ is consistent in estimating the true parameters as $t \to \infty$. In practice, we found this assumption can be validated empirically for implementation of DFP in Sections 4.1–4.3. In fact, the trace-plots of $\widehat{\boldsymbol{\Theta}}^{(t)}$ corresponding to representative regression parameters in Section 4 show convergence around the true data generating parameters. Efficient updating of DFP pseudo conditional posteriors using the sufficient statistics and point estimates of parameters from the previous time point lead to scalable inference.

Partitioning schemes: As discussed before, an efficient partitioning of parameter indices $\mathcal{G}^{(t)}$ at the tth time is achieved by heavily exploiting the nature of the model and prior distributions. We believe that a general partitioning scheme that is applicable to any model and/or any prior distribution is unappealing since it will not be able to fully exploit the specific features of the model and prior distributions. Since the main focus of this article is on Bayesian shrinkage and variable selection priors in high-dimensional linear regression models, broadly two different partitioning schemes are proposed, one for the model (1) with shrinkage priors and the other for spike-and-slab priors.

(A) Partitioning algorithm for shrinkage priors: Referring to the discussion in Sections 2.1 and 2.2, the computational bottleneck mainly arises due to sampling from the posterior full conditional of $\boldsymbol{\beta}$. Therefore, in the course of developing a partitioning strategy for the set of parameters in Equation (1) with shrinkage priors, the main focus rests on how to partition $\boldsymbol{\beta}$ into blocks of sub-vectors with a minimal loss of information due to separately updating these blocks residing in different subset partitions from their DFP full conditionals. To this end, we set the maximum size of each block of $\boldsymbol{\beta}$ residing in different partitions to be less than or equal to M at every time to keep a control on the computational complexity. M is user defined and its choice depends on the available computational resources. In our empirical investigations with high-dimensional linear regression with Bayesian shrinkage priors, we find $M = 100$ to be sufficient and provide discussion on how the choice of small values of M affects inference. Thereafter we envision the problem of partitioning $\boldsymbol{\beta}$ at time t as a graph partitioning problem. To elaborate, at time t, for $j, j' \in \{1, ..., p\}$, let the sample correlation between S iterates of β_j and $\beta_{j'}$ from time $(t-1)$ following the DFP algorithm, given by $\{\tilde{\beta}_j^{(s,t-1)}\}_{s=1}^S$ and $\{\tilde{\beta}_{j'}^{(s,t-1)}\}_{s=1}^S$, be denoted by $r_{j,j'}$. A graph is constructed with nodes as the predictor indices $\{1, ..., p\}$ and an edge between two nodes j, j' if $r_{j,j'} > c$ where $c \in (0, 1)$. Our proposed scheme constructs different graphs in this manner corresponding to different choices of the cutoff $c \in$ `seq(0.01,0.99,by=0.01)`. Thereafter we find connected components of all these constructed graphs and look for the smallest value of c (say c^*) for which the size of all connected components are less than M. Such an implementation is readily achieved by the functionalities in the `igraph` package in R. Let there be b_t connected components corresponding to the cut-off value c^* at time t, which we denote by $\{\mathcal{P}_1^{(t)}, ..., \mathcal{P}_{b_t}^{(t)}\}$. These b_t connected components at time t are recognized as partitions of the indices $\{1, ..., p\}$ and β_j's corresponding to different connected components go to different partitions of the parameter sets at time t. Thus, $\boldsymbol{\beta}_{\mathcal{P}_1^{(t)}}, ..., \boldsymbol{\beta}_{\mathcal{P}_{b_t}^{(t)}}$ go to different subsets in the implementation of DFP at time t. Since the data augmentation approaches in Sections 2.1 and 2.2 introduce latent vectors ($\boldsymbol{\tau}^2$ in Section 2.1, $\boldsymbol{\lambda}$ and $\boldsymbol{\nu}$ in Section 2.2) related to $\boldsymbol{\beta}$, we either keep all elements of a latent vector together in one partition or divide a latent vector into blocks with indices $\{\mathcal{P}_1^{(t)}, ..., \mathcal{P}_{b_t}^{(t)}\}$ and send the latent vector with indices $\mathcal{P}_k^{(t)}$ to the same parameter subset where $\boldsymbol{\beta}_{\mathcal{P}_k^{(t)}}$ lies. Variance σ^2 and other hierarchical parameters are kept together in a separate partition. Since a partition involves blocks of $\boldsymbol{\beta}$ with size at most M, sampling them together from their DFP full conditionals incurs complexity at most of $O(M^3)$. We later empirically establish that the subsets of parameters constructed by the above partitioning scheme stabilize over time. In fact, our empirical analysis also demonstrates that the optimal value c^* also stabilizes as time progresses.

(B) Partitioning algorithm for Spike and Lasso priors: Since the Spike and Lasso example in Section 2.3 involves coefficients belonging to one of the two mixture components at every iteration of the posterior sampling, the parameter partitioning scheme adopted for shrinkage priors appears to be less efficient here. Instead, we propose a dynamic partitioning scheme of the parameter space by tacitly exploiting the natural partitioning of the $\boldsymbol{\beta}$ parameters and associated latent vector $\boldsymbol{\tau}$ into important and unimportant components. Define $\boldsymbol{\Theta}_{1t} = \{(\beta_j, \tau_j^2) : \widehat{\gamma}_j^{(t-1)} = 1\}$ and $\boldsymbol{\Theta}_{2t} = \{(\beta_j, \tau_j^2) : \widehat{\gamma}_j^{(t-1)} = 0\}$, where $\widehat{\gamma}_j^{(t-1)} \in \{0, 1\}$ corresponds to the point estimate of γ_j at time $(t-1)$. Thereafter our partitioning scheme suggests keeping the entire $\boldsymbol{\Theta}_{1t}$ in one partition and dividing $\boldsymbol{\Theta}_{2t}$ into subsets, with each subset of $\boldsymbol{\Theta}_{2t}$ containing (β_j, τ_j^2) for a single j. Additionally, all γ_j's are kept in the same partition and $\lambda^2, \sigma^2, \theta$ in another partition. Since spike-and-slab priors are typically employed to recover $\boldsymbol{\beta}$ parameters which are sparse in nature in the truth, $\boldsymbol{\Theta}_{1t}$ is expected to be of small to moderate size with cardinality much smaller than p as time progresses. Thus, updating $(\beta_j : \beta_j \in \boldsymbol{\Theta}_{1t})'$ together requires computational complexity of order $|\boldsymbol{\Theta}_{1t}|^3 << p^3$. On the other hand, β_j's for $j \in \boldsymbol{\Theta}_{2t}$ are updated individually without incurring any notable computational burden. A similar strategy is followed when the double exponential slab distribution in the Spike-and-Lasso prior is replaced by any other distribution.

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

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

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

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

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

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

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

Competitors. The performance of DFP is compared with a set of competitors suitable for high-dimensional linear regression models. We specifically compare with (a) batch MCMC that draws S MCMC samples from the full conditional distributions at every time point with the full data $\boldsymbol{D}^{(t)}$ through time t at disposal; and (b) conditional density filtering (C-DF) (Guhaniyogi, Qamar, and Dunson 2013). Batch MCMC offers the "gold standard" for ordinary Gibbs sampling that uses the full data $\boldsymbol{D}^{(t)}$ at time t. At time t, batch MCMC initializes the MCMC chain at the last iterate in time $(t - 1)$. In examples (i)–(iii), the conditional posterior distributions depend on the data through lower dimensional sufficient statistics, and hence batch MCMC only stores and propagates the sufficient statistics to update the conditional distributions in successive time points. Conditional density filtering is proposed in the same vein as DFP with an important difference. While DFP proposes dynamic partitioning of the set of parameters, C-DF works with parameter partitions fixed over time. We find that the naive implementation of C-DF demonstrates considerably inferior performance than DFP. To make C-DF more competitive, we employ a version of C-DF that draws samples from parameter partitions sequentially rather than in parallel, to be able to use samples from one partition to construct more accurate point estimates for the other partitions at every time. Such an implementation of C-DF considerably improves its performance, though at the expense of added computational burden. Overall, comparison with this improved version of C-DF will demonstrate the advantages of dynamic partitioning over fixed partitioning as a tool to provide a better approximation to the full posterior distribution of parameters. Online variational inference provides an alternate strategy to draw approximate inference in presence of big data and a large number of parameters. However, in the absence of any open-source code for online variational inference in high-dimensional linear regression, we refrain from employing it as a competitor. Finally, we compare our approach with a variant of the Sequential Monte Carlo (SMC) approach. As discussed in the introduction, most of the developments in SMC and PL algorithms have taken place in the high-dimensional state-space models and they do not assume seamless extensions to high-dimensional static parametric models with p as high as 5000. Therefore, we adapt the recent sub-sampled SMC approach outlined in Gunawan et al. (2018) to our setting. Note that the approach in Gunawan et al. (2018) is designed for the scenario when the entire dataset is available to the user. To adapt it to the streaming data context, we employ a data annealing approach instead of the temperature annealing approach used by the authors. Our data annealing approach performs data sub-sampling from the entire data $\boldsymbol{D}^{(t)}$ when a new batch arrives at time t and uses the sub-sampling density approximation as well as the Hamiltonian Monte-Carlo technique for efficient drawing of high-dimensional Monte Carlo samples. This approach uses the entire data set (up to time t) $\boldsymbol{D}^{(t)}$ in drawing SMC samples at time t, and strictly speaking is not an online Bayesian competitor. Nevertheless, it can demonstrate the state-of-the-art performance from SMC which will be helpful in assessing the performance of DFP. We refer to this approach as sub-sampled SMC (SSMC).

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

Assessing predictive inference with DFP and competitors. To measure the predictive performance of competitors, we report:

(a1) mean squared prediction error (MSPE); (a2) Interval score (Gneiting and Raftery 2007) of the 95% predictive interval; (a3) coverage of the 95% predictive interval, and (a4) average run time for each batch or shard. Note that (a1) demonstrates the performance in terms of point prediction, while (a2) and (a3) show how well calibrated the predictions turn out to be. Finally, (a4) helps readers gauge the computation time vis-a-vis accuracy of the competitors. At time $(t-1)$, evaluations of predictive performance metrics (a1)–(a3) are based on the data shard observed at time t. All results are based on averages over 10 independent replications. All computation times are based on an R implementation in a cluster computing environment with three interactive analysis servers, 32 cores each with the Dell PE R820: 4x Intel Xeon Sandy Bridge E5-4640 processor, 16GB RAM and 1TB SATA hard drive.

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

4.1. DFP with Bayesian Lasso

We consider the first application of DFP with the popular Bayesian Lasso (Park and Casella 2008) shrinkage prior on high-dimensional predictor coefficients. Details of the Bayesian Lasso prior and challenges regarding posterior computation with the Bayesian Lasso prior has already been presented in Section 2.1.

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

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

where the lth partition, $l = 1, .., b_t$ consists of $2m_l$ parameters (m_l is also a function of t) and $i^{(t)}_{m_1+\cdots+m_{l-1}+1}, ..., i^{(t)}_{m_1+\cdots+m_l} \in \{1, ..., p\}$ correspond to the indices of predictor coefficients and latent variables belonging to the lth partition at time t. Let at time t, $\boldsymbol{\beta}_l = \left(\beta_{i^{(t)}_{m_1+\cdots+m_{l-1}+1}}, ..., \beta_{i^{(t)}_{m_1+\cdots+m_l}}\right)'$, $\boldsymbol{\tau}_l^2 = \left(\tau^2_{i^{(t)}_{m_1+\cdots+m_{l-1}+1}}, ..., \tau^2_{i^{(t)}_{m_1+\cdots+m_l}}\right)'$, $\boldsymbol{M}_{\tau,l} = \text{diag}(\boldsymbol{\tau}_l^2)$ and $\boldsymbol{\beta}_{-l}$ be the vector of all β_j's except those included in $\boldsymbol{\beta}_l$. $\widehat{\boldsymbol{\beta}}_l^{(t-1)}$, $\widehat{\boldsymbol{\beta}}_{-l}^{(t-1)}$, $\widehat{\boldsymbol{\tau}}_l^{2(t-1)}$ are the point estimates of $\boldsymbol{\beta}_l, \boldsymbol{\beta}_{-l}, \boldsymbol{\tau}_l^2$, respectively, at time $(t-1)$. $\boldsymbol{S}^{(t)}_{1,l}$ and $\boldsymbol{S}^{(t)}_{2,l}$ are analogously defined. Also assume $\boldsymbol{S}^{(t)}_{1,l,-l} = \boldsymbol{S}^{(t-1)}_{1,l,-l} + \boldsymbol{X}'_{t,l}\boldsymbol{X}_{t,-l}$, where $\boldsymbol{X}_{t,l}$ and $\boldsymbol{X}_{t,-l}$ are the sub-matrices of $\boldsymbol{X}_t$ corresponding to $\boldsymbol{\beta}_l$ and $\boldsymbol{\beta}_{-l}$, respectively. Section 4.1 of the supplementary material describes details of implementing of Algorithm 1 for the Bayesian Lasso.

Due to space constraint, density estimates for a few selected predictor coefficients are displayed at $t = 250, 500$. Since Simulation 1 is the most interesting scenario, posterior densities of a randomly chosen zero coefficient, a nonzero coefficient with a lower magnitude and a nonzero coefficient with a higher magnitude are presented in Figure 1. Posterior densities of the selected β_j's in the batch MCMC and DFP tend to show discrepancies in the earlier time points. These discrepancies diminish at $t = 500$, empirically validating the fact that approximate DFP draws converge to the full posterior distribution in time. This conclusion remains valid for Simulations 2 and 3.

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

We also present MSPE, coverage, interval score for the 95% predictive intervals and computation time in seconds per batch of the competing methods for Simulation 1 in Figure 1. Figures 3 and 4 highlight the same quantities for Simulations 2 and 3 respectively, except the computation time which is similar for competitors across the three simulations. Batch MCMC, being a batch method, is expected to converge faster. The predictive inference of DFP improves rapidly and becomes indistinguishable from batch MCMC within $t \approx 100 - 150$ for all three simulations. In contrast, the predictive performance of C-DF appears to be inferior to batch MCMC even at $t = 150$. To ensure that the faster decay in MSPE of DFP compared to C-DF can actually be attributed to dynamic construction of parameter subsets at each time, we explore three other versions of DFP for which we update partitions of the parameter set in every 10, 50, and 100 batches. We refer to them as lagged DFP with lag $= 10, 50, 100$, respectively. The regular DFP corresponds to lag $= 1$. The trace-plots of MSPE for the regular DFP (i.e., with lag$= 1$) along with lagged DFP for the Bayesian Lasso Model in the three simulation settings are shown in Figure 5.

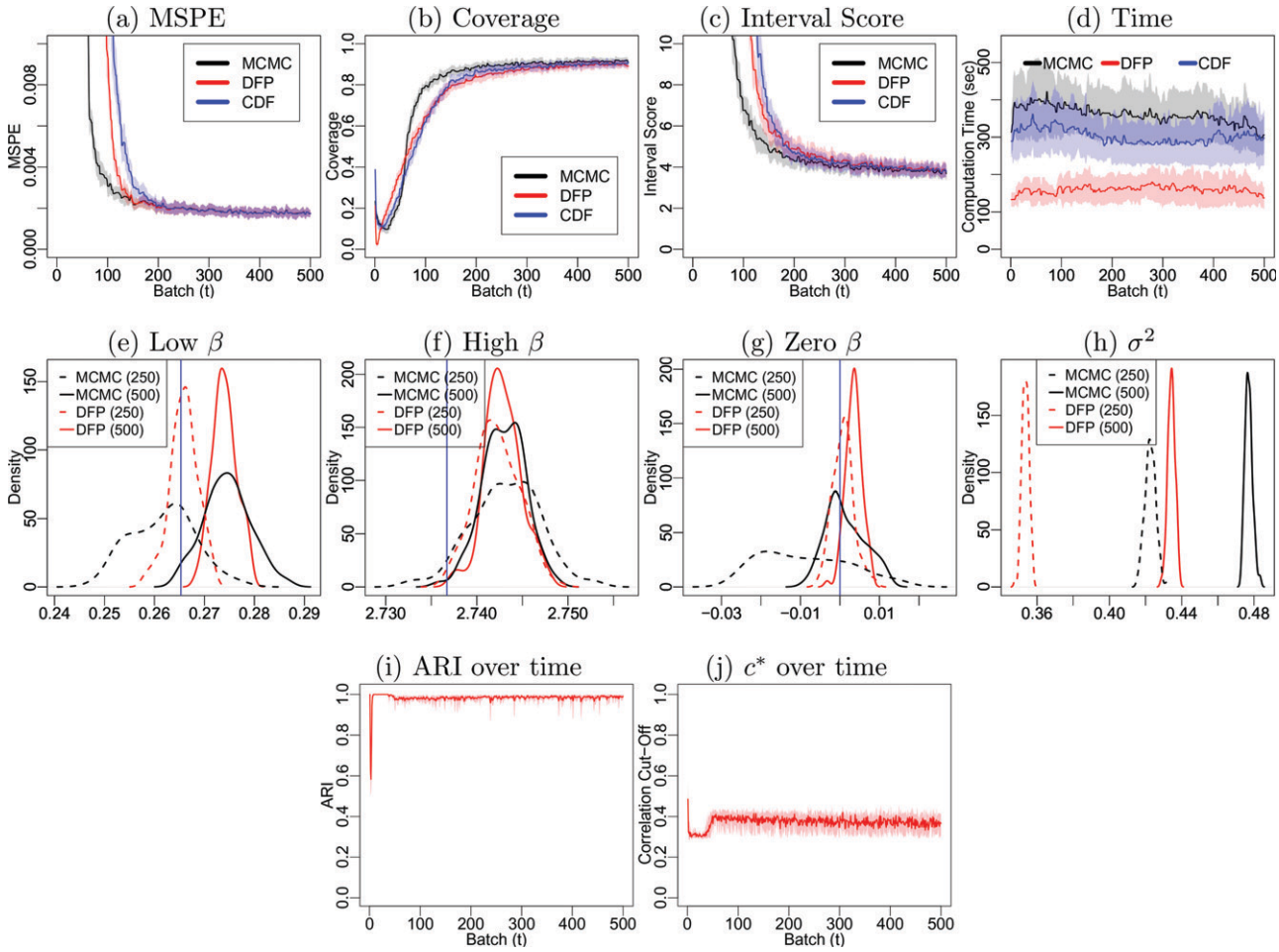

Figure 1. Performance measures for MCMC, DFP, and CDF in the case of Bayesian Lasso under the high and low sparse case are presented in the first row. Coverage and Interval scores are based on the average of the 95% predictive intervals. Confidence bands are based on repeating the analysis over 10 replications. The second row shows estimated densities of selected parameters at $t = 250$ and $t = 500$ for DFP and batch MCMC. Finally, third row presents the trace-plot of the ARI between partitions in two successive time points for DFP and the trace-plot for the optimal value c^* of DFP.

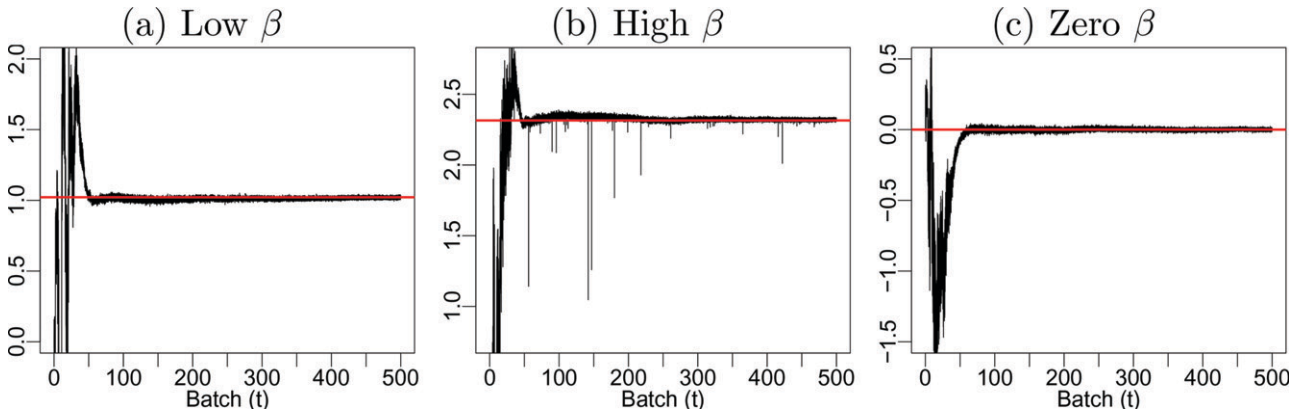

Figure 2. Trace-plots of $\widehat{\beta}_j^{(t)}$ for representative β_j parameters under DFP Bayesian Lasso implementation in Simulation 1. We present $\widehat{\beta}_j^{(t)}$ for a low signal, a high signal, and a zero signal in the truth. The horizontal line specifies the true value of a parameter.

As the value of lag increases, it takes more time for MSPE in the lagged DFP to stabilize. In fact, the figure shows that the MSPE for a lagged DFP with lag= 100 takes about 50 more data shards to stabilize compared to the MSPE of the regular DFP. Thus, dynamic partitioning learns posterior correlations among parameters accurately which yields a better approximation of the full posterior than C-DF or any other lagged version of DFP in the earlier time points.

The average MSPE, run time, coverage and interval scores of 95% predictive intervals over the last 100 time points for all the competitors are presented in Table 1. The results show that in all three simulations, DFP emerges as a computationally efficient

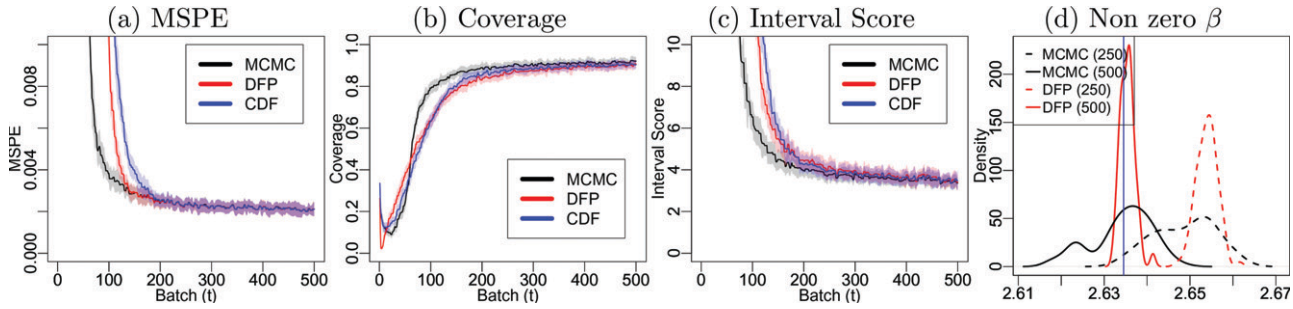

Figure 3. Performance measures for MCMC, DFP, and CDF for Bayesian Lasso under the sparse case (Simulation 2) are presented. Coverage and Interval scores are based on the average of the 95% predictive intervals. We also show estimated densities for a selected β_j at $t = 250$ and $t = 500$ for both batch MCMC and DFP.

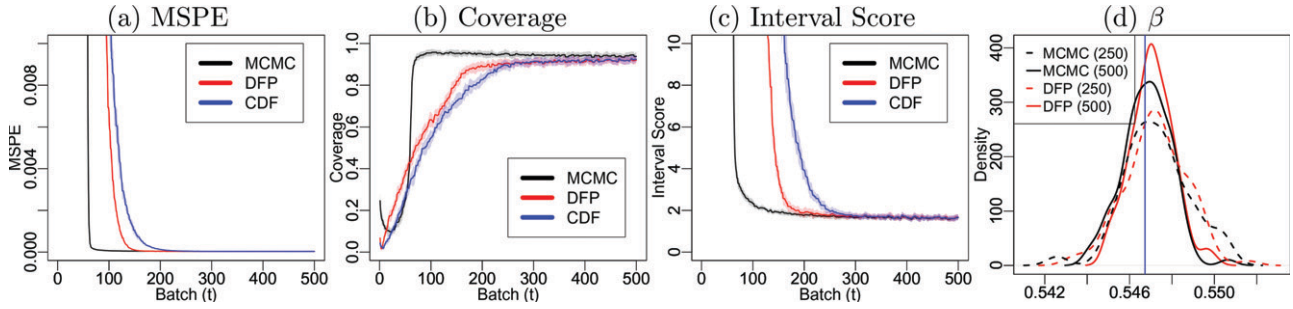

Figure 4. Performance measures for MCMC, DFP, and CDF for Bayesian Lasso under the dense case (Simulation 3). Coverage and Interval scores are based on the average of the 95% predictive intervals. Estimated densities of selected parameters at $t = 250$ and $t = 500$ for both batch MCMC and DFP are also added.

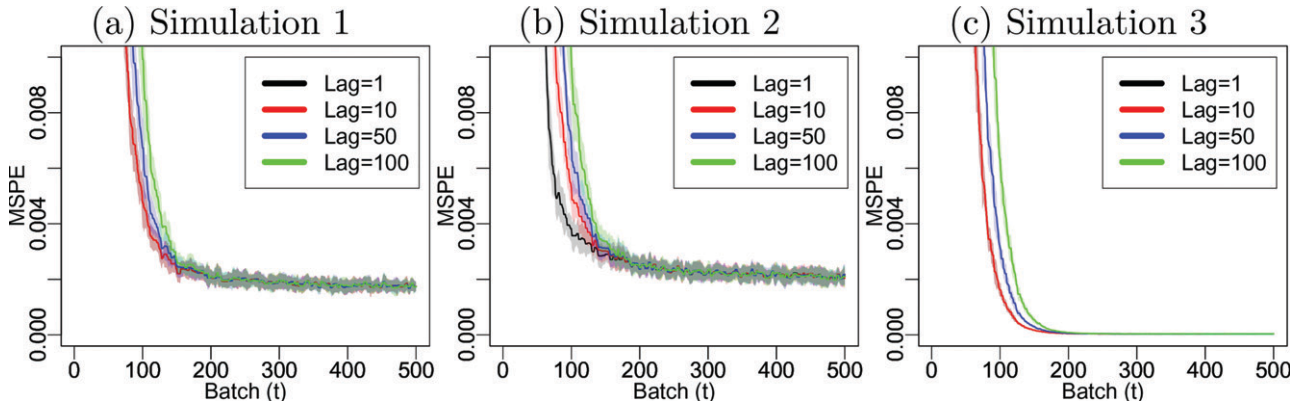

Figure 5. The trace-plots of MSPE for regular DFP (lag = 1) and lagged DFP with lag = 10, 50, 100 implemented using Bayesian Lasso in Simulations 1-3.

replacement for batch MCMC, both in terms of point prediction as well as characterizing predictive uncertainties. As mentioned earlier, naive implementation of C-DF demonstrates inferior predictive inference. An improved implementation of C-DF presented here, in contrast, loses appeal with minimal gain in computation time over batch MCMC. The SSMC approach also demonstrates similar inferential performance with DFP with a higher computation time.

Sensitivity to the choice of M. Our investigation reveals that for any choice of M, the mean squared prediction error (MSPE) starts decreasing as time progresses and finally stabilizes. It is also interesting to note that they stabilize at similar values for various choices of M. This is not surprising, since the posterior correlations between parameters become less important factors in prediction when sample size is much larger than the number of parameters. However, for a larger value of M, MSPE stabilizes much more rapidly over time. This is demonstrated for the Bayesian Lasso shrinkage prior with $M = 10$ and $M = 60$ under the three simulation settings, see Figure 6. We conclude that when inference is necessary at the earlier time points, one should perhaps adopt a larger choice of M. In contrast, when inference is only required at very large time points, one may construct a more efficient DFP algorithm with a smaller value of M.

4.2. DFP With Horseshoe

Our second application considers implementing DFP on the Horseshoe shrinkage prior (Carvalho, Polson, and Scott 2010). The full conditional distributions of parameters along with computational issues in implementing Gibbs sampling with the Horseshoe shrinkage prior are given in Section 2.2. The DFP algorithm is employed to incur computational benefits in situations with large p.

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

Algorithm 1: Dynamic Feature Partition

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

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

DFP$(\boldsymbol{D}_t, \mathcal{G}^{(t)}, \boldsymbol{S}^{(t-1)}, \widehat{\boldsymbol{\Theta}}^{(t-1)})$
begin
  /* Step 1: Update the partition of the set of parameters at time t: the partitioning schemes should ideally exploit the nature of the model and prior distributions. We propose partitioning schemes specific to the high-dimensional linear regression with shrinkage priors and spike and slab priors in Section 3, page 12 and 13. */
  $\mathcal{G}^{(t)} = \textbf{PartitionUpdate}\left(\tilde{\boldsymbol{\Theta}}^{(1,t-1)},\ldots,\tilde{\boldsymbol{\Theta}}^{(S,t-1)}\right)$
  /* step 2: Update Sufficient Statistics */
  Update $\boldsymbol{S}^{(t)} = g(\boldsymbol{D}_t, \boldsymbol{S}^{(t-1)})$
  /* step 3: Approximate Sampling for Parameter Blocks in Parallel */
  for $G_l^t \in \mathcal{G}^{(t)}$ **do**
    for $\theta_j \in \boldsymbol{\Theta}_{G_l^t}$ **do**
      for s=1:S **do**
        sample $\tilde{\theta}_j^{(s,t)} \sim f\left(\theta_j | \boldsymbol{\Theta}_{G_l^t \setminus \{j\}}, \boldsymbol{S}^{(t-1)}, \widehat{\boldsymbol{\Theta}}_{-G_l^t}^{(t-1)}\right)$
      end
    end
  end
  /* step 4: Update Estimates */
  for $G_l^t \in \mathcal{G}^{(t)}$ **do**
    for $\theta_j \in \boldsymbol{\Theta}_{G_l^t}$ **do**
      /* Compute relevant point estimates for the parameters from approximate MCMC samples. We consider the mean of the samples as the point estimate for each parameter */
      set $\widehat{\theta}_j^{(t)} \leftarrow \text{stat}\left(\tilde{\theta}_j^{(1,t)},\ldots,\tilde{\theta}_j^{(S,t)}\right)$
    end
  end
  return $\{\tilde{\boldsymbol{\Theta}}^{(1,t)},\ldots,\tilde{\boldsymbol{\Theta}}^{(S,t)}\}, \boldsymbol{S}^{(t)}, \widehat{\boldsymbol{\Theta}}^{(t)}$
end

subsets at time t given by

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

Table 1. Bayesian Lasso performance statistics for MCMC, CDF, DFP, and SSMC.

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

NOTES: Coverage and length are based on the average of the 95% credible predictive intervals in the last 100 batches. The subscript provides standard errors calculated over 10 replications.

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

Let $\boldsymbol{\beta}_l$ and $\boldsymbol{\lambda}_l$ be the vector of β_j's and λ_j^2's, respectively, corresponding to the lth partition. Define $\boldsymbol{S}_{1,l}^{(t)}$, $\boldsymbol{S}_{2,l}^{(t)}$, and $\boldsymbol{S}_{1,l,-l}^{(t)}$ as in Section 4.1. Let $\boldsymbol{M}_{\lambda,l} = \text{diag}(\boldsymbol{\lambda}_l)$ and $\boldsymbol{\beta}_{-l}$ be the β_js not contained in $\boldsymbol{\beta}_l$. A detailed implementation of DFP for the Horseshoe prior is described in Section 2.2 of the supplementary material.

Figure 7 presents dynamically evolving MSPE, coverage, interval score for the 95% predictive interval and computation time in seconds per batch of the competing methods for Simulation 1. As observed in Section 4.1, MSPE for DFP falls sharply as time progresses and becomes indistinguishable with the MSPE of batch MCMC after $t \approx 200-250$. While accurate point prediction is one of our primary objectives, characterizing uncertainty is of paramount importance given the recent development in the frequentist literature on characterizing uncertainties in high-dimensional regressions (Van de Geer et al. 2014; Zhang and Zhang 2014). Although Bayesian procedures provide an automatic characterization of uncertainty, the resulting credible intervals may not possess the correct frequentist coverage in nonparametric/high-dimensional problems (Szabó et al. 2015). An attractive adaptive property of the shrinkage priors, including Horseshoe, is that the lengths of the intervals automatically adapt between the signal and noise variables, maintaining close to nominal coverage. Approximate Bayesian inference with the DFP algorithm is found to preserve this desirable property of the Horseshoe prior. In fact, Figures 7, 8 and 9 show similar coverage and interval scores for DFP and batch MCMC as time progresses. This observation is further reinforced from Table 2 which demonstrates practically identical performances of batch MCMC, CDF, SSMC and DFP, with DFP having notably reduced computation time.

Density estimates for a few selected predictor coefficients are displayed at $t = 250, 500$. Since Simulation 1 is the most

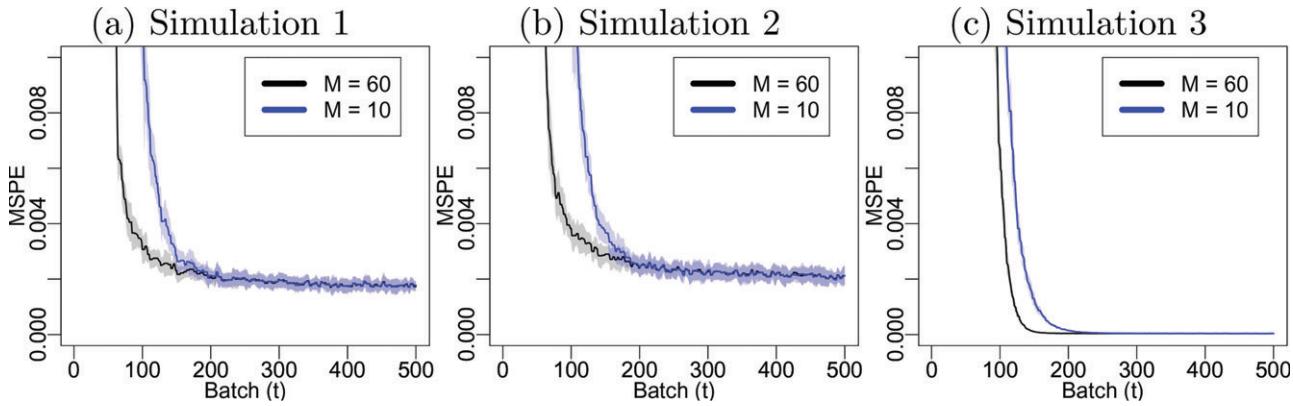

Figure 6. Trace-plots of MSPE for $M = 10, 60$ implemented using Bayesian Lasso prior in Simulations 1-3.

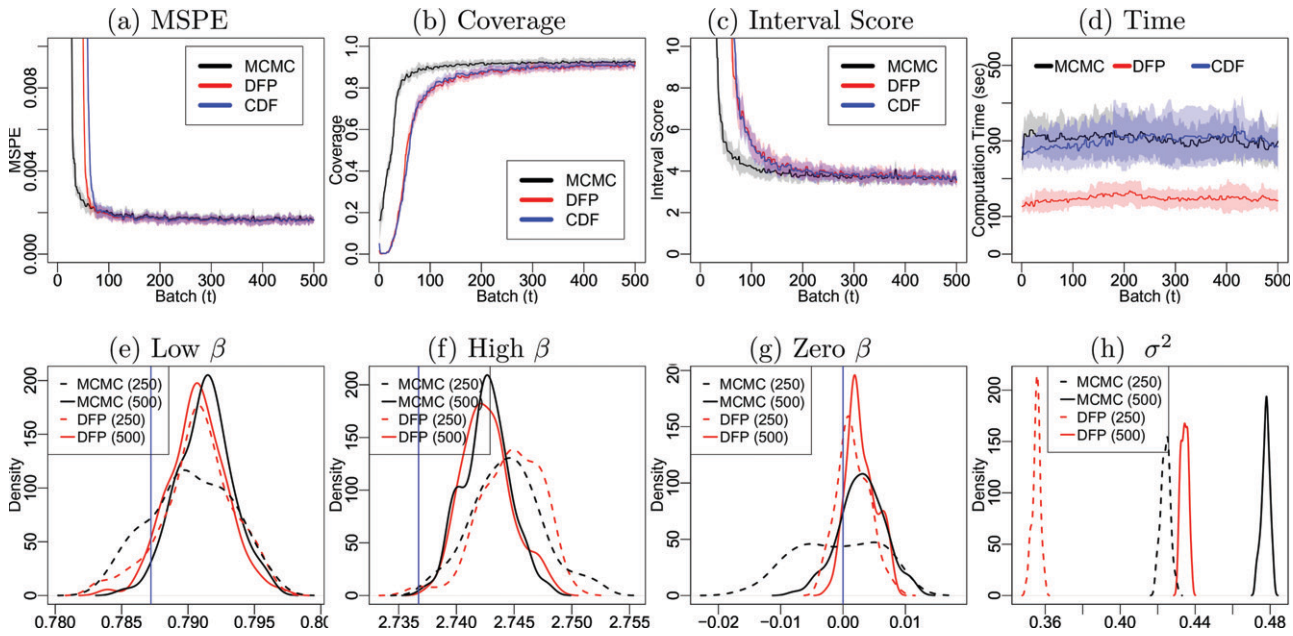

Figure 7. Performance measures for MCMC, DFP, and C-DF in the case of Horseshoe under the high and low sparse case (Simulation 1) are presented in the first row. Coverage and Interval scores are based on the average of the 95% predictive intervals. The second row shows estimated densities of selected parameters at $t = 250$ and $t = 500$ for both batch MCMC and DFP. Confidence bands are based on the analysis over 10 replications.

interesting scenario, posterior densities of a randomly chosen zero coefficient, a nonzero coefficient with a lower magnitude and a nonzero coefficient with a higher magnitude are presented in Figure 7. For nonzero coefficients, the density estimates seem to be similar in DFP and in batch MCMC, though DFP yields marginally narrower credible intervals than batch MCMC corresponding to zero coefficients. We refrain from adding any further discussion on the convergence of partitions or convergence of c^*, since the conclusion is very similar to Bayesian Lasso.

One fundamental advantage of the Horseshoe shrinkage prior over frequentist penalized optimization is its ability to accurately characterize parametric and predictive uncertainties without any user dependent choice of tuning parameters. However, it might lose this appeal due to its high computation time and inability to provide rapid inference with big n and p. DFP applied to the Horseshoe prior solves the computational bottleneck for big n and p, perhaps offering wider applicability to the Horseshoe prior in regression problems at a much larger scale. We expect similar conclusions to hold for other state-of-the-art shrinkage priors such as, the Generalized Double Pareto (Armagan, Dunson, and Lee 2013) and the normal gamma (Griffin et al. 2010) prior distributions.

4.3. Spike-and-Lasso

Since spike-and-slab prior distributions are primarily designed to identify important variables in sparse high-dimensional regressions, we investigate DFP with the Spike and Lasso prior for Simulations 1 and 2. Again, Section 4.3 of the supplementary material details out the implementation of Algorithm 1 of Spike & Lasso prior. Figure 10 presents the dynamic progression of various performance metrics for DFP, batch MCMC and C-DF over $T = 500$ time points. Unlike Sections 4.1 and 4.2, the operating characteristics of the Spike-and-Lasso applied to all three competitors take longer time to stabilize. This is not surprising, given that batch MCMC with spike and slab mixture priors is known to offer less accurate performance with a smaller sample size due to the high correlation between various γ_j's. As before, DFP approximates batch MCMC accurately in terms of

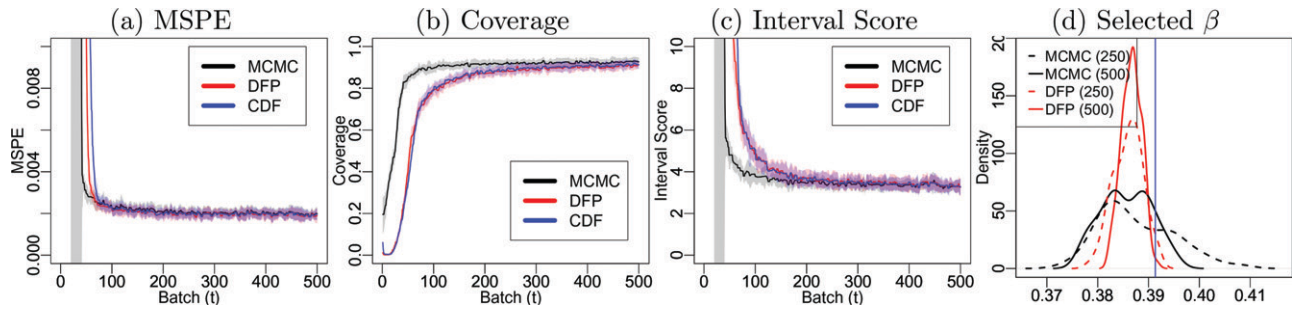

Figure 8. Performance measures for MCMC, DFP, and C-DF for Horseshoe under the sparse case (Simulation 2) are presented. Coverage and Interval scores are based on the average of the 95% predictive intervals. We also show estimated densities of a selected β_j at $t = 250$ and $t = 500$ for both batch MCMC and DFP.

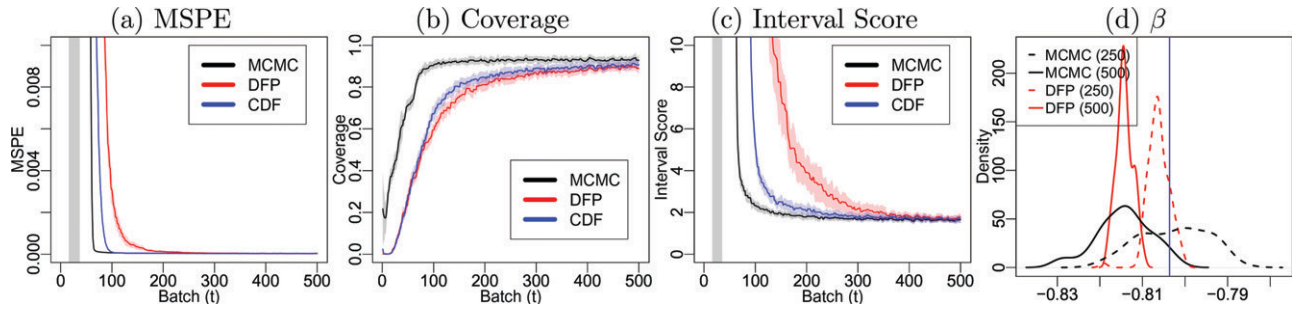

Figure 9. Performance measures for MCMC, DFP, and C-DF for Horseshoe under the dense case (Simulation 3) are presented. Coverage and Interval scores are based on the average of the 95% predictive intervals. We also show estimated densities of a selected β_j at $t = 250$ and $t = 500$ for both batch MCMC and DFP.

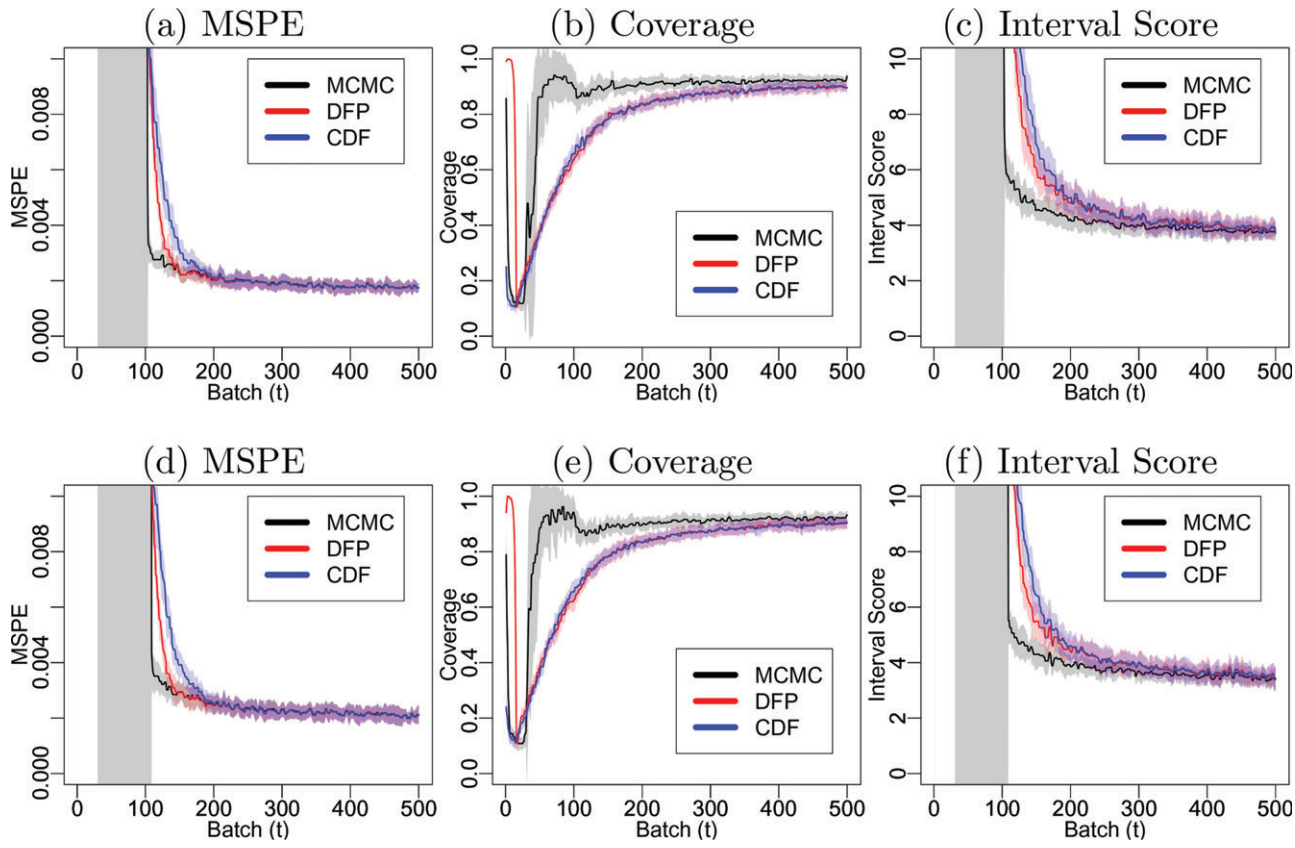

Figure 10. Performance measures for MCMC, DFP, and C-DF with the Spike-and-Lasso prior under Simulations 1 (1st row) and 2 (second row). Coverage and interval scores are based on the average of the 95% predictive intervals.

the operating characteristics. In fact, Table 3 shows practically indistinguishable performance of DFP and batch MCMC, while C-DF yields marginally larger interval scores even at latter time points. SSMC continues to show competitive performance with a much higher computation time compared to DFP. DFP dynamically learns the partition based on $\boldsymbol{\Theta}_{1t}$ and $\boldsymbol{\Theta}_{2t}$. Since we

Table 2. Horseshoe performance statistics for MCMC, C-DF, SSMC, and DFP.

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

NOTES: Coverage and interval scores are based on the average of the 95% credible predictive intervals of the last 100 batches. Subscripts provide standard errors over 10 simulations.

Table 3. Spike and Lasso performance statistics for MCMC, CDF, SSMC, and DFP.

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

NOTE: MSPE, coverage and interval scores are based on the average of the 95% credible predictive intervals for the last 100 batches.

consider sparse examples, the cardinality of the set $\boldsymbol{\Theta}_{1t}$ is never large, and hence the parameters therein can be updated quickly. Our detailed investigation also reveals that even a large number of partitions of $\boldsymbol{\Theta}_{2t}$ does not compromise the accuracy of the inference and prediction. This helps to accrue substantial gains in computation time for DFP compared to its competitors, as demonstrated in Table 3. In contrast, C-DF fixes the partitions in the beginning and is unable to leverage the information of the zero and nonzero β_j's as the approximate posterior sampling progresses.

Representative posterior densities of β_j's from DFP and batch MCMC (presented in Figure 11) are centered around the truth and have similar tails. Both Simulations 1 and 2 involve high sparsity, resulting in the posterior density of θ centered at a small value. Again there is a considerable agreement in the posterior densities of θ from DFP and batch MCMC. Finally, posterior densities of σ^2 for DFP and batch MCMC are found to differ by a small margin from the truth. The trace-plots of $\widehat{\beta}_j^{(t)}$ for representative coefficients with zero, low and high signals in the truth are also shown in Figure 12 and they are found to converge to the true parameter values. Finally, we explore how the partitions evolve dynamically and observe that the ARI between partitions at two successive time points quickly converges to 1 with time (see Figure 12).

4.4. Sensitivity to the Choice of S

One of the important ingredients in the development of DFP is the choice of the number of Monte Carlo samples S at every time and it is instructive to see the effect on inference with different choices of S. The simulation section presents results of DFP with $S = 500$. To assess the sensitivity to the choice of S in our simulations, we compute DFP after moderately perturbing S. Table 4 presents the predictive inference with DFP for $S = 500, 750, 1000$ in the different simulation cases with the Bayesian Lasso prior. The results show practically indistinguishable inference with different choices of S, with $S = 750$ and $S = 1000$ naturally incurring much more computational cost. In our experience, the inference can be marginally improved with much larger choices of S, though such choices practically diminish any computational advantage of DFP.

5. Application to Financial Stock Database

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

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

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

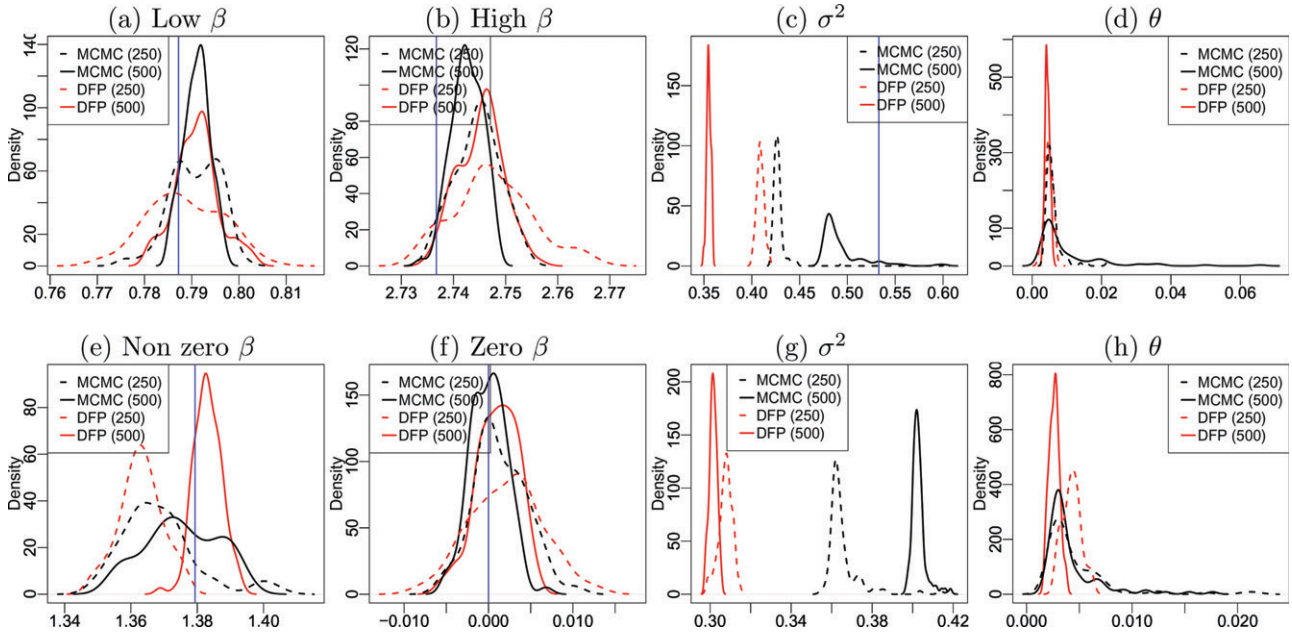

Figure 11. Estimated densities for a few selected β_js, σ^2 and θ at $t = 250$ and $t = 500$. The first row presents results for Simulation 1 while the second row demonstrates performance of DFP in Simulation 2.

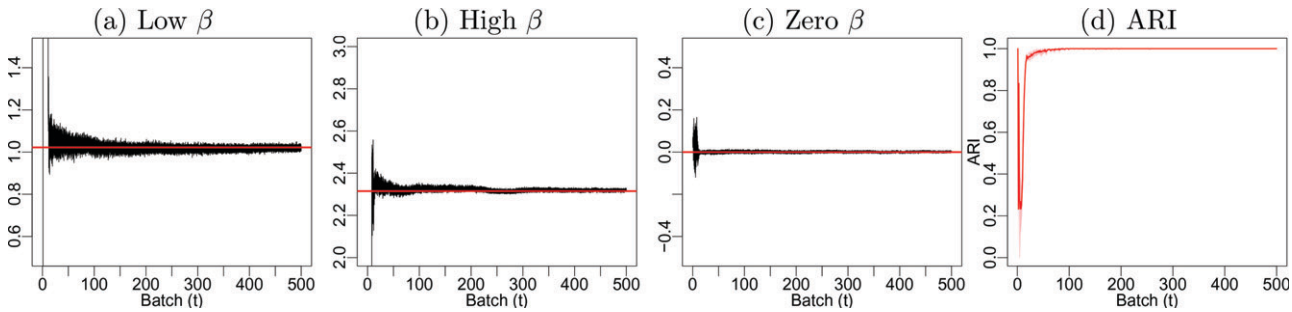

Figure 12. Trace-plots for $\widehat{\beta}_j^{(t)}$ for representative parameters in DFP Spike-and-Lasso implementation under Simulation 1. We include plots for representative predictor coefficients with low signal, high signal and zero signal in the truth. The horizontal line specifies the true value of the parameters. The left most column shows the trace-plot of the ARI for the parameter set partitions at two successive time points.

Figure 13 tracks the progression of MSPE, interval score and coverage of 95% predictive intervals for both DFP and batch MCMC as more batches are processed. At time t, the predictive inference is assessed with the data shard obtained at time $t + 1$. Similar to simulation studies, the behavior of DFP in the early batches is somewhat erratic due to the inaccurate posterior approximation in the initial phase of the algorithm, though it stabilizes as more data shards arrive. Furthermore, the performances of the competitors become closer as time progresses, with batch MCMC demonstrating marginally superior performance at higher time points. While batch MCMC runs 500 iterations per batch in 18.35 seconds, DFP finishes 500 iterations per batch in 0.40 seconds. Such a dramatic improvement in computation time can be attributed to efficient partitioning of the parameter space as well as parallel inference on parameter partitions at each time.

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

To ensure more reliable inference from DFP and the "gold standard" batch MCMC for variable selection, we run both these competitors 10 more times on the dataset of interest. In these 10 runs, the data is divided into 163 shards with the first shard having 20% observations, and the rest 162 shards all approximately equal. We observe that feeding more data early on leads to reliable variable selection with minimal variation between different runs. To provide concrete evidence on this observation, we refer to Table 5 which presents all predictors identified by either DFP or batch MCMC in any of the 10 runs. The table also records the number of times among the 10 runs

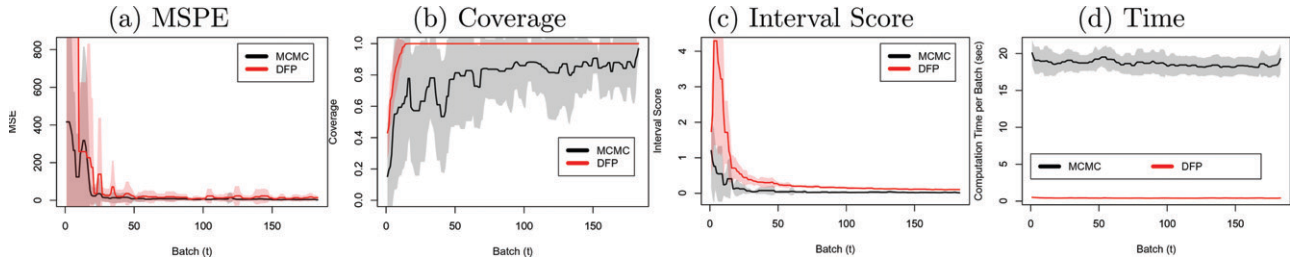

Figure 13. Performance measures for MCMC and DFP. MSPE, coverage, and interval scores for 95% predictive intervals are presented. Confidence bands (in a lighter color) are calculated by observing the variations of these metrics over 10 permutations.

Table 4. Bayesian Lasso performance statistics for DFP with $S = 500, 750,$ and 1000.

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

NOTES: Coverage and length are based on the average of the 95% predictive intervals on the last 100 batches. The subscript provides standard errors calculated over 10 replications.

Table 5. Number of times a stock is selected under DFP and MCMC out of 10 runs of both methods.

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

they are identified as influential. It shows that the number of times a predictor is selected by either batch MCMC or DFP is very close to 0 or 10, indicating quite reliable variable selection. Importantly, much less discrepancy is observed between DFP and batch MCMC, with them identifying 17 and 21 variables as influential respectively, with 14 identified by both.

6. Conclusion and Future Work

The emergence of large volumes of high-dimensional data mandates that model fitting tools evolve quickly to keep pace with the rapidly growing dimension and size of data. The DFP algorithm proposed in this article dynamically partitions the parameter space after observing every data shard and employs fast and approximate Bayesian inference at each partition in parallel. The detailed simulation studies of DFP with popular Bayesian shrinkage priors (Bayesian Lasso, Horseshoe and Spike-and-Lasso) show indistinguishable inference from batch MCMC with a considerable reduction of per batch computation time. The supplementary material contains the proof of convergence of the DFP algorithm for high-dimensional linear regression as time $t \to \infty$.

The scope of DFP extends well beyond the realm of high-dimensional linear regression with Gaussian errors. For example, as part of our future work, we will employ DFP for high-dimensional logistic and probit regressions. While data augmentation schemes (Albert and Chib 1993; Polson, Scott, and Windle 2013) in high-dimensional binary regression allow Gibbs sampling for parameter blocks, making the DFP formulation natural, they also violate assumptions (1) and (2) in the formulation of DFP in Section 3 which we seek to account for. We also propose to extend the DFP formulation for high-dimensional linear regression with heavy tailed error distributions. Notably, a heavy tailed error distribution can often be expressed as a scale mixture of Gaussian errors. Thus, upon using a data augmentation scheme, developing DFP under this model will require extending the DFP framework when the number of parameters increases with the onset of a new data shard. We would also like to extend our theoretical results on the convergence of the DFP kernel to the full posterior from a fixed partitioning set up to an adaptive dynamic partitioning set up.

Finally, this article constructs $\widehat{\boldsymbol{\Theta}}^{(t)}$ as the average of samples of $\boldsymbol{\Theta}$ drawn from the DFP algorithm at time t. It is be mentioned that the theory allows alternative constructions of $\widehat{\boldsymbol{\Theta}}^{(t)}$, as long as the sequence converges to the true data-generating parameter

as $t \to \infty$. As a future exploration, we plan to develop a hybrid DFP algorithm where $\widehat{\boldsymbol{\Theta}}^{(t)}$ is constructed separately by implementing a frequentist high-dimensional regression technique (e.g., lasso) at the onset of a new data shard at every time. This will guarantee consistency of $\widehat{\boldsymbol{\Theta}}^{(t)}$ and the purpose of fitting the DFP algorithm then becomes quantifying uncertainty in the posterior distribution of parameters. Some of these constitute our current area of research.

Funding

The research of Rajarshi Guhaniyogi is partially supported by grants from the Office of Naval Research (BAA N000141812741) and the National Science Foundation (DMS-1854662).

Supplementary Material

Supplementary Material 1: This document contains proof of the convergence behavior for the DFP algorithm. It also contains details of the DFP algorithm when applied to the linear high-dimensional regression with Bayesian Lasso prior, Horseshoe prior and Spike & Lasso prior on coefficients.

TestScript.R: The package to implement DFP is available at *https://github.com/Rene-Gutierrez/DynParRegReg*. We also upload `TestScript.R` file which uses functions from the package to run the simulations.

References

Albert, J. H., and Chib, S. (1993), "Bayesian Analysis of Binary and Polychotomous Response Data," *Journal of the American Statistical Association*, 88, 669–679. [239]

Armagan, A., Dunson, D. B., and Lee, J. (2013), "Generalized Double Pareto Shrinkage," *Statistica Sinica*, 23, 119. [224,235]

Armagan, A., Dunson, D. B., Lee, J., Bajwa, W. U., and Strawn, N. (2013), "Posterior Consistency in Linear Models Under Shrinkage Priors," *Biometrika*, 100, 1011–1018. [227]

Beskos, A., Crisan, D. O., Jasra, A., and Whiteley, N. (2014), "Error Bounds and Normalising Constants for Sequential Monte Carlo Samplers in High Dimensions," *Advances in Applied Probability*, 46, 279–306. [225]

Betancourt, M. (2018), "A Conceptual Introduction to Hamiltonian Monte Carlo," arxiv.org/pdf/1701.02434.pdf. [225]

Campbell, T., Straub, J., Fisher III, J. W., and How, J. P. (2015), "Streaming, Distributed Variational Inference for Bayesian Nonparametrics," in *Advances in Neural Information Processing Systems*, eds. C. Cortes, N. Lawrence, D. Lee, M. Sugiyama and R. Garnett, Montreal, Canada: NeurIPS Proceedings, pp. 280–288. [225]

Carvalho, C. M., Lopes, H. F., Polson, N. G., and Taddy, M. A. (2010), "Particle Learning for General Mixtures," *Bayesian Analysis*, 5, 709–740. [225]

Carvalho, C. M., Polson, N. G., and Scott, J. G. (2010), "The Horseshoe Estimator for Sparse Signals," *Biometrika*, 97, 465–480. [224,227,233]

Castillo, I., Schmidt-Hieber, J., and Van der Vaart, A. (2015), "Bayesian Linear Regression With Sparse Priors," *The Annals of Statistics*, 43, 1986–2018. [224,227]

Chopin, N. (2004), "Central Limit Theorem for Sequential Monte Carlo Methods and Its Application to Bayesian Inference," *The Annals of Statistics*, 32, 2385–2411. [225]

Christidis, A.-A., Lakshmanan, L., Smucler, E., and Zamar, R. (2020), "Split Regularized Regression," *Technometrics*, 62, 330–338. [226]

Doucet, A., De Freitas, N., and Gordon, N. (2001), "An Introduction to Sequential Monte Carlo Methods," in *Sequential Monte Carlo Methods in Practice*, eds. A. Doucet, N. De Freitas, and N. Gordon, New York: Springer, Springer, pp. 3–14. [225]

George, E. I., and McCulloch, R. E. (1997), "Approaches for Bayesian Variable Selection," *Statistica Sinica*, 7, 339–373. [224,227]

Gneiting, T., and Raftery, A. E. (2007), "Strictly Proper Scoring Rules, Prediction, and Estimation," *Journal of the American Statistical Association*, 102, 359–378. [231]

Griffin, J. E., and Brown, P. J. (2010), "Inference With Normal-Gamma Prior Distributions in Regression Problems," *Bayesian Analysis*, 5, 171–188. [235]

Guhaniyogi, R., Qamar, S., and Dunson, D. B. (2013), "Bayesian Conditional Density Filtering," *Journal of Computational and Graphical Statistics*, 27, 657–672. . [230,238]

Gunawan, D., Dang, K.-D., Quiroz, M., Kohn, R., and Tran, M.-N. (2018), "Subsampling Sequential Monte Carlo for Static Bayesian Models," *arxiv.org/pdf/1805.03317.pdf*. [225,230]

Hager, W. W. (1989), "Updating the Inverse of a Matrix," *SIAM Review*, 31, 221–239. [224]

Hoffman, M., Bach, F. R., and Blei, D. M. (2010), "Online Learning for Latent Dirichlet Allocation," in *Advances in Neural Information Processing Systems*, eds. J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel and A. Culotta, Vancouver, Canada: NeurIPS Proceedings, pp. 856–864. [225]

Hubert, L., and Arabie, P. (1985), "Comparing Partitions," *Journal of Classification*, 2, 193–218. [231]

Lindsten, F., Johansen, A. M., Naesseth, C. A., Kirkpatrick, B., Schön, T. B., Aston, J., and Bouchard-Côté, A. (2017), "Divide-and-Conquer With Sequential Monte Carlo," *Journal of Computational and Graphical Statistics*, 26, 445–458. [225]

Lopes, H. F., and Tsay, R. S. (2011), "Particle Filters and Bayesian Inference in Financial Econometrics," *Journal of Forecasting*, 30, 168–209. [225]

Moral, P. D., Jasra, A., and Y. Zhou (2017), "Biased Online Parameter Inference for State-Space Models," *Methodology and Computing in Applied Probability*, 19, 727–749. [225]

Park, T., and Casella, G. (2008), "The Bayesian Lasso," *Journal of the American Statistical Association*, 103, 681–686. [231]

Polson, N. G., and Scott, J. G. (2010), "Shrink Globally, Act Locally: Sparse Bayesian Regularization and Prediction," *Bayesian Statistics*, 9, 501–538. [224,226]

Polson, N. G., Scott, J. G., and Windle, J. (2013), "Bayesian Inference for Logistic Models Using Pólya–Gamma Latent Variables," *Journal of the American Statistical Association*, 108, 1339–1349. [239]

Rebeschini, P., and Handel, R. v. (2015), "Can Local Particle Filters Beat the Curse of Dimensionality?" *Annals of Applied Probability*, 25, 2809–2866. [225]

Ročková, V., and George, E. I. (2018), "The Spike-and-Slab Lasso," *Journal of the American Statistical Association*, 113, 431–444. [227]

Rue, H. (2001), "Fast Sampling of Gaussian Markov Random Fields," *Journal of the Royal Statistical Society*, Series B, 63, 325–338. [224]

Scott, J. G., and Berger, J. O. (2010), "Bayes and Empirical-Bayes Multiplicity Adjustment in the Variable-Selection Problem," *The Annals of Statistics*, 38, 2587–2619. [224,227]

Snyder, C., Bengtsson, T., Bickel, P., and Anderson, J. (2008), "Obstacles to High-Dimensional Particle Filtering," *Monthly Weather Review*, 136, 4629–4640. [225]

Szabó, B., van der Vaart, A., van Zanten, J. (2015), "Frequentist Coverage of Adaptive Nonparametric Bayesian Credible Sets," *The Annals of Statistics*, 43, 1391–1428. [234]

Van de Geer, S., Bühlmann, P., Ritov, Y., and Dezeure, R. (2014), "On Asymptotically Optimal Confidence Regions and Tests for High-Dimensional Models," *The Annals of Statistics*, 42, 1166–1202. [234]

Wigren, A., Murray, L., and Lindsten, F. (2018), "Improving the Particle Filter in High Dimensions Using Conjugate Artificial Process Noise," *IFAC-PapersOnLine*, 51, 670–675. [225]

Zhang, C.-H., and Zhang, S. S. (2014), "Confidence Intervals for Low Dimensional Parameters in High Dimensional Linear Models," *Journal of the Royal Statistical Society*, 76, 217–242. [234]