Topics in Modern Bayesian Computation

by

Shaan Qamar

Department of Statistical Science
Duke University

Date: __________________

Approved:

___________________________
David B. Dunson, Supervisor

___________________________
Surya T. Tokdar

___________________________
Merlise A. Clyde

___________________________
Henry D. Pfister

Dissertation submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in the Department of Statistical Science
in the Graduate School of Duke University

2015
Abstract

Topics in Modern Bayesian Computation

by

Shaan Qamar

Department of Statistical Science
Duke University

Date: ______________________

Approved:

__________________________
David B. Dunson, Supervisor

__________________________
Surya T. Tokdar

__________________________
Merlise A. Clyde

__________________________
Henry D. Pfister

An abstract of a dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Department of Statistical Science in the Graduate School of Duke University 2015
Abstract

Collections of large volumes of rich and complex data has become ubiquitous in recent years, posing new challenges in methodological and theoretical statistics alike. Today, statisticians are tasked with developing flexible methods capable of adapting to the degree of complexity and noise in increasingly rich data gathered across a variety of disciplines and settings. This has spurred the need for novel multivariate regression techniques that can efficiently capture a wide range of naturally occurring predictor-response relations, identify important predictors and their interactions and do so even when the number of predictors is large but the sample size remains limited.

Meanwhile, efficient model fitting tools must evolve quickly to keep pace with the rapidly growing dimension and complexity of data they are applied to. Aided by the tremendous success of modern computing, Bayesian methods have gained tremendous popularity in recent years. These methods provide a natural probabilistic characterization of uncertainty in the parameters and in predictions. In addition, they provide a practical way of encoding model structure that can lead to large gains in statistical estimation and more interpretable results. However, this flexibility is often hindered in applications to modern data which are increasingly high dimensional, both in the number of observations $n$ and the number of predictors $p$. Here, computational complexity and the curse of dimensionality typically render posterior computation inefficient. In particular, Markov chain Monte Carlo (MCMC) methods which remain the workhorse for Bayesian computation (owing to their generality and
asymptotic accuracy guarantee), typically suffer data processing and computational bottlenecks as a consequence of (i) the need to hold the entire dataset (or available sufficient statistics) in memory at once; and (ii) having to evaluate of the (often expensive to compute) data likelihood at each sampling iteration.

This thesis divides into two parts. The first part concerns itself with developing efficient MCMC methods for posterior computation in the high dimensional $large-n$ $large-p$ setting. In particular, we develop an efficient and widely applicable approximate inference algorithm that extends MCMC to the online data setting, and separately propose a novel stochastic search sampling scheme for variable selection in high dimensional predictor settings. The second part of this thesis develops novel methods for structured sparsity in the high-dimensional $large-p$ $small-n$ regression setting. Here, statistical methods should scale well with the predictor dimension and be able to efficiently identify low dimensional structure so as to facilitate optimal statistical estimation in the presence of limited data. Importantly, these methods must be flexible to accommodate potentially complex relationships between the response and its associated explanatory variables. The first work proposes a nonparametric additive Gaussian process model to learn predictor-response relations that may be highly nonlinear and include numerous lower order interaction effects, possibly in different parts of the predictor space. A second work proposes a novel class of Bayesian shrinkage priors for multivariate regression with a tensor valued predictor. Dimension reduction is achieved using a low-rank additive decomposition for the latter, enabling a highly flexible and rich structure within which excellent cell-estimation and region selection may be obtained through state-of-the-art shrinkage methods. In addition, the methods developed in these works come with strong theoretical guarantees.
To my loving parents and sisters.
Contents

Abstract iv
List of Tables xii
List of Figures xv
Acknowledgements xviii
1 Introduction 1
  1.1 Motivation .......................................................... 1
  1.2 Research questions and contributions ............................. 2
2 Bayesian Conditional density filtering 10
  2.1 Introduction .......................................................... 10
    2.1.1 A new class of approximate MCMC methods .................. 13
  2.2 Sufficiency and surrogate conditional statistics .................. 14
  2.3 The C-DF algorithm .................................................. 16
  2.4 Motivating examples .................................................. 17
    2.4.1 Linear regression ................................................. 18
    2.4.2 One-way Anova model ............................................ 19
  2.5 Advanced data models ............................................... 22
    2.5.1 Dynamic linear model ........................................... 24
    2.5.2 Binary regression model ....................................... 27
    2.5.3 Poisson mixed effects model ................................... 31
2.6 Theory guarantee for the C-DF algorithm .................................. 34
  2.6.1 Notation and framework ................................................. 34
  2.6.2 Finite sample error bound for approximate kernels ............... 35
  2.6.3 Convergence for a general approximation class ...................... 35
2.7 C-DF for online compressed regression ................................. 38
  2.7.1 Online inference and competitors ...................................... 39
  2.7.2 Simulation experiments .................................................. 41
2.8 Extensions and future work .................................................. 43

3 Stochastic neighborhood search for Bayesian variable selection 45
  3.1 Introduction ................................................................. 46
  3.2 Basic notation and background .......................................... 48
  3.3 Scaling MCMC variable selection ........................................ 49
    3.3.1 Weighted toggle proposals for predictor inclusion .............. 49
    3.3.2 The Swesenden-Wang prior ......................................... 51
  3.4 A framework for robust SSVS in high dimensions ..................... 52
    3.4.1 Sparse priors for inclusion vectors .............................. 52
    3.4.2 Reversible neighborhood sampling ............................... 53
    3.4.3 Paired-moved neighborhood search ............................... 55
    3.4.4 A multiple-try Metropolis algorithm for inclusion ........... 56
    3.4.5 A paired-move discrete MTM sampler ......................... 58
    3.4.6 Adaptive predictor importance .................................. 60
  3.5 Examples of variable selection for GLMM ............................ 61
    3.5.1 Variable selection in conjugate settings ...................... 61
    3.5.2 Variable selection in a non-conjugate setting ................ 62
  3.6 Analysis of pMTM .......................................................... 64
## 3.6.1 Real data comparisons

3.6.2 Simulation comparisons

## 3.7 Extensions and future work

3.7.1 Improved adaptivity

3.7.2 Speeding up pMTM via subsampling

### 4 Additive Gaussian process regression

4.1 Introduction

4.2 GP regression: a brief overview

4.2.1 Gaussian process regression: notation and setup

4.2.2 Existing theory guarantee for GP regression

4.2.3 A simple illustrating example

4.3 Variable selection and structured GP regression

4.4 Additive GP regression

4.4.1 An Additive-interactive framework for structured sparsity

4.4.2 A model for additive GP regression

4.4.3 Prior specification for aGP

4.4.4 Stochastic component adaptation

4.5 Posterior computation for aGP regression

4.5.1 Ensemble-wide moves

4.5.2 Efficient low-rank model fitting

4.5.3 Predictive inference

4.6 Simulation and real data analysis

4.6.1 Simulated experiments

4.6.2 Predictive performance

4.6.3 Real data illustrations
5 Bayesian Tensor regression

5.1 Introduction ................................................................. 110

5.2 Tensor regression ......................................................... 112

5.2.1 Basic notation ............................................................. 112

5.2.2 Model framework ......................................................... 113

5.2.3 Model identifiability ..................................................... 115

5.3 Bayesian Tensor regression ............................................. 116

5.3.1 Shrinkage priors .......................................................... 116

5.3.2 Criteria for a multiway prior .......................................... 118

5.4 Multiway shrinkage priors .............................................. 119

5.4.1 The multiway MGP prior .............................................. 119

5.4.2 The multiway Dirichlet GDP prior .................................. 121

5.4.3 Prior hyper-parameter elicitation .................................... 122

5.4.4 Prior calibration and initialization .................................. 125

5.5 Posterior computation and model fitting ............................ 126

5.5.1 Model fitting preliminaries .......................................... 127

5.5.2 M-DGDP Gibbs sampler .............................................. 127

5.6 Simulation studies .......................................................... 129

5.7 Posterior consistency for Tensor regression ...................... 134

5.7.1 Notation and framework .............................................. 135

5.7.2 Main result ................................................................. 135

5.8 Extensions and future work ............................................. 137

A Bayesian Conditional density filtering (C-DF) .................. 139

A.1 Additional tables and figures .......................................... 139
List of Tables

2.1 Inferential performance for C-DF and SMCMC for parameters of interest. Coverage and length are based on 95% credible intervals and is averaged over all the $\beta_j$'s ($j = 1, \ldots, 5$) and all time points and over 10 independent replications. We report the time taken to produce 500 MCMC samples with the arrival of each data shard. MSE along with associated standard errors are reported at different time points.  

2.2 Inferential performance for C-DF, SMCMC, and ADF for parameter $\zeta$. Coverage is based on 95% credible intervals averaged over all time points, all $\zeta$ and over 10 independent replications. We report the time taken to produce 500 MCMC samples with the arrival of each data shard. MSE along with associated standard errors are reported at different time points.  

2.3 Inferential performance for C-DF and Particle Learning. Coverage and length are based on 95% credible intervals for $\theta_t$ averaged over all time points and 10 independent replications. For truth $\theta_{t0}$ at time $t$, we report $\text{MSE} = \frac{1}{T_n} \sum_{t=1}^{T_n} (\hat{\theta}_t - \theta_{t0})^2$. We report the time taken to run C-DF with 50 Gibbs samples at each time for $\tau^2, \theta, \sigma^2$ and 500 MH samples for $\phi$.  

2.4 Computational and storage requirements for the Dynamic Linear Model using C-DF and PL. $C_{i,j}^t$ is the $i$-th CSS corresponding to the $j$-th particle in PL, $i = 1:4$, $j = 1:N$, $N = 100$ is the number of particles propagated by PL, and $G = 500$ is the number of Metropolis samples used by both PL and C-DF. Memory in terms of RAM used to store and propagate SCSS and CSS for C-DF and PL is reported. Sampling and update complexities are in terms of big-O.
2.5 Inferential performance for C-DF and SMCMC in simulation studies 
\((p, b, n, t) = (100, 500, 25, 100)\) and 
\((p, b, n, t) = (500, 3500, 100, 100)\). Coverage and length are based on 95% credible intervals averaged over all predictors and over 10 independent replications. MSE is reported over all predictors, while 
\(\text{MSE}_{10} = \frac{1}{10} \sum_{j=1}^{10} (\hat{\beta}_j - \beta_0)^2\). We report the total time taken to produce 500 MCMC samples after the arrival of each data shard. 

2.6 Computational and storage requirements for the latent variable probit model using C-DF and SMCMC. Budget \(b\) represents the number of latent scores updated by C-DF when processing data shard at time \(t\). Runtime is quickly dominated by the sampling complexity which scales linearly in time for the augmented Gibbs sampler (SMCMC). Memory is reported for case 2, 
\((p, b, n, t) = (500, 3500, 100, 100)\). Sampling and update complexities are in terms of big-O. 

2.7 Simulation experiments for supervised compressed regression. “High” signal corresponds to \(\beta_j \sim U(-3, 3)\), while “low” signal corresponds to \(\beta_j = 0.10\) for every nonzero feature. Sparse cases are denoted with (*). 

2.8 MSPE comparisons for each simulated experiment in Table 2.7. Subscripts denote bootstrapped standard errors calculated using independent replications. 

2.9 Performance comparison in terms of relative parameter MSE at 
\(t = 100, 200\). Relative MSE for \(\gamma = \Phi' \beta\) is computed as 
\(|\|\Phi' \beta - \gamma_0\|/\|\gamma_0\|\|^2\) for C-DF and SMCMC, \(\gamma_0\) is \(\gamma\) at the truth. For VB, we compute 
\(|\|\mu_\beta^{VB} - \gamma_0\|/\|\gamma_0\|\|^2\), where \(\mu_\beta^{VB}\) is the approximate posterior mean for \(\beta\). Subscripts denote bootstrapped standard errors calculated using independent replications. 

3.1 Descriptions of four datasets from the UCI repository. ‘CCS’: concrete compressive strength; ‘OC’: ozone concentration; ‘medv’: median value. Two-way interactions between original predictors were also included for each dataset (except ‘cookie’). For ‘ozone’, squared predictors were also included. 

3.2 Summary performance statistics. All methods run for approximately an equal amount of CPU time. Median values of summary statistics and standard-errors reported are obtained across 10 cross validated splits. \(sd(p_{med})\) is reported as an average over all \(p\) predictors. 

3.3 Simulated experiments
3.4 Simulation 1, ‘boston’; high correlation; SNR (1,1) 68
3.5 Simulation 2, ‘crime’; high correlation; SNR (1,1) 68
3.6 Simulation 3, ‘crime’; high correlation; SNR (1,2) 68
3.7 Simulation 4, ‘crime’; low correlation; SNR (1,1) 68

4.1 Simulated test functions. All experiments fix \( n = 100 \) and \( p = 1000 \). 98
4.2 Predictive RMSE for test functions defined in Section 4.6.1 with \( n = 100 \) and \( p = 1000 \). Standard errors over 10 replicates appear as subscripts on averaged RMSE values. 103
4.3 Predictive RMSE for test functions defined in Section 4.6.1 with \( n = 100 \) and \( p = 50 \). Standard errors over 10 replicates appear as subscripts on averaged RMSE values. 104

4.4 Performance of aGP against the same by BART, RF, Lasso, and the mean prediction rule (NULL) for well known real datasets. Left: the fraction of explained variance computed as \( R^2 = 1 - (\text{RMSE}/\bar{s}_y)^2 \) for aGP; Right: predictive (hold-out) RMSE averaged over random test/train splits with standard errors appearing as subscripts. 106

5.1 Percentiles for \(|B_{i_1,\ldots,i_D}|\) under the M-DGDP prior with default \( a_\lambda = 3 \), \( b_\lambda = \sqrt{a_\lambda} \), \( b_r = \alpha R^{1/D} \) and \( \alpha = 1/R \). Statistics are displayed as the dimension \( D \) of the tensor and its parafac rank decomposition \( R \) vary. 126

5.2 Comparison of voxel estimation as measured by root mean squared error (RMSE) for the six 2D tensor images portrayed in Figure 5.4. Results from “Zhou” (Zhou et al., 2013) use \( R = 5 \). By default, \( R = 10 \) is used in all M-DGDP runs. 132

5.3 M-DGDP coverage statistics on generated and ready-made 2D tensor images with simulated tensor predictor data. 133

5.4 Sensitivity analysis of voxel estimation error (RMSE) as the tensor dimension increases; here \( p_j = p \in \{64,100\} \) for the 2D tensor images ‘R5-ex’ and ‘Shapes’. 133

A.1 Predictive coverage for each simulation experiment in Table 2.7. Empirical 95% confidence intervals of the coverage probabilities over independent replications are also reported. 140

A.2 Effective sample sizes (number of MCMC samples) required until MSPE \( \leq 5 \) are shown above for each simulation experiment, with results averaged over 10 independent replications. 140
## List of Figures

2.1 Kernel density estimates for posterior draws using SMCMC and the C-DF algorithm at $t = 200,500$. Shown from left to right are plots of model parameters $\beta_1$, $\beta_4$, and $\sigma^2$, respectively. ........................................ 20

2.2 Row #1 (left to right): Kernel density estimates for posterior draws of $\zeta_1$, $\zeta_5$, $\zeta_{10}$ using SMCMC and the C-DF algorithm at $t = 200,500$; Row #2 (left to right): Kernel density estimates for model parameters $\tau^2$, and $\sigma^2$ at $t = 500$. ........................................ 23

2.3 Row #1 (left to right): Kernel density estimates for posterior draws of $\theta_t$ using PL and the C-DF algorithm at $t = 1000,2000,3000$; Row #2 (left to right) plots of model parameters $\tau^2$ and $\phi$, respectively. . . . . . . . . 28

2.4 Kernel density estimates for posterior draws using SMCMC and the C-DF algorithm at $t = 50,100$. Shown from left to right are plots of model parameters $\beta_1$, $\beta_5$, and $\beta_{10}$ (and top to bottom are case 1 and case 2), respectively. ........................................ 31

3.1 Left: prior for predictor inclusion probability $\tau$ with default hyperparameters; Middle: prior probability over component inclusion size for varying predictor dimensions $p$ and fixed inclusion probability $\tau = 1/2$; Right: for $p = 100$, the latter is compared to $\pi(\gamma|d^* = 1)$, the induced marginal prior over component inclusion size. ................................. 54

3.2 Marginal inclusion probabilities and log-posterior trace plots shown averaged over 10 cross validated replications for datasets considered in Table 3.1. ........................................ 72

3.3 Simulated data with setup given in Table 3.3. ........................................ 73
4.1 Plots of two 1D functions for various fixed choices of length-scale correlation coefficient $c(\Delta)$ corresponding to $\lambda = \sqrt{-\log c/\Delta^2}$. Row 1: plots $f(x)$ and Row 2: plots $h(x)$ given in Section 4.4.3. In red: pointwise predictive means on an equi-spaced grid on $[0,1]$, along with associated 95% credible bands for GP regression (GPR). In purple: same for BART; In black: true function $\mu(x) = \{f, h\}$. Observed response data appear as ‘x’.

4.2 Simulated test functions 1-4 for Section 4.6.1 with $p = 1000$. Left: number of active vs. non-empty (utilized) components, with median sizes appearing as vertical lines; Middle: marginal inclusion probabilities for the aGP model (index on log-scale); Right: an interaction graph among the most important predictors. Vertices (edges between predictor pairs) are drawn proportional to predictor importance (co-appearance across components).

4.3 A plot of the cumulative variance explained as a function of the median number of active components (sorted by increasing importance). The fraction of the marginal variance explained by an active GP component is $\rho^2_h/(1 + \sum_{h \in \mathcal{A}} \rho^2_h)$, $h \in \mathcal{A}$. Median active sizes appear as vertical lines in histograms shown in Figure 4.6.1.

4.4 Hold-out predictive performance for the modified Friedman function in Section 4.6.2 for experiments with $p = 50$. True response values are plotted along the x-axis and predicted values along the y-axis, with point-wise 95% credible intervals overlaid. Here, Bayesian Lasso is used in lieu of the Lasso to enable reporting of credible bands.

4.5 A plot of the cumulative variance explained as a function of the median number of active components (sorted by increasing importance). The fraction of the marginal variance explained by an active GP component is $\rho^2_h/(1 + \sum_{h \in \mathcal{A}} \rho^2_h)$, $h \in \mathcal{A}$. Median active sizes appear as vertical lines in histograms shown in Figure 4.6.

4.6 Comparison across well known datasets. Left: number of active vs. non-empty (utilized) components, with median sizes appearing as vertical lines; Middle: marginal inclusion probabilities for the aGP model (index on log-scale); Right: an interaction graph among the most important predictors. Vertices (edges between predictor pairs) are drawn proportional to predictor importance (co-appearance across components).

5.1 Visualization of points in the $S^2$ probability simplex for 500 independent realizations of $X \sim \text{Dirichlet}(\alpha)$. 

xvi
5.2 Induced voxel level prior distribution for default specification as a function of $a_\lambda$, with $b_\lambda = \frac{z_0}{\sqrt{r_\lambda}}$, $b_\tau = \alpha R^{1/D}$ and $\alpha = 1/R$.

5.3 Induced voxel level prior distribution for default specification for $a_\lambda \in \{2, 3, 5\}$, $b_\lambda \in \{1, 3\}$, with $b_\tau = \alpha R^{1/D}$ and $\alpha = 1/R$. Here, $R = 10$ and $D = 2$; note that $\frac{z_0}{\sqrt{r_\lambda}} \in (1.18, 1.50)$ for the range of $a_\lambda$ considered.

5.4 Simulated data with $64 \times 64$ 2D tensor images ($p = 64, D = 2$). Row 1: The first two images (from left) have a rank-3 and rank-5 parafac decomposition; the third image is “regular”, although does not have a low-rank parafac decomposition. Row 2: All three images are irregular, and do not have a low-rank parafac decomposition. Sparisty (% non-zero voxels) are displayed in sub-captions.

5.5 Recovered images for the $64 \times 64$ 2D tensor images in Figure 5.4. By default, $R = 10$ is used in all M-DGDP runs.

A.1 Parameter accuracy plotted over time as defined in (2.2) for the motivating examples of Section 2.4. Image 1: accuracy for representative regression coefficients $\beta_j$ and $\sigma^2$ in the linear regression example. Images 2 and 3: accuracy for representative group means $\zeta_j$, along with hierarchical parameters $\mu$, $\tau^2$ and $\sigma^2$ for the one-way Anova model.

C.1 Marginal predictor importance (inclusion probability) for BART, RF, and Lasso on each simulated test function considered in Section 4.6.1. Competitors use the same data set ($n = 100, p = 1000$) that generated Figure 4.6.1.

C.2 Posterior plots and comparisons for the real datasets considered in Section 4.6.3. Left: plot of predictor importance scores, $v_j$, $j = 1, \ldots, p$. Middle: bar-plots of GP scale parameters $\rho, \lambda$ for contributing components (i.e., having 75% quantile over posterior draws for $\rho_l > 0$). Right: trace plot of posterior draws for variance parameter, $\sigma^2$.
Acknowledgements

Sincere gratitude and thanks goes to my advisor, David Dunson, for being immensely generous and supportive throughout my doctoral studies. David’s enthusiasm and passion towards research has been a driving inspiration in my own work. He is a phenomenal mentor and provided me with much freedom to seek out, pursue and develop research ideas that I found exciting. Along the way, he has always provided great insights and feedback. David has a tremendous ability to synthesize research concepts and ideas, while also being able to see the potential of an idea and ways in which something limited might be generalized. I have learned a great deal from him and his ability to do both simultaneously.

I would also like to immensely thank my secondary advisor, Surya Tokdar, for his constant dedication and support to my research and to my development as a statistician. He has provided wonderful guidance and constant encouragement over the years. I have learned a great deal from his mentorship and direction, and have thoroughly enjoyed working with him. Surya taught me much in the way of how to approach and think deeply about problems, and has been an inspiration in the development of my interest in methodological and computational Bayesian statistics.

I would like to thank all my thesis committee members, David Dunson, Surya Tokdar, Merlise Clyde, and Henry Pfister for their time and support. They have provided valuable ideas, critiques, and guidance which I value immensely. I would also like to thank Alan Gelfand, Jim Berger and Mike West for their time and
support over the years—I very much enjoyed our chats. They have poured great
deal compassion into molding the department into what it has become today. The
Department of Statistical Science is certainly a one-of-a-kind research environment!

A big thanks goes to my colleague and friend, Rajarshi Guhaniyogi, with whom
I have had the opportunity and pleasure of working with over the past two years. I
have thoroughly enjoyed our collaboration and discussions, and have indeed learned
a great may things from it. Thanks also goes to my good friend Sanvesh Srivastava
for our enjoyable semi-regular Friday evening trips to dinner in Cary and Chapel
Hill, and for the many fun discussions we’ve shared.

To my fellow Ph.D. colleagues - Shi-han, Doug, Jacopo, James, Zoey, Michael
and Yun - thank you for all the enjoyable times. The past four years has been an
incredible journey, and it wouldn’t have been the same without you.

I’d also like to thank dear friends external to Duke whose friendship and support
I value and cherish; they are my extended family. Thank you Wasif Syed, Gokhan
Arikan, Saad Ahsan¹, Aziz Al Majid, Haider Syed, Hatice Bilici, Khalid Al-Kaabi,
and Maan Aldaiel. There are numerous other individuals who I am indebted to but
it would not be possible to name everyone.

Finally, I am infinitely indebted to my loving family - my parents and siblings -
without their eternal support and encouragement, I would not be where I am today.

This research was partially supported by grants R01-ES-017436 and R01-ES-
017240 from the National Institute of Environmental Health Sciences (NIEHS) of
the National Institutes of Health (NIH).

¹ WS, GA, SQ, SA are sole members of the “Cornell Dream team”
1

Introduction

1.1 Motivation

Bayesian methods provide a natural probabilistic characterization of uncertainty in the parameters and in predictions, in addition to practical and interpretable ways of encoding model structure that lead to efficient and robust estimation procedures. However, this flexibility is often hindered in applications to data which are increasingly high-dimensional, both in the number of observations $n$ and the number of measured predictors $p$. Examples of this include epigenetics, where several thousands of biomarkers are often measured in dose-response studies; genomics, where testing of associations between a very large number of genetic variants (single nucleotide polymorphisms) and quantitative traits is common; finance, with high frequency measurements of market instruments that comprise modern time-series data; and so on. In such settings, statistical methods increasingly employ additional modeling assumptions (e.g., low-rank, smoothness, sparsity, additivity, etc.) to combat the curse of dimensionality and to aid estimation efficiency and robustness. In high-dimensional regression, for example, the response is assumed to depend importantly on few predictors, with the majority being irrelevant or approximately so; in den-
density estimation, a low-rank assumption for the covariance of a multivariate response is often assumed; and in manifold learning, observed high-dimensional predictors are presumed to concentrate on (or near) a lower dimensional subspace. A unifying theme across such forms of “dimension reduction” is the incorporation of prior knowledge, structure, and modeling constraints as part of the statistical method, and that this is crucial to improving the estimation efficiency relative to the sample size. In fact, many Bayesian procedures with carefully constructed priors can adapt to such low-dimensional structure in an automatic, often optimal, manner (Stone, 1982; van der Vaart and van Zanten, 2008; Castillo et al., 2012; Tokdar, 2012; Shen et al., 2013).

Markov chain Monte Carlo (MCMC) methods for Bayesian computation remain at large, owing to their great generality. When the number of observations is truly massive, however, data processing and computational bottlenecks render many MCMC methods infeasible as a consequence of the fact that they demand (i) the entire dataset (or available sufficient statistics) be held in memory; and (ii) likelihood evaluations for updating model parameters at every sampling iteration which can often be costly. Various methods have been developed to approximate the joint posterior to alleviate one or both these shortcomings (e.g., Laplace approximation, assumed density filtering, variational methods etc.), but tractability has generally come at the expense of being able to obtain valid inference on model parameters of interest. Fortunately, recent years has seen a surge of methodological research attempting to rescue MCMC from intractability via data subsampling, subsetting, resampling, and density filtering approaches.

1.2 Research questions and contributions

The primary focus of this thesis is two-fold, dividing essentially along the lines of the well-known large-$p$ small-$n$ and large-$n$ paradigms. The first part of this the-
sis focuses on methodological and computational solutions for Bayesian inference in settings where standard Markov chain Monte Carlo (MCMC) methods do not scale because of the size or dimension of the data. Chapter 2 concerns itself with the development of a novel and efficient approximate inference algorithm to scale MCMC to the online (streaming) data setting. Dimension reduction is introduced by way of tracking new quantities which we call surrogate conditional sufficient statistics, and thereby approximating the full conditional distributions in time without having to observe, process or store all the data at once. Chapter 3 develops a novel MCMC framework for scaling variable selection in high dimensions which (i) similar to existing two-component mixture prior methods for predictor inclusion, is general and widely applicable; and (ii) synthesizes ideas from stochastic search sampling and ‘multiple-try’ Monte Carlo methods, with key innovations addressing the need for efficient mixing in the high-dimensional space of predictor inclusion to enable reliable estimation of marginal predictor inclusion probabilities.

The second part of this thesis focuses on developing practicable Bayesian methods in the large p small n data setting that tackle various limitations of existing approaches toward multivariate regression. The first work proposes a novel additive Gaussian process model to learn predictor-response relations that may be highly nonlinear and include numerous lower order interaction effects, possibly in different parts of the predictor space. The unknown regression function is theorized to decompose as a sum of a fixed number of component functions depending nonparametrically on a small number of predictors, each modeled using a Gaussian process prior. Careful prior specification is developed in light of performance and computational considerations in exploring the joint posterior over multiple subsets of high-dimensional predictor inclusion vectors in the $p \gg n$ setting. A second work proposes a novel class of Bayesian shrinkage priors for multivariate regression with a tensor valued predictor. Dimension reduction is achieved using a low-rank additive tensor decomposition, en-
abling a highly flexible and rich structure within which excellent cell-estimation and region selection may be obtained using state-of-the-art shrinkage methods. Finally, the developed methods are designed to have strong theoretical support.

The remainder of this section briefly summarizes the central research questions addressed in later chapters.

*Scalable MCMC for online (streaming) data*

Sampling algorithms remain as the workhorse for Bayesian computation, with the overwhelming emphasis being on Markov chain Monte Carlo (MCMC). State of the art performance is obtained in small to moderate sized problems, but MCMC fails to scale as the data and number of parameters grow (often due to the need to update many parameters with expensive likelihood evaluations). In addition, there is a general lack of scalable inference algorithms having guarantees on accuracy. Frequently used variational methods developed in the machine learning literature attempt to address various difficulties facing MCMC methods by making additional approximations to the joint posterior distribution over model parameters. These methods work well in simple low dimensional settings, but fail to capture dependence in the joint posterior, severely underestimate uncertainty in more complex or higher dimensional settings and generally come with no accuracy guarantee.

Recent data subsampling approaches attempt to mitigate the MCMC computational bottleneck by choosing a small set of points at each iteration for which the likelihood is to be evaluated, and doing so in a way that preserves validity of the draws as samples from the desired target posterior (Korattikara et al., 2013; Quiroz et al., 2014; Maclaurin and Adams, 2014). Building on ideas reminiscent of sequential importance resampling and particle filtering (Doucet et al., 2000; Arulampalam et al., 2002), these methods are promising and it remains an active area of research to extend them to more complex model settings. In fact, a separate literature has
recently tackled the problem of scaling Bayesian inference for nonparametric Gaussian process regression in data with large sample sizes (Snelson and Ghahramani, 2006; Banerjee et al., 2008; Lawrence et al., 2009; Banerjee et al., 2012). In addition, deterministic data subsetting approaches have also been proposed, but face the additional challenge of developing valid recombination techniques that recombine “subset posteriors” in a principled fashion (Scott et al., 2013; Minsker et al., 2014).

Chapter 2 proposes a new class of Conditional Density Filtering (C-DF) algorithms for efficient online Bayesian inference. C-DF adapts MCMC sampling to the online setting, sampling from approximations to conditional posterior distributions obtained by propagating functions of data and parameter estimates as new data arrive. This eliminates the need to store or process the entire dataset simultaneously. The proposed C-DF algorithm is general and widely applicable. In addition to delivering state-of-the-art inferential and predictive performance across a variety of model settings, C-DF often results in significant computational efficiencies and storage gains over other competitors. Finally, we establish sufficient conditions under which C-DF samples converge to the target posterior distribution asymptotically as sampling proceeds and more data arrives.

Scaling variable selection in high dimensions

Variable selection in the high dimensional large-\(p\) small-\(n\) paradigm has received a tremendous amount of attention in recent years, with numerous penalized approaches including the Lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001), Cosso (Lin et al., 2006), and the Dantzig selector (Candes and Tao, 2007) enjoying much success. Likewise, the development of Bayesian methods for variable selection has been an extremely productive area of research for over two decades. Although the spike-and-slab approach to Bayesian variable selection has received much attention (George and McCulloch, 1993; Geweke, 1994; Carlin and Chib, 1995; Clyde et al., 1996;
Hoeting et al., 1999) and possesses attractive theoretical properties (Castillo et al.,
2012), intractability of exploring an exponentially large space of predictor inclusion
along with the belief that many regression coefficients may be small rather than
exactly zero has led to considerable growth in the appeal for continuous shrinkage
priors, see Park and Casella (2008); Carvalho et al. (2010b); Armagan et al. (2013);
Bhattacharya et al. (2014) and references therein. Indeed, it has become somewhat
of an art to design continuous shrinkage priors that are computationally tractable
and have desirable concentration properties around sparse neighborhoods.

Chapter 3 introduces a novel Markov chain sampler for posterior inference over
predictor inclusion by synthesizing ideas from shotgun stochastic search (Hans et al.,
2007; Scott and Carvalho, 2008) with the multiple-try Metropolis algorithm (Liu
et al., 2000), suitably adapted to the discrete setting for sampling a \( p \)-dimensional
predictor inclusion vectors \( \gamma = (\gamma_1, \ldots, \gamma_p) \in \{0, 1\}^p \). Key innovations address chal-
 lenges in exploring the high dimensional posterior over predictor inclusion, with
efficient local moves encouraged through an adaptive MCMC scheme. We propose a
new “paired neighborhood” approach to allow for the consideration of distinct types
of proposals separately and in quick succession. This enables easier transitions which
results in improved mixing in the space of single predictor changes to the predictor
inclusion vector. Scalability is addressed by way of incorporating data subsampling
methods as an efficient means of cutting the computational overhead associated with
scoring predictor inclusion neighbors (a collection of inclusion vectors). Applications
to linear and nonparametric regression are presented, and results demonstrate that
our proposed method leads to reliable\(^1\) estimates for marginal predictor inclusion
probabilities over other leading sampling based competitors.

\(^1\) A measure of variability for the estimates is given across repeated simulation studies. Compar-
isons on the basis of log-posterior probability over the MCMC chain, predictive estimation error,
etc. are also provided.
Structured sparsity for high-dimension regression

The second part of this thesis focuses on new Bayesian methods for structured additive regression in the high-dimensional setting. Much of the high-dimensional regression literature focuses on parametric linear models regularized with sparsity and shrinkage (Tibshirani, 1996; Candes and Tao, 2007; Hastie et al., 2009). Association discovery and intervention effect quantification (e.g., in observational studies and survey data) are routinely required in high impact scientific research, and by requiring a parametric relation to hold globally, such models frequently fail to adequately capture many naturally occurring predictor-response relations. In particular, when the response and the associated predictors are nonlinear related and may include low-order interaction effects, simple parametric models remain prone to introducing bias in the estimation of the effect-size for key predictors of interest. In addition, these methods run the risk of over-parametrization when attempting to adjust for non-linearity or to uncover predictor interaction.

Chapter 4 proposes a novel additive Gaussian process (aGP) model for nonparametric regression to learn such a predictor-response relationship. Indeed, several smoothing based nonparametric regression methods exist and accommodate a wider range of predictor-response relations, but they tend to scale poorly in the predictor dimension (O’Hagan and Kingman, 1978; Bertin and Lécué, 2008; Williams and Rasmussen, 1996; Friedman, 1991; Breiman, 2001; Lafferty and Wasserman, 2008). In particular, a traditional sparse nonparametric regression model in which the response is assumed to depend on \( d \) of the original \( p \) predictors, statistical estimation suffers greatly from the curse of dimensionality unless it is known that \( d = o(\log n) = o(\log \log p) \), i.e., the response is extremely sparse in the observed predictors. Under the recently proposed additive-interactive framework for sparse additive regression (Yang and Tokdar, 2014), our proposed aGP method offers state-of-the-art support and interaction recovery. aGPs versatility stems from (i) its ad-
ditive nonparametric structure; (ii) carefully constructed prior specification which allow the method to adapt to varying patterns of sparsity and interaction in the data; and (iii) a robust and scalable sampling scheme for posterior inference in the \( p \gg n \) setting. Results from real data studies provide strong evidence that the additive-interactive framework is indeed an attractive modeling platform for high-dimensional nonparametric regression.

In chapter 5 we develop a Bayesian regression method for data with tensor valued predictors. The method is developed in the linear regression setting, with extensions to the generalized linear model setting being straightforward (Hastie et al., 1986). For a \( D \)-way tensor having margins dimensions \( p_1, p_2, \ldots, p_D \), the number of parameters to estimate in the saturated (full) model is \( p_1 \times p_2 \times \cdots \times p_D \). Naively vectorizing the tensor predictor thus leads to a very high-dimensional vector that fails to exploit the structure of the tensor, often resulting in poor estimation and predictive performance. Moreover, vectorization fails to preserve spatial structure in the multidimensional array which carries crucial information that ought to be incorporated in the estimation procedure so as to enable efficient learning. To obtain an adequate dimension reduction, we impose a low-rank assumption on the tensor parameter, supposing it to decompose as a PARAFAC sum of order \( R \), thus reducing the number of free parameters to \( R \sum_{j=1}^{D} p_j \). Within this framework, we develop a novel class of prior distributions on tensor structured predictors that automatically regulates model complexity by adaptively collapsing on a lower dimensional structure in a data dependent way. In particular, the proposed class of multiway shrinkage priors are carefully designed to adaptively collapse on an appropriate rank, induce heavy-tailed shrinkage priors on the cell-specific coefficients, borrow information exploiting the tensor structure to accommodate applications with enormous numbers of cells in the tensor and to facilitate efficient posterior computation. Of course, one naturally wonders whether such a dramatic dimension reduction retains sufficient
flexibility so as to recover accurate estimates of the true tensor parameter. This is understood by verifying if shapes corresponding to regions of interest for a tensor parameter are reconstructed well under our proposed model. The answer from extensive simulation studies and theoretical analysis is in the affirmative. In particular, the proposed prior is shown to be posterior consistent, and simulation studies demonstrate that our method delivers substantial improvement over state-of-the-art methods in terms of estimation and inference on voxel-level parameters.

The flexibility in both of the proposed structured additive regression models depends crucially on (i) carefully constructed shrinkage priors that adaptively regulate model complexity based on the degree of sparsity and structure expressed in the data; and (ii) an efficient and tractable posterior computation scheme that enables meaningful inferences to be drawn on important functionals of the MCMC output.
We propose a Conditional Density Filtering (C-DF) algorithm for efficient online Bayesian inference. C-DF adapts MCMC sampling to the online setting, sampling from approximations to conditional posterior distributions obtained by propagating surrogate conditional sufficient statistics (a function of data and parameter estimates) as new data arrive. These quantities eliminate the need to store or process the entire dataset simultaneously and offer a number of desirable features. Often, these include a reduction in memory requirements and runtime and improved mixing, along with robust parameter inference and prediction. These improvements are demonstrated through several illustrative examples including an application to high dimensional compressed regression. Finally, we show that C-DF samples converge to the target posterior distribution asymptotically as sampling proceeds and more data arrives.

2.1 Introduction

Modern data is increasingly high dimensional, both in the number of observations $n$ and the number of predictors measured $p$. Statistical methods increasingly make use
of low-dimensional structure assumptions to combat the curse of dimensionality (e.g., sparsity assumptions). When the number of observations is truly massive, however, data processing and computational complexity bottlenecks render many learning algorithms infeasible. Bayesian methods provide a natural probabilistic characterization of uncertainty in the parameters and in predictions, but few scalable Bayesian inference algorithms have guarantees on accuracy. Here, sampling algorithms remain the gold standard, with the overwhelming emphasis being on Markov chain Monte Carlo (MCMC). State of the art performance is obtained in small to moderate sized problems, but MCMC fails to scale as the data and number of parameters grow (often due to the need to update many parameters with expensive likelihood evaluations).

MCMC can be extended to accommodate data collected over time by adapting the transition kernel $K_t$, and drawing a few samples at time $t$, so that samples converge in distribution to the joint posterior distribution $\pi_t$ in time (Yang and Dunson, 2013). Sequential MCMC (SMCMC) requires the full data to be stored, leading to storage and processing bottlenecks as more data arrive in time. Propagating sufficient statistics (SS) of the data when they are available can help, although modern data models routinely involve large numbers of latent variables, including observation-specific random variables. In addition, SMCMC often faces poor mixing requiring longer runs with additional storage and computation. A number of alternative strategies have been proposed for scaling MCMC to large datasets. One possibility is to parallelize computation within each MCMC iteration using GPUs or multicore architectures to free bottlenecks in updating unknowns specific to each sample and in calculating likelihoods (Medlar et al., 2013). Another possibility is to rely on Hamiltonian Monte Carlo with stochastic gradient methods used to approximate gradients with subsamples of the data (Welling and Teh, 2011). Korattikara et al. (2013) instead use sequential hypothesis testing to choose a subsample to use in approximating the acceptance ratio in Metropolis-Hastings.
In simple conjugate models, such as Gaussian state-space models, efficient updating equations can be obtained using methods related to the Kalman filter. Assumed density filtering (ADF) was proposed (Lauritzen, 1992; Boyen and Koller, 1998; Opper, 1998) to extend this computational tractability to broader classes of models. ADF approximates the posterior distribution with a simple conjugate family, leading to approximate online posterior tracking. The predominant concern with this approach is the propagation of errors with each additional approximation to the posterior in time. Expectation-propagation (EP) (Minka, 2001; Minka et al., 2009) improves on ADF through additional iterative refinements, but the approximation is limited to the class of assumed densities and has no convergence guarantees. Moreover in fairly standard settings, arbitrary approximation to the posterior through an assumed density severely underestimates parametric and predictive uncertainties. A parallel literature on online variational approximations (Hoffman et al., 2010; Wang and Dunson, 2011) focus primarily on improving batch inferences by feeding in data sequentially. Broderick et al. (2013) recently proposed a streaming variational Bayes algorithm, while Hoffman et al. (2013) combined stochastic approximation with variational inference for better scaling. However, variational methods rely on a factorized form of the posterior that typically fails to capture dependence in the joint posterior and severely underestimates uncertainty. Recent attempts to design careful online variational approximations (Luts and Wand, 2013), though successful in accurately estimating marginal densities, are limited to specific models and no theoretical guarantees on accuracy are established except for stylized cases.

Sequential Monte Carlo (SMC) (Chopin, 2002; Arulampalam et al., 2002; Lopes et al., 2011; Doucet et al., 2001) is a popular technique for online Bayesian inference that relies on propagating and resampling samples in time. Unfortunately, it is difficult to scale SMC to problems involving large $n$ and $p$ due to the need to employ very large numbers of particles to obtain adequate approximations and pre-
vent particle degeneracy. The latter is addressed through rejuvenation steps using all the data (or sufficient statistics), which becomes expensive in an online setting. One could potentially rejuvenate particles only at earlier time points, but this may not protect against degeneracy for models involving many parameters. More recent particle learning (PL) algorithms (Carvalho et al., 2010a) reduce degeneracy for the dynamic linear model, with satisfactory density estimates for parameters – but they too require propagating a large number of particles, which significantly adds to the per-iteration computational complexity. Manolopoulou et al. (2010) consider an application arising in flow cytometry where interest lies in identifying and characterizing rare cell types in data that are assumed to arise from a mixture model. Their approach involves speeding up MCMC in a two-step approach, first using a random subsample to construct a weight function directed around a region of interest, which is subsequently used to draw from a targeted subsample in a Sequential Monte Carlo framework. The targeted subsample is augmented as more information becomes available and until no new data points appear to be informative.

### 2.1.1 A new class of approximate MCMC methods

We propose a new class of Conditional Density Filtering (C-DF) algorithms that extend MCMC sampling to streaming data. Sampling proceeds by drawing from conditional posterior distributions, but instead of conditioning on conditional sufficient statistics (CSS) (Carvalho et al., 2010a; Johannes et al., 2010), C-DF conditions on surrogate conditional sufficient statistics (SCSS) using sequential point estimates for parameters along with the data observed. This eliminates the need to store the data in time (process the entire dataset at once), and leads to an approximation of the conditional distributions that produce samples from the correct target posterior asymptotically.

Section 2.4 illustrates using the C-DF algorithm in simple regression models where
the full conditional distributions are parameterized in terms of low-dimensional suffi-
cient statistics and there is no significant computational benefit to be had by using
C-DF over SMCMC. Instead, our main focus is assessing the accuracy of the C-DF
approximation in these simple settings. More complex models are considered in Sec-
tion 2.5, namely extensions of the C-DF algorithm to a dynamic linear model and
binary regression using the probit model. Here, we investigate the performance of
C-DF in settings with an increasing parameter space (e.g., data augmentation us-
ing latent variables for the probit regression example). The C-DF algorithm is also
used in the context of a Poisson mixed effects model to update the parameters for
a variational approximation to the posterior distribution. In particular, updates to
the variational parameters admit surrogate statistics which are propagated to yield
approximate online inference.

Theoretical accuracy of the C-DF approximation is established in Section 2.6 for
finite parameter settings where conditional distributions admit surrogate statistics
and may be sampled from directly or via a Metropolis step (see examples in Section
2.4). Although the theory guarantee for C-DF does not apply to the examples in
Section 2.5, the algorithm proves its versatility by producing excellent parameter
estimation, uncertainty characterization, and predictive performance. In addition,
C-DF offers potentially large runtime and memory utilization improvements as com-
pared to other sampling based competitors. Finally, the C-DF algorithm is tested
in an application to high dimensional compressed regression in Section 2.7.1, and
performance remains robust.

2.2 Sufficiency and surrogate conditional statistics

Define $\Theta = (\theta_1, \theta_2, \ldots, \theta_k)$ as the collection of unknown parameters in probability
model $P(Y|\Theta)$ and $Y \in \mathcal{Y}$, with $\theta_j \in \Psi_j$, and $\Psi_j$ denoting an arbitrary sample
space (e.g., a subset of $\mathbb{R}^p$). Data $D_t$ denotes the data observed at time $t$, while
$D^{(t)} = \{D_s, \ s = 1, \ldots, t\}$ defines the collection of data observed through time $t$. Below, $\theta_{-j} = \Theta \setminus \theta_j$ and let $\theta_{-j} = (\theta_{-j,1}, \theta_{-j,2})$ for each $j = 1, \ldots, k$. Either of $\theta_{-j,1}$ or $\theta_{-j,2}$ is allowed to be a null set.

**Definition 2.1.** $S_j^{(t)}$ is defined to be a conditional sufficient statistic (CSS) for $\theta_j$ at time $t$ if $\theta_j \perp D^{(t)} | \theta_{-j,1}, S_j^{(t)}$. Suppose $\theta_j | \theta_{-j}, D_t \overset{\mathcal{D}}{=} \theta_j | \theta_{-j,1}, h(D_t, \theta_{-j,2})^1$. Then it is easy to show that for known functions $f, h$, $S_j^{(t)} = f(h(D_1, \theta_{-j,2}), \ldots, h(D_t, \theta_{-j,2}))$. This satisfies $\theta_j | \theta_{-j}, D^{(t)} \overset{\mathcal{D}}{=} \theta_j | \theta_{-j,1}, S_j^{(t)}$, or equivalently $\theta_j \perp D^{(t)} | \theta_{-j,1}, S_j^{(t)}$.

Because $S_j^{(t)}$ depends explicitly on $\theta_{-j,2}$, its value changes whenever new samples are drawn for this collection of parameters. This necessitates storing entire data $D^{(t)}$ or available sufficient statistics. The following issues inhibit efficient online sampling: (1) sufficient statistics, when they exist, often scale poorly with the size of the parameter-space, thus creating a significant storage overhead; (2) updating $S_j^{(t)}$, $j = 1, \ldots, k$, for every iteration at time $t$ causes an immense per-iteration computational bottleneck; and (3) updating potentially massive numbers of observation-specific latent variables can lead to significant computational overhead unless conditional independence structures allow for parallelized draws. To address these challenges, we propose surrogate conditional sufficient statistics (SCSS) as a means to approximate full conditional distribution $\theta_j | \theta_{-j}, D^{(t)}$ at time $t$.

**Definition 2.2.** Suppose $\theta_j | \theta_{-j}, D_t \overset{\mathcal{D}}{=} \theta_j | \theta_{-j,1}, h(D_t, \theta_{-j,2})$. For known functions $g, h$, define $C_j^{(t)} = g(C_j^{(t-1)}, h(D_t, \hat{\theta}_{-j,2}^{(t)}))$ as the surrogate conditional sufficient statistic for $\theta_j$, with $\hat{\theta}_{-j,2}^{(t)}$ being a consistent estimator of $\theta_{-j,2}$ at time $t$. Then, $\theta_j | \theta_{-j,1}, C_j^{(t)}$ is the C-DF approximation to $\theta_j | \theta_{-j}, D^{(t)}$.

---

$^1 \theta_j | \theta_{-j}, D$ denotes the conditional distribution of parameter $\theta_j$ given all other model parameters $\theta_{-j}$ and a dataset $D$. 

---

15
If the full conditional for $\theta_j$ admits a surrogate quantity $C_j^{(t)}$, approximate sampling via C-DF proceeds by drawing $\tilde{\theta}_j \sim \tilde{\pi}_j(\cdot|\theta_{-j,1}, C_j^{(t)})$. Crucially, $C_j^{(t)}$ depends only an estimate for $\theta_{-j,2}$ which remain fixed while drawing samples from approximate full conditional $\tilde{\pi}_j(\cdot)$. This avoids having to update potentially expensive functionals $h(D^{(t)}, \theta_{-j,2})$ (i.e., CSS) for each parameter $\theta_j$ at every MCMC iteration. If the approximating full conditionals are conjugate, then sampling proceeds in a Gibbs-like fashion. Otherwise, draws from $\tilde{\pi}_j(\theta_j|\theta_{-j,1}, C_j^{(t)})$ may be obtained via a Metropolis step for a proposal distribution $p_j(\cdot|\cdot)$ for $\theta_j$ and acceptance probability

$$
\alpha_j(\theta, \theta') = \frac{\tilde{\pi}_j(\theta'|\theta_{-j,1}, C_j^{(t)}) p_j(\theta|\theta_j)}{\tilde{\pi}_j(\theta|\theta_{-j,1}, C_j^{(t)}) p_j(\theta'|\theta_j)}.
$$

(2.1)

2.3 The C-DF algorithm

Define a fixed grouping of parameters $\theta_j$, $j = 1, \ldots, p$, into sets $G_l$, $l = 1, \ldots, k$, subject to $\sum_l I(j \in G_l) = 1$, $j = 1, \ldots, p$ (i.e., predictors belong to one and only one group). The model specification and conditional independence assumptions often suggest natural parameter partitions, though alternate partitions may be more suitable for specific tasks. For streaming data, these sets are identified to maximize computational and storage gains. In some applications, one may be interested in conditional inference, updating only estimates for certain sets of nuisance parameters$^2$.

The C-DF algorithm alternates between (1) obtaining consistent estimates $\hat{\Theta}_l$ for $\Theta_l = \{\theta_j, j \notin G_l\}$, obtained by averaging $S$ draws from the approximating full-conditionals$^3$; and (2) sampling from approximate full conditionals for a group of parameters indexed by $G_l$, $\{\theta_j^l : j \in G_l, C_j^{(t)} = f_j(\hat{\Theta}_l)\}$, $1 \leq l \leq k$. This leads to effi-

$^2$ Examples for how to select parameter partitions are presented in the context of illustrating examples; see Sections 2.4 and 2.5.

$^3$ In cases where closed-form expression for the conditional mean is available, this may be used instead of Monte Carlo estimates.
cient updating equations for $C_p^q$ based on $C_p^{q-1}$ and incoming data $D_t$, resulting in scalable inference using sampling-based approximations to the posterior distribution.

A general outline for the C-DF algorithm is given in Algorithm 3.

Algorithm 1 A sketch of the C-DF algorithm for approximate MCMC

\textbf{Input:} (1) Data shard $D_t$; (2) Parameter partition index sets $G_l$, $l = 1, \ldots, k$; and (3) $C_p^{q-1}$, $j = 1, \ldots, p$. Require that $\sum_l I(j \in G_l) = 1, j = 1, \ldots, p$.

\textbf{Output:} SCSS $C_p^q$, and approximate samples $\tilde{\theta}_j^{(1)}, \ldots, \tilde{\theta}_j^{(S)}$, $j = 1, \ldots, k$.

1: function CDF.SAMPLE($D_t, C_p^{q-1}, G_l$)
2: \hspace{1em} for $l = 1 : L$ do
3: \hspace{2em} for $s = 1 : S$ do
4: \hspace{3em} for $j \in G_l$ do
5: \hspace{4em} Define $\Theta_{G_l} = \{\theta_j, j \in G_l\}$ and $\Theta_{G_l}^{(j)} = \Theta_{G_l} \setminus \theta_j$
6: \hspace{4em} Sample $\tilde{\theta}_j^{(s)} \sim \pi_j(\cdot|\Theta_{G_l}^{(j)}, C_p^q)$ \hspace{1em} $\Rightarrow [\text{or } \tilde{\theta}_j^{(s)} \sim q_j(\cdot|\Theta_{G_l}^{(j)}, C_p^q)]$
7: \hspace{3em} end for
8: \hspace{2em} end for
9: \hspace{1em} end for
10: \hspace{1em} For $j \in G_l$: Set $\hat{\theta}_j \leftarrow \text{mean}(\tilde{\theta}_j^{(1:S)})$,
11: \hspace{1em} For $j \in G_l$: Store $\tilde{\theta}_j^{(1)}, \ldots, \tilde{\theta}_j^{(S)}$ as approximate posterior samples
12: \hspace{1em} For $m \neq l$ and $j' \in G_m$: $C_j^{(m)} \leftarrow g(C_j^{(l-1)}, h(D_t, \hat{\Theta}_m))$, $\hat{\Theta}_m = \Theta \setminus \Theta_{G_l}$
13: \hspace{1em} end for
14: end function

2.4 Motivating examples

The following notation is used examples considered in this section as well as those presented in Section 2.5. Data $D_t$ denotes the data observed at time $t$, while $D^{(t)} = \{D_s, s = 1, \ldots, t\}$ defines the collection of data observed through time $t$. Where appropriate, $D_t = (X^t, y^t)$ with $X^t = (x_1^t, \ldots, x_n^t)'$ and $y^t = (y_1^t, \ldots, y_m^t)'$. Shards of a fixed size arrive sequentially over a $T = 1000$ time horizon, and 500 MCMC iterations are drawn to approximate the corresponding posterior distribution $\pi_t$ at every time point $t = 1, \ldots, T$. Where applicable, the following quantities are reported to measure estimation and inferential performance for competing methods: (i) Mean squared estimation accuracy on parameters of interest; (ii) mean squared prediction error (MSPE); and (iii) length and coverage of 95% predictive intervals. Results are
averaged over 10 independent replications with associated standard errors appearing as subscripts in Tables.

Finally, plots of kernel density estimates for marginal posterior densities on representative model parameters are shown at various time points. At time \( t \), let

\[
1 - \frac{1}{2} \int |\pi_t(\theta_j) - \tilde{\pi}_t(\theta_j)| \, d\theta_j
\]

(2.2)

be a measure of ‘accuracy’ between approximating C-DF density \( \tilde{\pi}_t \) and the full conditional distribution \( \pi_t \) obtained using sequential MCMC (SMCMC)\(^4\). Accuracy is plotted as a function of time for examples of this section, with figures appearing in Appendix A.

2.4.1 Linear regression

For the Gaussian error model, a response \( y \in \mathbb{R} \) given an associated \( p \)-dimensional predictor \( x \in \mathbb{R}^p \) is modeled in the linear regression setting as

\[
y = x' \beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2).
\]

(2.3)

A standard Bayesian analysis proceeds by assigning conjugate priors \( \beta \sim N(0, I_p) \) and \( \sigma^2 \sim IG(a, b) \), with associated full conditionals given as

\[
\sigma^2 | \beta, D^{(t)} \sim IG(a', b'), \quad a' = a + nt/2, \quad b' = b + \frac{1}{2} (S_t^{YY} - 2\beta'S_t^{XY} + \beta' S_t^{XX} \beta)
\]

\[
\beta | \sigma^2, D^{(t)} \sim N(\mu_t, \Sigma_t), \quad \Sigma_t = (S_t^{XX} / \sigma^2 + I_p)^{-1}, \quad \mu_t = \Sigma_t S_t^{XY} / \sigma^2.
\]

These are parametrized in terms of sufficient statistics \( S_t^{XX} = S_{t-1}^{XX} + X' X, \ S_t^{XY} = S_{t-1}^{XY} + X' y \) and \( S_t^{YY} = S_{t-1}^{YY} + y' y \), thus enabling efficient inference using SMCMC.

A C-DF algorithm from approximate online inference in this setting begins by defining a partition over the model parameters; here, we choose \( \Theta_{\tilde{\varphi}_2} = \{ \beta \} \) and \( \Theta_{\tilde{\varphi}_2} = \{ \sigma^2 \} \). Sampling then proceeds as:

\(^4\) As defined, accuracy ranges between 0 and 1 (larger values are better).
(1) Observe data \( D_t \) at time \( t \). If \( t = 1 \) initialize all parameters at some default values (e.g., \( \beta = 0, \sigma^2 = 1 \) assuming a centered and scaled response); otherwise set \( \hat{\sigma}_t^2 \leftarrow \hat{\sigma}_{t-1}^2 \);

(2) Define \( C_1^t = \{C_{1,1}^t, C_{1,2}^t\} \) as SCSS for \( \beta \). Update \( C_1^t \) as \( C_{1,1}^t = C_{1,1}^{t-1} + X^t X^t / \hat{\sigma}_t^2 \) and \( C_{1,2}^t = C_{1,2}^{t-1} + X^t y^t / \hat{\sigma}_t^2 \);

(3) Draw \( S \) samples from the approximate Gibbs full conditional \( \beta | \hat{\sigma}_t^2, C_1^{(t)} \sim N(\hat{\mu}_t, \hat{\Sigma}_t) \) \((\hat{\Sigma}_t = (C_{1,1}^t + I_p)^{-1}, \hat{\mu}_t = \hat{\Sigma}_t C_{1,2}^t)\), and set \( \hat{\beta}_t \leftarrow \text{mean}(\beta^{(1:S)}) \) (or use the analytical expression for the posterior mean);

(4) Define \( C_2^t = \{C_{2,1}^t, C_{2,2}^t\} \) as SCSS for \( \sigma^2 \). Update \( C_2^t \) as \( C_{2,1}^t = C_{2,1}^{t-1} + \hat{\beta}_t^t X^t X^t \hat{\beta}_t \) and \( C_{2,2}^t = C_{2,2}^{t-1} + \hat{\beta}_t^t X^t y^t \);

(5) Draw \( S \) samples from the approximate Gibbs full conditional \( \sigma^2 | \hat{\beta}_t, C_2^{(t)} \sim IG(a', b + (S^t_{Y'Y} - 2C_{2,2}^t + C_{2,1}^t)/2) \), and set \( \hat{\sigma}_t^2 \leftarrow \text{mean}(\sigma^{2(1:S)}) \) (or use the analytical expression for the posterior mean).

Data is generated using predictors drawn from \( U(0, 1) \), with true parameters \( \beta_0 = (1.00, 0.50, 0.25, -1.00, 0.75) \) and \( \sigma_0 = 5 \). Density estimates for model parameters are displayed at \( t = 200, 500 \), with accuracy comparisons in Figure A.1 validating that approximate C-DF draws converge to the true stationary distribution in time. Excellent parameter MSE and coverage using the C-DF algorithm is reported in Table 2.1.

### 2.4.2 One-way Anova model

Consider the one-way Anova model with \( k \) fixed ‘treatment’ groups

\[
\begin{align*}
    y_{ij} &= \zeta_i + \epsilon_{ij}, \quad \epsilon_{ij} \sim N(0, \sigma^2) \\
    \zeta_i &\sim N(\mu, \tau^2), \quad i = 1, \ldots, k
\end{align*}
\]  

(2.4)
Table 2.1: Inferential performance for C-DF and SMCMC for parameters of interest. Coverage and length are based on 95% credible intervals and is averaged over all the $\beta_j$’s ($j = 1, ..., 5$) and all time points and over 10 independent replications. We report the time taken to produce 500 MCMC samples with the arrival of each data shard. MSE along with associated standard errors are reported at different time points.

<table>
<thead>
<tr>
<th></th>
<th>Avg. coverage $\beta$</th>
<th>Length</th>
<th>Time (sec)</th>
<th>MSE  $= \sum_{j=1}^{p} (\hat{\beta}_t - \beta_0)^2/p$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$t = 200$</td>
</tr>
<tr>
<td>C-DF</td>
<td>1.0</td>
<td>0.600,01</td>
<td>95.4,12</td>
<td>0.270,001</td>
</tr>
<tr>
<td>SMCMC</td>
<td>1.0</td>
<td>0.600,01</td>
<td>119.4,64</td>
<td>0.120,001</td>
</tr>
</tbody>
</table>

Figure 2.1: Kernel density estimates for posterior draws using SMCMC and the C-DF algorithm at $t = 200, 500$. Shown from left to right are plots of model parameters $\beta_1$, $\beta_4$, and $\sigma^2$, respectively.

with default priors $\pi(\mu) \propto 1$, $\tau^2 \sim \text{IG}(a, b)$, and $\sigma^2 \sim \text{IG}(\alpha, \beta)$. As was the case in the linear regression model, here sufficient statistics may propagated to yield an efficient SMCMC sampler. With the arrival of a new data shard at time $t$, group-specific sufficient statistics are updated as $S_t^i = S_{t-1}^i + \sum_{j=1}^{n} y_{ij}^2$ and $S_t^{2(t)} = S_t^{2(t-1)} + ||y_t||^2$, $i = 1, \ldots, k$. At time $t$, inference for SMCMC proceeds by drawing from the following full conditionals distributions:

$\zeta_i | \sigma, \mu, \tau, y \sim N\left(\frac{\tau^2 S_t^i + \sigma^2 \mu}{n \tau^2 + \sigma^2}, \frac{\tau^2 \sigma^2}{n \tau^2 + \sigma^2}\right)$, $\sigma^2 | \zeta, y \sim \text{IG}\left(\alpha + \frac{nkt}{2}, \beta + \frac{\sum_{i=1}^{k} (S_{t}^{2(i)} - 2\zeta_i S_t^i + n\zeta_i^2)}{2}\right)$

$\mu | \zeta, \tau \sim N\left(\frac{\sum_{i=1}^{k} \zeta_i}{k}, \frac{\tau^2}{k}\right)$,

$\tau^2 | \zeta, \mu \sim \text{IG}\left(a + \frac{k}{2}, b + \frac{\sum_{i=1}^{k} (\zeta_i - \mu)^2}{2}\right)$.

(2.5)

For the C-DF algorithm, a natural partition suggested by the hierarchical struc-
ture of model (2.4) is $\Theta_{g_1} = \{\zeta, \sigma^2\}$ and $\Theta_{g_2} = \{\mu, \tau^2\}$. For this parameter partition, modified full conditionals are defined in terms of surrogate quantities as well as the previously defined group-specific sufficient statistics. Approximate inference for C-DF then proceeds as

1. Observe data $y_1^t, \ldots, y_k^t$ at time $t$. If $t = 1$, set $\zeta_t = 0, \sigma = \text{sd}(\text{vec}(y_1, \ldots, y_k))$, $\mu = \text{mean}(\text{vec}(y_1, \ldots, y_k)), \tau = 1$. Otherwise, set $\hat{\mu}_t \leftarrow \hat{\mu}_{t-1}, \hat{\tau}_t \leftarrow \hat{\tau}_{t-1}$;

2. Update surrogate statistic $C_t^i$ component-wise as $C_{1t}^i \leftarrow C_{1t}^{i-1} + \hat{\tau}_t^2 S_i^t, \ i = 1, \ldots, k$;

3. For $s = 1, \ldots, S$: draw from (a) modified full conditional $\zeta_i|\sigma, \hat{\mu}_t, \hat{\tau}_t, C_t^i, \ i = 1, \ldots, k$; and (b) $\sigma^2|\zeta,y$ as given in (2.5). C-DF full conditional $\zeta_i|\sigma, \hat{\mu}_t, \hat{\tau}_t, C_t^1$ is given by

$$
\zeta_t|\sigma, \hat{\mu}_t, \hat{\tau}_t, C_t^1 \sim N\left(\frac{C_{1t}^i + \sigma^2 \hat{\mu}_t}{nt\hat{\tau}_t^2 + \sigma^2}, \frac{\hat{\tau}_t^2 \sigma^2}{nt\hat{\tau}_t^2 + \sigma^2}\right), \ i = 1, \ldots, k;
$$

(2.6)

4. Set $\hat{\zeta}_t \leftarrow \text{mean}(\zeta^{(1:S)}), \ i = 1, \ldots, k$, and $\hat{\sigma}^2 \leftarrow \sigma^{2(1:S)}$. Update surrogate statistic $C_t^2 = (C_{2,1}^t, C_{2,2}^t)$: $C_{2,1}^t \leftarrow ||\hat{\zeta}||^2$ and $C_{2,2}^t \leftarrow \sum_{i=1}^k \hat{\zeta}_i$;

5. For $s = 1, \ldots, S$: draw from modified full conditional distributions (a) $\mu|\zeta, C_t^2 \sim N(C_{2,2}^t/k, \tau^2/k)$ and (b) $\tau^2|\mu, C_t^2 \sim IG(a + k/2, b + (C_{2,1}^t - 2\mu C_{2,2}^t + k\mu^2)/2)$;

6. Finally, set $\hat{\mu}_t \leftarrow \text{mean}(\mu^{(1:S)})$ and $\hat{\tau}_t \leftarrow \text{mean}(\tau^{(1:S)})$.

Data is generated according to model (2.4) with parameters $\zeta_i \overset{iid}{\sim} N(4, \tau^2 = 0.01)$ and $\sigma = 10$. Figure 2.2 displays kernel density estimates for posterior draws using SMCMC and the C-DF algorithm at $t = 200,500$. ADF is added as an additional competitor for this example. Here, the joint posterior over $\theta = (\zeta_1, \ldots, \zeta_k, \log(\sigma^2))$ is approximated in time assuming a multivariate-normal density $\pi_t(\theta) \sim N(\mu_t, \Sigma_t)$. To begin, we integrate over hyper-parameters $\mu, \tau^2$ to obtain marginal prior $\pi(\zeta, \log(\sigma^2))$. 

21
For $t > 1$, the approximate posterior at time $(t - 1)$ becomes the prior at $t$, and parameters $\mu_t, \Sigma_t$ are updated using Newton-Raphson steps in the sense of McCormick et al. (2012). In particular, $\Sigma_t = (-\nabla^2 \ell(\mu_{t-1}))^{-1}$ and $\mu_t = \mu_{t-1} + \Sigma_{t-1} \nabla \ell(\mu_{t-1})$, with $\ell(\theta) = \log\{p(y|\theta)\pi(\theta)\}$.

ADF is extremely sensitive to good calibration for $\sigma^2$, without which estimates for $\zeta$ are far from the truth even at $t = 500$. Optimal performance is obtained by using the first data shard and performing the ADF approximation at $t = 1$ until convergence. Thereafter, parameters estimates are propagated as described above. Though resulting parameters point estimates are accurate, ADF severely underestimates uncertainty as seen by the length and coverage of resulting 95% credible intervals reported in Table 2.2. This occurs in-part because ADF makes a global approximation on the joint posterior (restricting the propagation of uncertainty in a very specific way), whereas C-DF makes local approximations to set of full conditional distribution. The C-DF approximation results in steadily increasing accuracy along with good parameter inference as shown in Table 2.2.

Table 2.2: Inferential performance for C-DF, SMCMC, and ADF for parameter $\zeta$. Coverage is based on 95% credible intervals averaged over all time points, all $\zeta$ and over 10 independent replications. We report the time taken to produce 500 MCMC samples with the arrival of each data shard. MSE along with associated standard errors are reported at different time points.

<table>
<thead>
<tr>
<th></th>
<th>Avg. coverage $\zeta$</th>
<th>Length</th>
<th>Time (sec)</th>
<th>MSE</th>
<th>$t = 200$</th>
<th>$t = 400$</th>
<th>$t = 500$</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-DF</td>
<td>0.87$_{0.09}$</td>
<td>0.53$_{0.005}$</td>
<td>85.05$_{25}$</td>
<td>0.77$_{0.22}$</td>
<td>0.29$_{0.11}$</td>
<td>0.26$_{0.15}$</td>
<td></td>
</tr>
<tr>
<td>SMCMC</td>
<td>0.92$_{0.10}$</td>
<td>0.52$_{0.002}$</td>
<td>119.4$_{8.43}$</td>
<td>0.41$_{0.05}$</td>
<td>0.22$_{0.14}$</td>
<td>0.20$_{0.16}$</td>
<td></td>
</tr>
<tr>
<td>ADF</td>
<td>0.36$_{0.23}$</td>
<td>0.08$_{0.02}$</td>
<td>0.88$_{0.01}$</td>
<td>0.42$_{0.18}$</td>
<td>0.28$_{0.12}$</td>
<td>0.27$_{0.11}$</td>
<td></td>
</tr>
</tbody>
</table>

2.5 Advanced data models

The C-DF algorithm introduced in Section 2.2 modifies the set of full conditional distributions to depend on propagated surrogate quantities which yields an approxi-
Figure 2.2: Row #1 (left to right): Kernel density estimates for posterior draws of \(\zeta_1\), \(\zeta_5\), \(\zeta_{10}\) using SMCMC and the C-DF algorithm at \(t = 200, 500\); Row #2 (left to right): Kernel density estimates for model parameters \(\tau^2\), and \(\sigma^2\) at \(t = 500\).

mate kernel. This enables efficient online MCMC, with guarantees on correctness of C-DF samples as data accrue over time (see Section 2.6). However, in settings where (a) the model involves an increasing number of parameters growing with the sample size, or (b) surrogate quantities for some (or all) of the full conditional distributions do not exist and alternate distributional approximations are employed (e.g., Laplace approximation, variational Bayes, etc.), then theoretical consistency for the parameters often do not hold. This section considers extensions of the C-DF algorithm to three models that face one or more of these difficulties.

Section 2.5.1 presents an application to a dynamic linear model for a continuous response centered on a first-order auto-regressive process. The ‘forward-filtering and backward-sampling’ Kalman filter updates for the latent process enables online
posterior computation, albeit with an increasing computational cost as time goes on. Section 2.5.2 considers an application to binary regression where the conditional posterior distribution over the regression coefficients does not assume a closed form. For logistic regression, a variational approximation to the posterior introduces additional variational parameters for each observation to obtain a lower-bound for the likelihood (Jaakkola and Jordan, 1997). Additional non-conjugate message passing approximations have been considered in Braun and McAuliffe (2010) and Tan et al. (2014). One may also resort to ADF using a Laplace approximation to the posterior over regression coefficients and propagating associated mean and covariance parameters in time. However, the latter are known to severely underestimate parameter uncertainty. Fortunately, data augmentation via the probit model enables conditionally conjugate sampling. We discuss how to use the C-DF algorithm to overcome the computational and storage bottlenecks arising in both examples.

Finally, Section 2.5.3 considers an application to Poisson regression where no augmentation scheme is available and the full conditional distributions also do not admit surrogate quantities. Here, we demonstrate how the online strategy of (Luts and Wand, 2013) is a special case of the C-DF algorithm applied to a variational approximation of the posterior distribution.

2.5.1 Dynamic linear model

We consider the first-order dynamic linear model (West and Harrison, 2007), namely

\[ y_{t+1} \sim N(\theta_{t+1}, \sigma^2), \quad \theta_{t+1} \sim N(\phi \theta_t, \tau^2) \]

where noisy observations \( y_t, t \geq 1 \) are modeled as arising from an underlying stationary AR(1) process with lag parameter \( \phi, |\phi| < 1 \). Default priors \( \sigma^2 \sim IG(a_0, b_0), \tau^2 \sim IG(c_0, d_0), \phi \sim U(-1, 1) \) are chosen to complete the hierarchical model, and assume \( \theta_0 \sim N(0, h_0) \), the stationary distribution for the latent process. The full conditional
distributions are given by

\[
[\theta_{t+1}] \sim N\left( \frac{\sigma^{-2} y_{t+1} + \tau^{-2} \phi \theta_{t}}{\sigma^{-2} + \tau^{-2}}, \frac{1}{\sigma^{-2} + \tau^{-2}} \right);
\]

\[
[\theta_{s}] \sim N\left( \frac{\sigma^{-2} y_{s} + \tau^{-2} \phi (\theta_{s-1} + \theta_{s+1})}{\sigma^{-2} + (\phi^2 + 1) \tau^{-2}}, \frac{1}{\sigma^{-2} + (\phi^2 + 1) \tau^{-2}} \right), \quad 1 \leq s \leq t,
\]

\[
[\sigma^2] \sim \text{IG}(a_t, b_t), \quad a_{t+1} = a_t + 1/2, \quad b_{t+1} = b_t + (y_{t+1} - \theta_{t+1})^2/2,
\]

\[
[\tau^2] \sim \text{IG}(c_t, d_t), \quad c_{t+1} = c_t + 1/2, \quad d_{t+1} = d_t + (\theta_{t+1} - \phi \theta_t)^2/2,
\]

\[
[\phi] \propto \pi(\phi) N\left( \frac{\sum_{s=1}^{t} \theta_s \theta_{s-1}}{\sum_{s=1}^{t} \theta_s^2}, \frac{\tau^2}{\sum_{s=1}^{t-1} \theta_s^2} \right) I(|\phi| < 1).
\]

Forecasting future trajectories of the response is a common goal, hence good characterization of the distribution over $\theta_t, \phi, \tau^2$ is of interest. Hence, retrospective sampling of $\{\theta_s, s < t\}$ is only meaningful in as much as it propagates uncertainty to the current current time point.

To extend the use of the C-DF algorithm in this growing parameter setting, we propose sampling of the latent process over a moving time-window of size $b$. This eliminates the propagation of errors that might otherwise result from poor estimation at earlier time points. As the sampling window shifts forward, trailing latent parameters are fixed at their most recent point estimates. Parameter partitions for the C-DF algorithm must therefore also evolve dynamically, and are defined at time $t$ as $\Theta_{t_1} = \{\theta_t, \ldots, \theta_{t-b+1}, \tau^2, \sigma^2, \phi\}$, $\Theta_{t_2} = \{\theta_{t-b}, \ldots, \theta_1\}$, $t > b$. Unlike previous examples, conditional distribution $\pi(\phi|\cdot)$ is not available in closed-form. Nevertheless, propagated surrogate quantities (SCSS) enable approximate MCMC draws to be sampled from C-DF full conditional $\tilde{\pi}(\phi|\cdot)$ via Metropolis-Hastings. Here, steps for approximate MCMC using the C-DF algorithm in the context of a Metropolis-within-Gibbs sampler are:

1. At time $t$ observe $y_t$;

2. If $t \leq b$ : Draw $S$ samples for $(\theta_1, \ldots, \theta_t, \tau^2, \sigma^2, \phi)$ from the Gibbs full condition-
als. In addition, $C_1^t = C_2^t = C_3^t = C_4^t = 0, \ t \leq b$.

(3) If $t > b$ : Repeat $S$ times the following

(a) for $t - b < s < t$, draw sequentially from $[\theta_s|\theta_{s-1}, \theta_{s+1}, -]$ and finally from $[\theta_t|\theta_{t-1}, -]$, noting that $\theta_{t-b} = \hat{\theta}_{t-b}$;

(b) draw $\tau^2 \sim IG(a_t, b_t)$, and $b_t = C_4^{(t-1)} - 2\phi C_3^{(t-1)} + \phi^2 C_2^{(t-1)} + \frac{1}{2}((\theta_{t-b+1} - \phi \hat{\theta}_{t-b})^2 + \sum_{j=2}^{b} (\theta_{t+j-b} - \phi \theta_{t+j-b})^2);$

(c) draw $\sigma^2 \sim IG(a_t, C_1^{(t-1)} + \frac{1}{2} \sum_{j=1}^{b} (y_{t+j-b} - \theta_{t+j-b})^2);$

(d) sample $\phi \sim \tilde{\pi}(\cdot|C_3^{(t-1)}, C_4^{(t-1)}, \theta_{(t-b+1):t})$ via a Metropolis-Hastings step.

(4) Set $\hat{\theta}_{t-b} \leftarrow \text{mean}(\theta_{t-b})$, and then update surrogate quantities $C_1^t \leftarrow C_1^{(t-1)} + \frac{1}{2}(y_{t-b} - \hat{\theta}_{t-b})^2, C_2^t \leftarrow C_2^{(t-1)} + \hat{\theta}_{t-b-1}, C_3^t \leftarrow C_3^{(t-1)} + \hat{\theta}_{t-b-1} \hat{\theta}_{t-b}, C_4^t = C_4^{(t-1)} + \frac{\hat{\theta}_{t-b}^2}{2}.$

We vary signal-to-noise ratio $\tau^2/\sigma^2$ for the data generating process in simulation experiments. A number of high-signal cases were examined, and results are reported for a representative case with $\tau = \sqrt{2}, \sigma = 0.1$. A sufficiently large window size $b$ is necessary to prevent C-DF from suffering due to poor estimation at early time-points, causing propagation of error in the defined surrogate quantities. For the PL competitor, 100 particles were propagated in time, and hence we fix $b = 100$ as well. Kernel density estimates for $\theta_t$ using PL and the C-DF algorithm are shown in Figure 2.3. In the case of C-DF, estimates for all model parameters are found to be concentrated near their true values, whereas for PL, $\theta_t$ at different times are centered correctly, albeit with much higher posterior variance (presumably due to poor estimation of noise parameters $\tau^2, \sigma^2$). In addition to being 35% faster than PL, average coverage for the latent AR(1) process using the C-DF algorithm is near 80% with credible intervals roughly 10 times narrower than PL (see Table 2.3). C-DF is
substantially more efficient in terms of memory and storage utilization, requiring only 6% of what PL uses for an identical simulation. Finally, C-DF produces accurate estimates for latent parameters $\tau^2, \phi$ (in contrast to PL), although their posterior spread appears somewhat over shrunk. This may be remedied by using larger window size $b$ (at the expense of increased runtime) depending on the task at hand.

Table 2.3: Inferential performance for C-DF and Particle Learning. Coverage and length are based on 95% credible intervals for $\theta_t$ averaged over all time points and 10 independent replications. For truth $\theta_{t0}$ at time $t$, we report $\text{MSE} = \frac{1}{T_n} \sum_{t=1}^{T_n} (\hat{\theta}_t - \theta_{t0})^2$. We report the time taken to run C-DF with 50 Gibbs samples at each time for $\tau^2$, $\theta$, $\sigma^2$ and 500 MH samples for $\phi$.

<table>
<thead>
<tr>
<th></th>
<th>Avg. coverage $\theta$</th>
<th>Length</th>
<th>Time (sec)</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-DF</td>
<td>0.78$_{0.10}$</td>
<td>0.33$_{0.11}$</td>
<td>1138.60$_{0.10}$</td>
<td>0.011$_{0.001}$</td>
</tr>
<tr>
<td>PL</td>
<td>1.00$_{0.00}$</td>
<td>3.36$_{0.46}$</td>
<td>1750.58$_{0.10}$</td>
<td>0.096$_{0.027}$</td>
</tr>
</tbody>
</table>

Table 2.4: Computational and storage requirements for the Dynamic Linear Model using C-DF and PL. $C_{i,j}^t$, is the $i$-th CSS corresponding to the $j$-th particle in PL, $i = 1 : 4$, $j = 1 : N$, $N = 100$ is the number of particles propagated by PL, and $G = 500$ is the number of Metropolis samples used by both PL and C-DF. Memory in terms of RAM used to store and propagate SCSS and CSS for C-DF and PL is reported. Sampling and update complexities are in terms of big-O.

<table>
<thead>
<tr>
<th></th>
<th>Stats</th>
<th>Data</th>
<th>Sampling</th>
<th>Updating</th>
<th>Memory (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-DF</td>
<td>$C_i^t$</td>
<td>${y_{i,t}}_{t=1}^{nt-b}$</td>
<td>$S(N + G)$ flops</td>
<td>$N$ flops</td>
<td>128</td>
</tr>
<tr>
<td>PL</td>
<td>$C_{i,j}^t$</td>
<td>${y_{i,t}}_{t=1}^{b}$</td>
<td>$NG$ flops</td>
<td>$N$ flops</td>
<td>3330</td>
</tr>
</tbody>
</table>

2.5.2 Binary regression model

We consider an application of the C-DF algorithm for the probit model $\text{Pr}(y_i = 1|x_i) = \Phi(x'_i\beta)$, with standard normal distribution function $\Phi(\cdot)$. The Albert and Chib (1993) latent variable augmentation applies Gibbs sampling, with $y_i = \text{sign}(z_i)$, and $z_i|\beta \sim N(x'_i\beta, 1)$. Assuming a conditionally conjugate prior $\beta \sim N(0, \Sigma_\beta)$, the Gibbs sampler alternates between the following steps: (1) conditional on latent
variables $z = (z_1, z_2, \ldots)'$, sample $\beta$ from its $p$-variate Gaussian full conditional; and (2) conditional on $\beta$ and the observed binary response $y$, impute latent variables from their truncated normal full conditionals. However, imputing latent variables $\{z_i : i \geq 1\}$ presents an increasing computational bottleneck as the sample size increases, as does recalculating the conditional sufficient statistics (CSS) for $\beta$ given these latent variables at each iteration. C-DF alleviates this problem by imputing latent scores for the last $b$ training observations and propagating surrogate quantities. “Budget” $b$ is allowed to grow with the predictor dimension, and as a default we set $b = p \log p$. For data shards of size $n$, define $I_t = \{i > nt - b\}$ as an index over the final $b$ observations at time $t$. Parameter partitions in this setting are dynamic as in Section 2.5.1, with $\Theta_{G_1} = \{\beta; z_i, i \in I_t\}$ and $\Theta_{G_2} = \{z_i : i \leq nt - b\}$. The C-DF algorithm proceeds as follows:
(1) Observe data $X^t, y^t$ at time $t$;

(2) If $t = 1$, set $\beta = 0$, and draw $z_i \sim N(0, 1)$, $i = 1, \ldots, n$. If $t \leq b/n$, $C_t = 0$ and draw $S$ Gibbs samples for $(z_1, \ldots, z_n, \beta)$ from the full conditionals.

(3) If $t > b/n$, set $\beta \leftarrow \hat{\beta}_{t-1}$, update sufficient statistic $S^X_t \leftarrow S^X_{t-1} + X^t X^t$, and compute $\Sigma^X_t = (S^X_t + \Sigma^{-1}_\beta)^{-1}$;

(4) For $s = 1, \ldots, S$: (a) draw $z_i \sim N(x_i^t \beta)$, $i \in I_t$, and define $z_{I_t}$ and $X_{I_t}$ respectively as the collection of latent draws and data for moving window $b$;

(b) draw $\beta | z_{I_t}, C_{t-1} \sim N(\Sigma^X_tC_t, \Sigma^X_tC_t)$, where $C_t(z_{I_t}) = C_{t-1} + X_{I_t}^t z_{I_t}^{(s)}$;

(5) Set $\hat{\beta}_t \leftarrow \text{mean}(\beta^{(s)})$. For $u_i(\beta) = y_i \phi(-x_i^t \beta)/\Phi(y_i x_i^t \beta)$, the trailing $n$ latent scores within moving window $b$ are fixed at their expected value, namely $\hat{z}_i \leftarrow x_i^t \hat{\beta}_t + u_i(\hat{\beta}_t)$, $i \in I^{\text{out}}_t = \{(nt + b - 1) : (n(t + 1) + b - 1)\}$;

(6) Denote $X_{I^{\text{out}}_t}$ and $z_{I^{\text{out}}_t}$ as predictor data and latent scores for the “outgoing” set of data indexed by $I^{\text{out}}_t$. Update surrogate CSS $C_t \leftarrow C_{t-1} + X_{I^{\text{out}}_t} \hat{z}_{I^{\text{out}}_t}$, and set $\Theta_{G_1} \leftarrow \Theta_{G_1} \cup \hat{z}_{I^{\text{out}}_t}$.

We report simulation results for the following examples:

(Case 1) $(p, b, n, t) = (100, 500, 25, 100)$: $\beta_{j,0} \sim U(-3/4, 3/4)$ for $j \in [11, 100]$;

(Case 2) $(p, b, n, t) = (500, 3500, 100, 100)$: $\beta_{j,0} \sim U(-1/3, 1/3)$ for $j \in [11, 200]$.

The first 10 regression coefficients are $(3.5, -3.5, -2.0, 2.0, -1.5, 1.5, -1.5, 1.5, -1.0, 1.0)$, with $\beta_{j,0} = 0$ for $j > 200$ in case 2. Data are generated as $x_{ij} \sim N(0, \sigma = 0.25)$, and $P(y_i = 1) = \Phi(x_i^t \beta)$. Table 2.5 summarizes inferential performance for the regression parameters in each of the simulated cases. In case 2, although coverage for predictors with large coefficients (i.e., $\beta_1$ and $\beta_2$) is less than the nominal value, the average coverage across all predictors produced by C-DF is 70% despite the high
dimension with a significant number of “noisy” predictors. In addition, C-DF has very good mean-square estimation of parameter coefficient in both cases. Figure 2.4 compares kernel density estimates of marginal posteriors at \( t = 200, 500 \) for both cases.

Table 2.5: Inferential performance for C-DF and SMCMC in simulation studies \((p, b, n, t) = (100, 500, 25, 100)\) and \((p, b, n, t) = (500, 3500, 100, 100)\). Coverage and length are based on 95% credible intervals averaged over all predictors and over 10 independent replications. MSE is reported over all predictors, while \( \text{MSE}_{10} = \frac{1}{10} \sum_{j=1}^{10} (\hat{\beta}_j - \beta_j)^2 \). We report the total time taken to produce 500 MCMC samples after the arrival of each data shard.

<table>
<thead>
<tr>
<th>Case</th>
<th>Method</th>
<th>Avg. coverage ( \beta )</th>
<th>Length</th>
<th>Time (sec)</th>
<th>MSE_{10}</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C-DF</td>
<td>0.77_{0.08}</td>
<td>0.42</td>
<td>28.9_{2.5}</td>
<td>0.025_{0.01}</td>
<td>0.04_{0.01}</td>
</tr>
<tr>
<td></td>
<td>SMCMC</td>
<td>0.96_{0.01}</td>
<td>0.65</td>
<td>152.5_{10.3}</td>
<td>0.018_{0.01}</td>
<td>0.02_{0.01}</td>
</tr>
<tr>
<td>2</td>
<td>C-DF</td>
<td>0.70_{0.03}</td>
<td>0.23</td>
<td>461.5_{29.6}</td>
<td>0.20_{0.05}</td>
<td>0.016_{0.003}</td>
</tr>
<tr>
<td></td>
<td>SMCMC</td>
<td>0.92_{0.02}</td>
<td>0.33</td>
<td>2196.3_{170.5}</td>
<td>0.050_{0.01}</td>
<td>0.009_{0.001}</td>
</tr>
</tbody>
</table>

SMCMC scales linearly in the number of training observations and has the worst-case storage requirement. At each Gibbs iteration, draws for the latent scores is \( O(nt) \) for SMCMC compared to \( O(b) \) for C-DF. For both methods, sampling from the full conditional for \( \beta \) is \( O(p^3) \), updating sufficient statistics is \( O(np^2) \), and updating surrogate quantities is \( O(np) \). Computational and storage requirements for both methods are summarized in Table 2.6.

Table 2.6: Computational and storage requirements for the latent variable probit model using C-DF and SMCMC. Budget \( b \) represents the number of latent scores updated by C-DF when processing data shard at time \( t \). Runtime is quickly dominated by the sampling complexity which scales linearly in time for the augmented Gibbs sampler (SMCMC). Memory is reported for case 2, \((p, b, n, t) = (500, 3500, 100, 100)\). Sampling and update complexities are in terms of big-O.

<table>
<thead>
<tr>
<th>Method</th>
<th>Stats</th>
<th>Data</th>
<th>Sampling</th>
<th>Updating</th>
<th>Memory (M-bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-DF</td>
<td>( S^{XX.C} )</td>
<td>{( x_i, y_i }_{i \geq nt-b} }</td>
<td>( S(p^2 + bp) ) flops</td>
<td>( p^3 + np^2 ) flops</td>
<td>20.0</td>
</tr>
<tr>
<td>SMCMC</td>
<td>( S^{XX} )</td>
<td>{( x_i, y_i }_{i \geq 1} }</td>
<td>( S(p^2 + ntp) ) flops</td>
<td>( p^3 + np^2 ) flops</td>
<td>46.0</td>
</tr>
</tbody>
</table>
Figure 2.4: Kernel density estimates for posterior draws using SMCMC and the C-DF algorithm at $t = 50, 100$. Shown from left to right are plots of model parameters $\beta_1$, $\beta_5$, and $\beta_{10}$ (and top to bottom are case 1 and case 2), respectively.

2.5.3 Poisson mixed effects model

Consider the Poisson additive mixed effects model where count data $y^t$ with predictors $X^t$ are modeled as

$$y^t \sim \text{Poisson}(\exp(X^t \beta + Zu))$$

$$\beta \sim N(0, \sigma_\beta^2 I_k), \quad u \sim N(0, \text{diag}(\sigma_1^2 I_{k_1}, \ldots, \sigma_r^2 I_{k_r}))$$

$$\sigma_s^2 \sim \text{IG}(0.5, 1/b_s), \quad b_s \sim \text{IG}(1/2, 1/a_s^2), \quad s = 1, 2, \ldots, r.$$ 

Unfortunately, the application of the C-DF algorithm in this setting faces an immediate problem as conditional posterior distributions over model parameters have no closed-form and more seriously, do not admit surrogate quantities to propagate. However, an additional approximation made to the joint posterior does in fact enable tracking of SCSS, leading to efficient online inference.
The global approximation made by an ADF approximation to the posterior distribution in this setting tends to be overly restrictive, leading to unreliable estimation even for moderate dimensions. A less restrictive approximation seeks to obtain an “optimal” approximation to the posterior subject to ignoring posterior dependence among (subsets of) parameters. This is given by a variational approximation to the posterior over model parameters \( \beta, u, \sigma_1^2, \ldots, \sigma_r^2 \) (\( \sigma_\beta \) treated as a fixed hyperparameter), namely

\[
\pi(\beta, u, \sigma_1^2, \ldots, \sigma_r^2 | D) \approx q_1(\beta, u) q_2(\sigma_1^2, \ldots, \sigma_r^2) q_3(b_1, \ldots, b_r).
\]

Closed-form expressions for variational distributions \( q_1, q_2, q_3 \) are derived to be

\[
q_1 = N(\mu_\beta, u, \Sigma_\beta, u)
\]

\[
q_2 = \prod_{s=1}^r \text{IG}\left((k_s + 1)/2, \mu_{1/b_s} + (||\mu_{u_s}||^2 + \text{tr}(\Sigma_{u_s}))/2\right)
\]

\[
q_3 = \prod_{s=1}^r \text{IG}(1, \mu_{1/\sigma_s^2} + 1/\sigma_s^2)
\]

where \( \mu_u \) and \( \Sigma_u \) denote the mean and covariance corresponding to the random effect vector \( \mu \) under approximating density \( q_1 \), and expressions for \( \mu_{\beta,u} \) and \( \Sigma_{\beta,u} \) are given in step (2) of the C-DF algorithm below. Additionally, \( \mu_{1/\sigma_s^2} = \int 1/\sigma_s^2 q(\sigma_s^2) d\sigma_s^2 \), with \( \mu_{1/b_s} \) defined similarly. The variational distribution is completely specified in terms of \( \mu_{\beta,u}, \Sigma_{\beta,u} \) and \( \{\mu_{1/b_s}, \mu_{1/\sigma_s^2} : s = 1, \ldots, r\} \), hence the C-DF algorithm is applied directly on these parameters. In particular, one only needs point estimates for these parameters to fully specify the approximate posterior, hence the sampling steps in Algorithm 3 are here replaced by fixed-point iteration. With a partition \( \Theta_{G_1} = \{\mu_{\beta,u}, \Sigma_{\beta,u}\}, \Theta_{G_2} = \{\mu_{1/b_s}, \mu_{1/\sigma_s^2} : s = 1, \ldots, r\} \) for the parameters of the variational distribution, the C-DF algorithm proceeds as follows:

1. Observe data shard \((y^t, X^t)\) at time \( t \). If \( t = 1 \), initialize all the parameters using draws from respective priors. Otherwise set \( \mu_{\beta,u}^{(t-1)} \leftarrow \mu_{\beta,u}^{(t-1)}, \Sigma_{\beta,u}^{(t-1)} \leftarrow \)

32
\[ \Sigma^{(t-1)}_{\beta, u}, \mu^{(t-1)}_{1/\sigma^2_I}, \mu^{(t-1)}_{1/b_s}, \mu^{(t-1)}_{1/b_u}; \]

(2) Repeat for a maximum of \( S \) iterations (or until convergence):

(a) Update \( D \leftarrow \text{blockdiag}(\sigma_{\beta}^2 I_k, \hat{\mu}_{1/\sigma^2_I} I_{k_1}, \ldots, \hat{\mu}_{1/\sigma^2_I} I_{k_r}); \)

(b) Update \( w_{\beta,u} \leftarrow \exp(\mathbf{c}^T \mathbf{\mu}_{\beta,u} + \frac{1}{2} \mathbf{c}^T \Sigma_{\beta,u} \mathbf{c}) \), with \( \mathbf{c} = [\mathbf{X}^T, \mathbf{Z}]; \)

(c) Set \( H_1 = C_{1,1}^{(t-1)} + \mathbf{c}^T \mathbf{y}, H_2 = C_{1,2}^{(t-1)} + \mathbf{c}^T w_{\beta,u}, \) and \( H_3 = C_{1,3}^{(t-1)} + w_{\beta,u} \mathbf{c}^T \mathbf{c}'; \)

(d) Update \( \mathbf{\mu}_{\beta,u} \leftarrow \mathbf{\mu}_{\beta,u}^{(t-1)} + \Sigma_{\beta,u}(H_1 - H_2 - D \mathbf{\mu}_{\beta,u}), \) and \( \Sigma_{\beta,u} \leftarrow (H_3 + D)^{-1}. \)

(3) Set \( \mathbf{\mu}_{\beta,u}^t, \Sigma_{\beta,u}^t \) at the final values after \( S \) iterations;

(4) Repeat for a maximum of \( S \) iterations (or until convergence):

(a) For \( s = 1, \ldots, r: \) update \( \mu_{1/b_s} \leftarrow (\mu_{1/\sigma^2_s} + 1/\sigma^2_s)^{-1} \) and \( \mu_{1/\sigma^2_s} \leftarrow (k_s + 1)/(2 \mu_{1/b_s} + ||\mathbf{\mu}_{u}^t||^2 + \text{tr}(\Sigma_{u}^t)). \)

(5) Set \( \mu_{1/\sigma^2_s}^t, \mu_{1/b_s}^t \) at the final values after \( S \) iterations;

(6) Update surrogate quantities \( C_{1,1}^t \leftarrow C_{1,1}^{(t-1)} + \mathbf{c}^T \mathbf{y}; \)

\( C_{1,2}^t \leftarrow C_{1,2}^{(t-1)} + \mathbf{c}^T w_{\beta,u}^t; \)

\( C_{1,3}^t \leftarrow C_{1,3}^{(t-1)} + w_{\beta,u} \mathbf{c}^T \mathbf{c}. \)

With the arrival of new data, \( S \) fixed-point iterations update parameters estimates for the approximating distributions. Consistent with the definition of SCSS, an update to the parameters for \( \Theta_{G_1} \) under \( q_1 \) may involve estimates from the previous time-point, in addition to estimates for parameters \( \Theta_{G_2} \) and vice versa. This online sampling scheme has excellent empirical performance, both in terms of parametric inference and prediction (Luts and Wand, 2013). This establishes the broad applicability for propagating surrogate quantities (SCSS) using the C-DF algorithm in a non-conjugate setting using variational approximation methods.
2.6 Theory guarantee for the C-DF algorithm

We establish convergence behavior for a general class of approximate MCMC algorithms of which the C-DF algorithm is shown to be a special case. We first characterize the limiting approximation error that results when a kernel having the desired target stationary distribution is approximated by another kernel in the finite sample setting. This result is general and is not limited to any specific class of approximations. Next, we show that such kernel approximations improve under an increasing sample size to yield draws from the exact posterior asymptotically.

2.6.1 Notation and framework

Let \( \pi_t(\cdot|D^{(t)}) \) denote the posterior distribution having observed data \( D^{(t)} \) through time \( t \). A sequence of probability measures \( \pi_t(\cdot|D^{(t)}) \) is defined on a corresponding sequence of measure spaces, \( (H_t, \mathcal{H}_t), \ t \geq 1 \). Below, we take \( H_t = \mathbb{R}^p \), with \( \mathcal{H}_t = \mathcal{B}(\mathbb{R}^p) \) denoting the Borel \( \sigma \)-algebra on \( \mathbb{R}^p \) for a fixed \( p \)-dimensional parameter space \( \Theta = (\theta_1, \cdots, \theta_p) \). \( \pi_t \) admits density \( \pi_t(\Theta) \) with respect to the Lebesgue measure \( d\nu(\Theta) = d\nu_1(\theta_1) d(\theta_2), \ldots, d\nu_p(\theta_p) \). For a transition kernel \( T_t : \mathbb{R}^p \times \mathbb{R}^p \to \mathbb{R}^+ \) at time \( t \), we assume

1. \( T_t(x, \cdot) \) is a probability measure for all \( x \in \mathbb{R}^p \)

2. \( T_t(\cdot, A) \) is a measurable function w.r.t the \( \sigma \)-algebra for all \( \nu \)-measurable sets \( A \).

A function \( f_t : \mathbb{R}^p \to \mathbb{R}^+ \) defined over \( \mathcal{H}_t \) is the invariant distribution of \( T_t \) if

\[
 f_t(\Theta') = \int T_t(\Theta, \Theta') f_t(\Theta) \, d\nu(\Theta). \tag{2.7}
\]

We study the convergence of a sequence of distributions in total variation norm, namely \( d_{TV}(\mu_1, \mu_2) = \sup_A |\mu_1(A) - \mu_2(A)| \).
2.6.2 Finite sample error bound for approximate kernels

In the finite sample setting, transition kernels and stationary distributions omit the subscript $t$. We begin by characterizing the propagation of error induced by approximating one kernel with another (valid) kernel.

**Lemma 2.3.** Let $K$ be a kernel approximated by a kernel $T$ s.t.

$$\sup_{\Theta} ||K(\Theta, \cdot) - T(\Theta, \cdot)||_{TV} \leq \rho$$

for some constant $\rho > 0$. Let $\mu_1, \mu_2$ denote the stationary distributions for $T$ and $K$, respectively, and assume $||T^{(r)} - \mu_1||_{TV} \to 0$ and $||K^{(r)} - \mu_2||_{TV} \to 0$. Then there exists an $r_0$ s.t.

$$\begin{align*}
\sup_{\Theta} ||K^{(r)}(\Theta, \cdot) - T^{(r)}(\Theta, \cdot)||_{TV} &\leq r\rho, & \text{if } r \leq r_0 \\
\sup_{\Theta} ||K^{(r)}(\Theta, \cdot) - T^{(r)}(\Theta, \cdot)||_{TV} &\leq \min\{\rho r, 2||\mu_1 - \mu_2||_{TV}\}, & \text{if } r > r_0.
\end{align*}$$

(2.8)

When the sample size is small and the data is observed sequentially or at once, the approximating (C-DF) transition kernel remains unchanged once surrogate quantities (SCSS) have been calculated over all of the observed data. Lemma 2.3 states that the (C-DF) approximation error increases initially before stabilizing. If additional data is observed in time (e.g., in the streaming data setting), the C-DF kernel (as defined below) can be shown to generate draws from the exact posterior distribution asymptotically under a few additional assumptions.

2.6.3 Convergence for a general approximation class

Assume that a $p$-dimensional model parameter $\Theta$ is partitioned into two groups, $\Theta_1 = (\theta_{11}, \cdots, \theta_{1p_1})' \in \mathbb{R}^{p_1}$ and $\Theta_2 = (\theta_{21}, \cdots, \theta_{2p_2})' \in \mathbb{R}^{p_2}$ where $p = p_1 + p_2$.

As described in Section 2.2, the C-DF algorithm relies on the existence of surrogate quantities (SCSS) that parameterize the approximating kernels to the full conditional distributions. Within this framework and without loss of generality, we assume that $\theta_{1i} | \theta_{1, -i}, \hat{\Theta}_2, i = 1, \ldots, p_1$ are conditionally conjugate distributions.
The C-DF transition kernel

Assuming two sequences of estimators \(\{\hat{\Theta}_{1,t}\}_{t \geq 1}, \{\hat{\Theta}_{2,t}\}_{t \geq 1}\), the approximating C-DF kernel \(T_t : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}^+\) at time \(t\) may be written in one of two forms:

1. Approximate (C-DF) conditional distributions \(\theta_{2i} | \theta_{2,-i}, \hat{\Theta}_1, i = 1, \ldots, p_2\) are conjugate: parameter updates using approximate transition kernel \(T\) proceed in a Gibbs-like fashion with C-DF transition kernel \(T_t\) defined as

\[
T_t(\Theta, \Theta') = \prod_{i=1}^{p_1} \pi_t(\theta'_{1i} | \hat{\Theta}_{1,t-1}, \theta'_{1l}, l < i, \theta_{1l}, l > i) \times \prod_{i=1}^{p_2} \pi_t(\theta'_{2i} | \hat{\Theta}_{1,t-1}, \theta'_{2l}, l < i, \theta_{2l}, l > i) \tag{2.9}
\]

2. Some (or all) of the approximate (C-DF) conditional distributions \(\theta_{2i} | \theta_{2,-i}, \hat{\Theta}_1, i = 1, \ldots, p_2\) are non-conjugate. Then \(\Theta_2\) is updated using a Metropolis-Hastings step with kernel \(Q(\Theta_2, \Theta'_2 | \hat{\Theta}_{1,t-1})\) with C-DF transition kernel \(T_t\) defined as

\[
T_t(\Theta, \Theta') = \prod_{i=1}^{p_1} \pi_t(\theta'_{1i} | \hat{\Theta}_{1,t-1}, \theta'_{1l}, l < i, \theta_{1l}, l > i) \times Q(\Theta_2, \Theta'_2 | \hat{\Theta}_{1,t-1}) \tag{2.10}
\]

Lemma 2.4 specifies the unique stationary distribution \(f_t : \mathbb{R}^p \rightarrow \mathbb{R}^+\) of \(T_t\).

**Lemma 2.4.** C-DF approximate kernel \(T_t\) in (2.9) and (2.10) have unique stationary distribution \(f_t(\Theta) = \pi_t(\Theta_1 | \hat{\Theta}_{2,t-1}) \pi_t(\Theta_2 | \hat{\Theta}_{1,t-1})\), where \(\pi_t(\Theta_1 | \cdot)\) and \(\pi_t(\Theta_2 | \cdot)\) are the stationary distributions for the respective parameter full conditionals.

**Remark 2.5.** Illustrations in Section 2.4 fall into the first scenario where all model parameters have conjugate full conditionals that admits surrogate quantities (SCSS). The dynamic linear model presented in Section 2.5.1 is an example where full conditional distributions admit SCSS, although some require sampling via Metropolis-Hastings. For all other examples presented in Section 2.5, the C-DF algorithm is modified to accommodate situations in which (i) the parameter space is increasing...
over time; or (ii) some (or all) of the conditional distributions fail to admit SCSS.

Though asymptotic guarantees on samples drawn are not established by Theorem 2.6 for such cases, C-DF proves its versatility by producing excellent inferential and predictive performance in all the examples considered (see Sections 2.4 and 2.5).

Main convergence result

Let \( \pi_0 \) denote the initial distribution from which parameters are drawn.

**Theorem 2.6.** Assume (i) there exists an \( \alpha \) s.t. for all \( t \), \( \sup_{\Theta} d_{TV}(T_t(\Theta, \cdot); f_t) \leq 2\alpha_t \); (ii) \( d_{TV}(f_t, f_{t-1}) \to 0 \); and (iii) \( d_{TV}(f_t, \pi_t) \to 0 \). Then, there exists \( \{n_t\}_{t \geq 1} \) s.t. \( d_{TV}(T_t^{(n_t)} \cdots T_1^{(n_1)} \pi_0, \pi_t) \to 0 \).

**Remark 2.7.** In essence, Theorem 2.6 states that running a Markov chain with approximate kernel \( T_t \) for \( n_t \) iterations at each time point \( t \) will asymptotically have draws from the true joint posterior distribution \( t \to \infty \).

**Remark 2.8.** Condition (i) in Theorem 2.6 is referred to as the “universal ergodicity condition” (Yang and Dunson, 2013), wherein they show that the universal ergodicity condition is weaker than uniform ergodicity condition on the transition kernel \( T \). Condition (ii) ensures that the stationary distribution of the approximating kernel changes slowly as time proceeds. Lemma 3.7 in (Yang and Dunson, 2013) shows that condition (ii) is satisfied for any regular parametric model by applying a Bernstein-Von Mises theorem. Finally, condition (iii) requires that the stationary distribution of the approximating kernel becomes ‘close’ to the true posterior distribution at later time. Sufficient conditions under which (iii) holds is outlined in Lemma 2.10.

**Definition 2.9.** A posterior \( \Pi(\cdot|D^{(t)}) \) is defined to be consistent at \( \Theta^0 \) if, for every neighborhood \( B \) of \( \Theta^0 \), \( \Pi(B|D^{(t)}) \to 1 \) under the true data generating law at \( \Theta^0 \).

**Lemma 2.10.** Assume that the likelihood function \( p_\Theta(\cdot) \) is continuous as a function of \( \Theta \) at \( \Theta^0 = (\Theta^0_1, \Theta^0_2) \) and \( \sqrt{t}p_\Theta(D^{(t)}) \) in limit is bounded away from 0 and \( \infty \).
Suppose $\Theta^0$ is an interior point in the domain and prior distribution $\pi_0(\Theta_1, \Theta_2)$ is positive and continuous at $\Theta^0$. Further, assume $\hat{\Theta}_{1,t} \rightarrow \Theta_1^0$, $\hat{\Theta}_{2,t} \rightarrow \Theta_2^0$ a.s. under the data generating law at $\Theta_0$, and $f_t$ and $\pi_t$ are both consistent at $\Theta^0$. Then
\[
\int |\pi_t(\Theta) - f_t(\Theta)| \, d\Theta \rightarrow 0 \text{ as } t \rightarrow \infty
\]
almost surely under the true data generating model at $\Theta^0$.

Remark 2.11. Lemma 2.10 states that if the likelihood under the data generating model grows at a certain rate (satisfied under standard regularity conditions) and estimates $\hat{\Theta}_{1,t}, \hat{\Theta}_{2,t}$ are consistent estimators of true parameters, the stationary distribution of C-DF approaches the stationary distribution of the Gibbs sampler, thus satisfying condition (iii) of Theorem 2.6. We note that our primary focus is on static models where it is generally easy to obtain a consistent sequence of estimates for the model parameters (see Section 2.4).

2.7 C-DF for online compressed regression

We consider an application to compressed linear regression where $y_1, \ldots, y_t \in \mathbb{R}^n$ are a sequence of $n$-dimensional response vectors with associated predictors $X_1, \ldots, X_t \in \mathbb{R}^{n \times p}$ observed over time. Data are modeled according to
\[
y_t|\Phi, \beta, \sigma^2 \sim N(X_t \Phi' \beta, \sigma^2 I_n)
\] (2.11)
for an $m \times p$ projection matrix $\Phi$ with $m \ll p$. A Bayesian analysis proceeds by sampling from posterior $[\beta, \Phi, \sigma^2|D^{(t)}]$ using the following default prior specification: $\beta \sim N(0, \sigma^2 \Sigma_{\beta})$; $\sigma^2 \sim \text{IG}(a,b)$; $\Phi \sim \text{MN}(\Phi_0, K, \text{1}_m)$ centered on a row ortho-normalized random projection matrix, $\Phi_0$, with row-specific scaling $\kappa_i \sim \text{IG}(1/2, 1/2)$, $1 \leq i \leq m$, and $K = \text{diag}(\kappa_1, \ldots, \kappa_m)$. Note that $\Phi' \beta = \Phi' LL' \beta$ for any orthogonal matrix $L$, hence $\Phi$ and $\beta$ are only identified up to an orthogonal
transformation. Nevertheless, regression coefficients $\gamma = \Phi'\beta$ are identifiable, and valid inference is obtained using posterior draws of the associated model parameters.

2.7.1 Online inference and competitors

Conditional on $\Phi$, the posterior distribution over $\beta, \sigma^2$ factorizes as

$$
\beta | \Phi, D^{(t)} \sim T_n(\mu_t, \Sigma_t) \quad \sigma^2 | \Phi, D^{(t)} \sim IG(a_{1,t}/2, b_{1,t}/2)
$$

$$
\Sigma_t = b_{1,t}/n W^{-1} \quad a_{1,t} = nt
$$

$$
\mu_t = (b_{1,t}/n)^{-1} \Sigma_t \Phi F_t^{XY} \quad b_{1,t} = F_t^{yy} - F_t^{XY} \Phi' W^{-1} \Phi F_t^{XY'},
$$

where $T_\nu(\cdot)$ is the multivariate t-distribution with $\nu$ degrees of freedom, with hyperparameters defined in terms of sufficient statistics $F_t^{yy}, F_t^{XY} = F_{t-1}^{yy} + y'_t y_t$, $F_t^{XX} = F_{t-1}^{XX} + X'_t X_t$ and $W = \Phi F_t^{XX} \Phi' + \Sigma_\beta^{-1}$. Sampling for $\Phi = [\Phi_1, \Phi_2, \ldots, \Phi_p]$ proceeds by drawing successively from the set of columnspecific full conditionals, namely

$$
\Phi_j|\{\Phi\}_{-j}, K, \beta, D^{(t)} \sim N_m(\mu_{\Phi_j} \Sigma_{\Phi_j}), \quad \kappa_i|\Phi \sim IG(c_i/2, d_i/2)
$$

$$
\Sigma_{\Phi_j} = \left(\sum_{s=1}^t \beta X'_{js} X_{js} \beta'/\sigma^2 + K^{-1}\right)^{-1} \quad c_i = c + p
$$

$$
\mu_{\Phi_j} = \Sigma_{\Phi_j} \left(\sum_{s=1}^t \beta X'_{js} z_{js} \sigma_z^2 + K^{-1} \Phi_{0j}\right) \quad d_i = d + (\Phi^{(i, \cdot)} - \Phi_0^{(i, \cdot)})'(\Phi^{(i, \cdot)} - \Phi_0^{(i, \cdot)}).
$$

For column $j$, $z_{jt} = y_t - \sum_{l \neq j} X_{lt} \Phi_l / \beta$, with $X_{lt}$ denoting the $l$-th column of $X_t$, and $\Phi^{(i, \cdot)}$ represents the $i$-th row of $\Phi$. The SMCMC sampling scheme for model (2.11) propagates sufficient statistics $F_t^{yy}, F_t^{XY}$ and $F_t^{XX}$ after observing $\{X_t, y_t\}$ at time $t$ and draws sequentially from full conditional distributions (2.12) and (2.13). Due to high autocorrelation between $\beta$ and $\Phi$, however, the online Gibbs sampler faces poor mixing in the joint parameter space.

The C-DF algorithm applied to this setting partitions model parameters into $\Theta_{\tilde{\nu}_1} = \{\beta, \sigma^2\}$ and $\Theta_{\tilde{\nu}_2} = \{\Phi, K\}$, and sampling proceeds as:
(1) Observe data \( D_t = \{ X_t, y_t \} \) at time \( t \). If \( t = 1 \), set \( \beta = 0, \Phi = \Phi_0, \sigma = \tilde{s}(\text{vec}(y_1, \ldots, y_k)) \), and here we assume \( y_t \) is zero-mean. Otherwise, set \( \hat{\Phi}_t \leftarrow \hat{\Phi}_{t-1}, \hat{\kappa}_{i,t} \leftarrow \hat{\kappa}_{i,t-1} \);

(2) Update surrogate quantities \( C_{1,1}^{(t)} \leftarrow C_{1,1}^{(t-1)} + \hat{\Phi}_t X'_t \bar{X}_t \hat{\Phi}'_t \) and \( C_{1,2}^{(t)} \leftarrow C_{1,2}^{(t-1)} + \Phi X'_t y_t \). Using the notation in (2.12), redefine \( W = C_{1,1}^{(t)} + \Sigma_{\beta}^{-1} \) and set \( a_{1,t} = nt, b_{1,t} = \Gamma_t^{yy} - C_{1,2}^{(t)} W^{-1} C_{1,2}^{(t)'} \);

(3) Draw \( S \) samples compositionally from \( \sigma^2|\hat{\Phi}_t, C_{1,1}^{(t)} \) and \( \beta|\sigma^2, \hat{\Phi}_t, C_{1,1}^{(t)} \). Here, a single \( m \times m \) matrix inversion is required, instead of once at every sample using SMCMC;

(4) Set \( \hat{\sigma}_t^2 \leftarrow b_{1,t}/(a_{1,t} + 1) \) and \( \hat{\beta}_t \leftarrow \mu_t \), where \( \Sigma_t = b_{1,t}/n W^{-1} \), and \( \mu_t = (b_{1,t}/n)^{-1} \Sigma_t C_{1,2}^{(t)} \) (using closed-form expressions for the posterior MAP);

(5) For \( j = 1, \ldots, p \) : update surrogate quantities \( C_{21,j}^{(t)} \leftarrow C_{21,j}^{(t-1)} + \hat{\beta}_j \hat{\beta}'_t (X'_{jt} X_{jt}) \)

and \( C_{22,j}^{(t)} \leftarrow C_{22,j}^{(t-1)} + \hat{\beta}_j X'_{jt} y_{jt} \);

(6) Draw \( S \) samples from \( \Phi, K|\hat{\beta}_t, \hat{\sigma}_t^2, C_{2}^{(t)} \) by modifying full conditionals (2.13), i.e.,

\[
[K|\Phi, -] \text{ and } [\Phi_j|\{\Phi\}_-, K, -], 1 \leq j \leq p, \text{ in terms of the defined surrogate quantities;}
\]

(a) compute \( H_{jt}(\Phi_{-j}) = C_{22,j}^{(t)} - \sum_{l \neq j} C_{21,l}^{(t)} \Phi_l \). Draw from \([\Phi_j|\{\Phi\}_{-j}, K, -] \)

with \( \Sigma_{\Phi_j} = (C_{21,j}^{(t)}/\hat{\sigma}_t^2 + K^{-1})^{-1} \) and \( \mu_{\Phi_j} = \Sigma_{\Phi_j} (H_{jt}/\hat{\sigma}_t^2 + K^{-1} \Phi_{0j}) \);

(b) draw \([K|\Phi, -] \) by sampling independently from \([\kappa_i|\Phi, -], i = 1, \ldots, m \).

(7) Set \( \hat{\Phi}_t \) and \( \{ \hat{\kappa}_{i,t} \} \) as the sample mean over these \( S \) draws.

By propagating surrogate quantities instead of the much larger sufficient statistics, the C-DF algorithm significantly reduces storage requirements while improving on
mixing efficiency and providing state-of-the-art inference (see Tables in Section 2.7.2).

Bayesian shrinkage methods such as Bayesian Lasso (Park and Casella, 2008) and GDP (Armagan et al., 2013) were attempted but rendered infeasible by the need to invert a \( p \times p \) matrix at each MCMC iteration. SMC (Doucet et al., 2001) suffers from severe particle degeneracy for learning the high-dimensional parameter \( \Phi \), and while sufficient statistics for particle rejuvenation are available, inference fairs no better than SMCMC. In addition, the need to store and propagate a large number of high-dimensional particles is completely impractical. Instead, we derive a variational Bayes (VB) approximation to the joint posterior using a GDP\(^5\) shrinkage prior on the coefficients of a standard linear regression model, \( y_i = N(x_i^\prime \beta, \sigma^2) \) and \( \beta_j | \sigma \sim GDP(\zeta = \frac{\alpha}{\alpha}, \alpha) \). The latter is equivalent to hierarchical prior \( \beta_j | \sigma, \tau_j \sim N(0, \sigma^2 \tau_j) \), with \( \tau_j \sim \text{Exp}(\lambda_j^2/2) \) and \( \lambda_j \sim \text{Ga}(\alpha, \eta) \). For \( \Theta = (\beta, \tau, \lambda, \sigma^2)^\prime \), \( \tau = (\tau_1, \ldots, \tau_p)^\prime \), and \( \lambda = (\lambda_1, \ldots, \lambda_p)^\prime \), we approximate \( \pi(\Theta|D) \) by a variational posterior with product form \( q(\Theta) = \prod_j q_j(\theta_j) \). Optimal densities \( q_j(\theta_j) \propto \exp \left[ E_{-q(\theta_j)} \{ \log \pi(\Theta, D) \} \right] \) are then obtained by minimizing the KL distance between \( \pi(\Theta|D) \) and \( q(\Theta) \), where \( E_{-q(\theta_i)} \) denotes the expectation over \( \prod_{i \neq j} q_i(\theta_i) \). The latter are well known results established in the variational Bayes literature (Braun and McAuliffe, 2010; Luts and Wand, 2013).

2.7.2 Simulation experiments

The C-DF algorithm provides robust parameter inference and predictive performance across numerous simulation experiments which consider varying degrees of sparsity as well as predictor dimension and correlation (see Tables 2.8 and A.1). Shards of \( n = 100 \) observations arrive sequentially over a \( T = 1000 \) time horizon, and predictor data are generated as \( x_i \sim N(0, R) \), \( R_{jk} = \rho^{|j-k|} \), for \( j, k = 1, 2, \ldots, p \) and

\[ \begin{align*}
\text{GDP prior has been shown to induce attractive shrinkage properties, with a carefully constructed hierarchical prior that allow Cauchy-like tails leading to better robustness (less bias due to over shrinkage) in estimating true signals, while having Laplace-like support near zero, leading to better concentration around sparse coefficient vectors.}
\end{align*} \]
correlation $\rho \in (0, 1)$. The response is generated as $y_i \sim N(\mathbf{x}_i^T \mathbf{\beta}, \sigma^2 = 4)$, where true coefficient vector $\mathbf{\beta}$ used in each case is specified in Table 2.7.

Table 2.7: Simulation experiments for supervised compressed regression. “High” signal corresponds to $\beta_j \sim U(-3, 3)$, while “low” signal corresponds to $\beta_j = 0.10$ for every nonzero feature. Sparse cases are denoted with (*).

<table>
<thead>
<tr>
<th>Case</th>
<th>$\rho$</th>
<th>$p$</th>
<th>$#\beta_j \neq 0$</th>
<th>Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>1*</td>
<td>0.1</td>
<td>500</td>
<td>10</td>
<td>high</td>
</tr>
<tr>
<td>2*</td>
<td>0.1</td>
<td>1000</td>
<td>10</td>
<td>high</td>
</tr>
<tr>
<td>3*</td>
<td>0.4</td>
<td>500</td>
<td>10</td>
<td>high</td>
</tr>
<tr>
<td>4*</td>
<td>0.4</td>
<td>1000</td>
<td>10</td>
<td>high</td>
</tr>
<tr>
<td>5</td>
<td>0.1</td>
<td>500</td>
<td>500</td>
<td>high</td>
</tr>
<tr>
<td>6</td>
<td>0.1</td>
<td>500</td>
<td>500</td>
<td>low</td>
</tr>
</tbody>
</table>

Table 2.8 reports predictive MSE for each simulation experiment averaged over 50 simulated datasets. In all cases, the results appear robust to the choice of $m$, the dimension of the subspace which $\Phi$ maps into. Choosing $m$ large can add significantly to the computational overhead of the algorithm, so we suggest setting $m = \max\{10, \log(p)\}$ by default. VB-GDP yields the lowest MSPE for sparse truths, though C-DF and SMCMC demonstrate competitive performance. Increasing the number of predictors causes MSPE to increase for all methods, as does an increase of correlation between predictors. The C-DF algorithm performs well in all cases, while VB-GDP suffers for dense truths, especially in the low-signal setting. In addition, C-DF results in excellent parameter estimation and variable selection, including cases with high predictor correlation (see Table 2.9). Table A.1 reports coverage probabilities for 95% predictive intervals for the competing methods. While C-DF and SMCMC show proper coverage, VB suffers due to the restrictive assumption of independence between parameters $a posteriori$.

Propagating surrogate quantities using the C-DF algorithm for model (2.11) re-
Table 2.8: MSPE comparisons for each simulated experiment in Table 2.7. Subscripts denote bootstrapped standard errors calculated using independent replications.

<table>
<thead>
<tr>
<th></th>
<th>Case 1*</th>
<th>Case 2*</th>
<th>Case 3*</th>
<th>Case 4*</th>
<th>Case 5</th>
<th>Case 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB</td>
<td>3.43_{0.005}</td>
<td>4.20_{0.006}</td>
<td>3.49_{0.005}</td>
<td>4.23_{0.006}</td>
<td>3.52_{0.007}</td>
<td>8.79_{0.010}</td>
</tr>
<tr>
<td>C-DF</td>
<td>3.49_{0.010}</td>
<td>4.38_{0.010}</td>
<td>3.62_{0.007}</td>
<td>4.40_{0.020}</td>
<td>3.81_{0.006}</td>
<td>3.64_{0.020}</td>
</tr>
<tr>
<td>SMCMC</td>
<td>3.56_{0.020}</td>
<td>4.40_{0.020}</td>
<td>3.58_{0.020}</td>
<td>4.43_{0.020}</td>
<td>3.68_{0.020}</td>
<td>3.70_{0.020}</td>
</tr>
</tbody>
</table>

Table 2.9: Performance comparison in terms of relative parameter MSE at $t = 100, 200$. Relative MSE for $\gamma = \Phi'\beta$ is computed as $||\hat{\Phi}_t\hat{\beta}_t - \gamma_0||^2/||\gamma_0||^2$ for C-DF and SMCMC, $\gamma_0$ is $\gamma$ at the truth. For VB, we compute $||\mu^\beta_{VB} - \gamma_0||^2/||\gamma_0||^2$; where $\mu^\beta_{VB}$ is the approximate posterior mean for $\beta$. Subscripts denote bootstrapped standard errors calculated using independent replications.

<table>
<thead>
<tr>
<th>Time</th>
<th>Case 1*</th>
<th>Case 2*</th>
<th>Case 3*</th>
<th>Case 4*</th>
<th>Case 5</th>
<th>Case 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB</td>
<td>100</td>
<td>0.009_{0.001}</td>
<td>0.019_{0.001}</td>
<td>0.018_{0.002}</td>
<td>0.024_{0.002}</td>
<td>0.002_{0.001}</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.004_{0.001}</td>
<td>0.008_{0.001}</td>
<td>0.005_{0.001}</td>
<td>0.008_{0.001}</td>
<td>0.001_{0.001}</td>
</tr>
<tr>
<td>C-DF</td>
<td>100</td>
<td>0.010_{0.001}</td>
<td>0.033_{0.003}</td>
<td>0.029_{0.004}</td>
<td>0.074_{0.001}</td>
<td>0.013_{0.001}</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.004_{0.001}</td>
<td>0.011_{0.002}</td>
<td>0.011_{0.002}</td>
<td>0.027_{0.001}</td>
<td>0.003_{0.001}</td>
</tr>
<tr>
<td>SMCMC</td>
<td>100</td>
<td>0.014_{0.003}</td>
<td>0.032_{0.004}</td>
<td>0.024_{0.003}</td>
<td>0.034_{0.009}</td>
<td>0.004_{0.001}</td>
</tr>
<tr>
<td></td>
<td>200</td>
<td>0.006_{0.001}</td>
<td>0.015_{0.002}</td>
<td>0.010_{0.002}</td>
<td>0.017_{0.004}</td>
<td>0.002_{0.000}</td>
</tr>
</tbody>
</table>

Results in a dramatic efficiency gain over SMCMC. A measure of this efficiency is the “effective sample-size,” namely, the number of samples required for the predictive MSE to drop below a chosen threshold. Table A.2 reports effective sample sizes for each simulated experiments in Table 2.7. In addition, C-DF results in dramatically lower memory footprint (0.7/2.0 MB for $p = 500$; 1.3/8.1 MB for $p = 1000$).

Finally, storage of the sufficient statistics for model (2.11) in the predictor dimension, and updating these quantities is $O(np^2) + O(np)$ at each time point. In comparison, updating surrogate quantities for C-DF is $O(mpn) + O(m^2n)$. The C-DF algorithm dramatically reduces storage requirements for online inference by reducing the quadratic dependence on $p$ to a linear order.

2.8 Extensions and future work

The Conditional Density Filtering (C-DF) algorithm facilitates efficient online Bayesian inference by adapting MCMC to the online setting, with sampling based approxima-
tions to conditional posterior distributions obtained by propagating surrogate statistics as new data arrive. Across a variety of simulated and real data settings, the C-DF algorithm proves its versatility by producing excellent parameter estimation, uncertainty characterization, and predictive performance.

It remains to extend C-DF to other complex model settings. Of particular interest is mixture models that flexibly estimate the conditional density of a response given its associated predictors (Geweke and Keane, 2007; Dunson et al., 2007; Dunson and Park, 2008). Such models are commonly used to obtain reliable predictive inference or quantile estimation (e.g., tail probabilities in survival data). Here and elsewhere, hierarchical specification for the richly structured model is routinely expressed in terms of a growing set of observation-specific nuisance parameters. Though C-DF and its associated theoretical guarantee were developed in the finite parameter setup, careful modification of the C-DF algorithm using a “budgeted” window to impute a subset of these latent variables was discussed in the context of a first-order autoregressive model and in an application to probit (binary) regression. Performance is surprisingly good, however, we believe a more principled approach is necessary to bring C-DF into the fold of complex hierarchical modeling framework which predominate most real world applications.

Another avenue for future research includes exploring connections between C-DF and the approximate Bayes computation literature for intractable likelihoods (Bissiri et al., 2013), as is the case where no surrogate statistics for some (all) of the conditional distributions exist.
Stochastic neighborhood search for Bayesian variable selection

The failure of maximum likelihood estimation in the high-dimensional $p > n$ setting naturally gives rise to the variable selection task. When predictors are known to have lower dimensional structure or it is known that only a small subset of predictors are predictive of the response, exploiting such structure can lead to dramatic improvements in the statistical efficiency of the learning algorithm. Classic step-wise procedures based on likelihood ratio tests for nested models or penalized model scores (e.g., Akaike’s information criterion (AIC) or the Bayesian information criterion (BIC)) are generally unreliable in high dimensions. Modern Bayesian approaches to variable selection in the regression setting divide largely into two groups: (1) exact predictor inclusion-exclusion via spike-slab priors, and (2) continuous shrinkage priors which mimic the latter. With the number of possible models growing as $2^p$, direct enumeration of the model space is intractable for $p \geq 30$. While the former comes equipped with natural measures of uncertainty, such as the posterior probability of each visited model and marginal predictor inclusion probabilities, the latter often leads to more tractable inferential procedures in terms of posterior computation.
3.1 Introduction

Variable selection has received a tremendous amount of attention in frequentist literature, with numerous penalized approaches enjoying much success. Most well known methods, including the Lasso (Tibshirani, 1996), SCAD (Fan and Li, 2001), Cosso (Lin et al., 2006), and the Dantzig selector (Candes and Tao, 2007) are one-stage procedures, focusing on simultaneous selection and estimation of unknown model parameters; in fact, many of these come with appealing oracle properties and asymptotic guarantees. There is an equally overwhelming body of work in the Bayesian variable selection and model averaging literature dating back to Zeller’s $g$-prior (1986). Since then, a populous set of shrinkage priors have been developed along similar lines. Recent and notable among these include the Bayesian Lasso (Park and Casella, 2008), Horseshoe (Polson and Scott, 2012), Generalized Double Pareto (Armagan et al., 2013), and Dirichlet-Laplace (Bhattacharya et al., 2014). However, proponents of two-stages procedures (SIS, Fan and Lv, 2010), (VANISH, Radchenko and James, 2010), argue that simultaneous selection and estimation is often too ambitious, instead proposing efficient variable screening algorithms which promise retaining the true support in the GLM setting with high probability under regularity conditions on the design matrix. Projection pursuit regression (PPR) (Friedman and Stuetzle, 1981), Likelihood Basis Pursuit (LBP) (Zhang et al., 2004), and the Leaps and bounds algorithm (Furnival and Wilson, 1974; Hoeting et al., 1999; Brusco and Steinley, 2011) are classic approaches to selection that utilize various optimization methods including tabu search and the branch-and-bound algorithm. Relying on penalized likelihood scoring, these methods can be effective model selection tools in simple model settings but offer no uncertainty quantification.

The spike-slab approach to variable selection has been predominantly developed in the linear regression setting, largely due to analytical tractability (George and
McCulloch, 1993; Geweke, 1994; Draper, 1995; Carlin and Chib, 1995; Clyde et al., 1996; Hoeting et al., 1999). Here, analytical expressions for the marginal likelihood $p(y|\gamma)$ for prediction inclusion vector $\gamma \in \{0, 1\}^p$ enable efficient stochastic search over the model space. The MC3 algorithm (Raftery et al., 1997) and stochastic search variable selection (SSVS, George and McCulloch, 1993) are two early Markov chain samplers that enable variable selection. SSVS traverses the model space by scanning over the $p$-variates successively, allowing each predictor to have its state flipped, confining the search for important predictors to a local neighborhood of size $p$ at every MCMC iteration. Sequential scanning is conceptually simple, but tends to be slow as the predictor dimension grows and can suffer mixing problems in correlated predictor settings. Kohn et al. (2001), Dellaportas et al. (2002) and Cottet et al. (2008) provide some extensions of variable selection in the over-dispersed generalized linear model setting.

Several other stochastic search procedures have been proposed in various contexts, including applications to Gaussian graphical models and social networks (Jones et al., 2005; Scott and Carvalho, 2008), with a focus on enumerating models having high posterior probability. The authors argue that in the enormous model space, the Metropolis criterion is “less useful as an MCMC transition kernel, and far more useful as a search heuristic for finding and cataloguing good models,” and reliable computation of model probabilities based on frequency of occurrence in a Monte Carlo seems dubious. Shotgun Stochastic Search (SSS) (Hans et al., 2007) proposes a neighborhood search procedure to quickly identify inclusion vectors with large posterior mass in high dimensions, and is demonstrated to perform well in linear regression and graphical model setting of moderate dimension. Clyde et al. (1996) and Clyde and George (2004) discuss various Bayesian variable selection strategies for model averaging, taking advantage of specific model structure such as orthogonalized design to obtain closed-form posterior model probabilities. In addition, Clyde et al. (2011)
propose Bayesian adaptive sampling (BAS) to sequentially learn marginal inclusion probabilities using a without replacement sampling algorithm. BAS improves best over the baseline MCMC competitors when the algorithm has access to warm starts for the marginal predictor inclusion probabilities. Berger and Molina (2005) propose a stochastic search algorithm that incorporates local proposals which explore a neighborhood around a catalogue of previously sampled models using initial estimates for posterior model and predictor inclusion probabilities to guide the traversal. Using a path-sampling approach to efficiently compute Bayes factors between one-away pairs of models in the linear regression setting, their strategy yields a connected graph between all explored models with the hope that this set is large enough to reliably estimate approximate inferential statistics.

3.2 Basic notation and background

In the absence of prior information about predictor importance, it is natural to treat predictors as a priori exchangeable with common prior inclusion probability $\tau \in (0, 1)$ and i.i.d univariate conditional distributions

$$\beta_j | \gamma_j \sim (1 - \gamma_j) F_0(\beta_j) + \gamma_j F(\beta_j) \quad \gamma_j | \tau \sim \text{Bernoulli}(\tau) \quad (3.1)$$

based on the $p$-dimensional inclusion state $\gamma = (\gamma_1, \ldots, \gamma_p)$, allowing the state-specific marginals $\beta_j \sim F_{\gamma_j}, \gamma_j \in \{0, 1\}$ to be chosen in a variety of ways. Stochastic search variable selection (SSVS, George and McCulloch, 1993) and Gibbs variable selection (GVS, Dellaportas et al., 2002) typically place continuous distributions on both components, though a more common choice (and the one pursued here) is the degenerate spike-slab prior for $\gamma_j = 0$ component, $F_0(\cdot)$. Variable selection in the linear regression setting models response data as $y|\gamma, \beta, \sigma^2 \sim N(X\gamma, \beta, \sigma^2 I_n)$ where $\beta_\gamma := \{\beta_j : \gamma_j = 1; 1 \leq j \leq p\}$. George and Foster (2000) discuss an empirical Bayes approach for choosing prior inclusion probability $\hat{\tau}$ and scaling $\hat{c}$ for a $g$-prior
\( \beta_\gamma \sim N(0, c(X_\gamma X_\gamma)^{-1}) \) on the non-zero regression coefficients. An alternative is to model \( \tau \sim \text{Beta}(\omega, \nu) \); see Section 3.4.1.

3.3 Scaling MCMC variable selection

3.3.1 Weighted toggle proposals for predictor inclusion

Liu (1996) showed that incorporating a Metropolis-within-Gibbs step to replace the usual conjugate SSVS predictor update is superior in the Peskun sense of having smaller asymptotic variance for functionals based on MCMC draws from the latter; see Peskun (1973). However, it is well known that one-at-a-time schemes for updating predictor inclusion can face mixing issues in the presence of correlated predictors, a commonplace occurrence in high dimensional applications.

One often considers multiple types of proposal moves to balance between local and global exploration more effectively. Green (2003) remarks “In a typical application with multiple parameter subspaces \( \{\zeta_k\} \) of different dimension, it will be necessary to devise different types of moves between the subspaces. These will be combined to form what Tierney (1994) calls a hybrid sampler, by random choice between available moves at each transition, in order to traverse freely across the combined parameter space \( \zeta \).” In addition, proposals that change the inclusion status of a randomly chosen variable index are inefficient even when \( p \) is moderately large. When the assumed prior concentrates on sparse inclusion vectors, such random walks are heavily biased toward adding additional predictors rather than removing unwanted ones. This is solved by considering a transition kernel that proposes add/remove/swap changes to inclusion state \( \gamma \) with probabilities \( (|\gamma| + 1)^{-1}, (p - |\gamma| + 1)^{-1} \) and \( w_S(|\gamma|) = 1 - \frac{1}{|\gamma|+1} - \frac{1}{p-|\gamma|+1} \), which is positive if \( |\gamma|(p - |\gamma|) > 1 \). A random-walk on predictor changes given the selection move type is as follows: If ‘add’, select one of the \( p - |\gamma| \) predictors not currently in \( \gamma \) to include; if ‘remove’, select a predictor from \( |\gamma| \) to
remove; otherwise, select a single predictor from each disjoint set and switch their inclusion state. In particular,

$$T(\gamma, \gamma') = \begin{cases} 
\frac{1}{(p+1)|\gamma|} & \text{if } \gamma' = \gamma + 1_j, \text{ for some } j \in [\gamma]^c \\
\frac{1}{(p-|\gamma|+1)|\gamma|} & \text{if } \gamma' = \gamma - 1_j, \text{ for some } j \in [\gamma] \\
\frac{|w_s|}{|\gamma|} & \text{if } \gamma' = \gamma + 1_j - 1_k, \text{ for some } j \in [\gamma], k \in [\gamma]^c \\
0 & \text{otherwise.}
\end{cases}$$

Since $T$ is symmetric, proposals are accepted with probability $\min\{1, \pi(\gamma')/\pi(\gamma)\}$ leading to a Markov chain $(\gamma(t) : t \geq 1)$ with $\pi$ as its stationary distribution.

The inefficiency of random selection has been addressed in various MCMC schemes developed in the context of exploring high dimensional discrete spaces (Kwon et al., 2011; Kohn et al., 2001). The Kohn-Smith-Chan algorithm proposals operate on the premise that there are a large number of useless predictors, so proposed changes should be light-weight, and avoid unnecessarily likelihood evaluations. Block updating components (analogous to swap transitions) may also be considered, but requires an efficient strategy to choose predictor subsets adaptively and without adding significantly to the per iteration computational overhead. Attempts to incorporate more complex global moves in high dimensional problems have also been considered in applications to statistical physics and genomics (Hamze et al., 2013; Bottolo et al., 2010; Mansinghka et al., 2009; Strens, 2003; Liang and Wong, 2001). Adaptive sampling schemes have also been developed (Nott and Kohn, 2005; Ji and Schmidler, 2013). Finally, the possibility of using structured priors to enforce group sparsity for highly correlated predictors as discussed in Section 3.3.2 is a relatively unexplored area of research.
3.3.2 The Swendsen-Wang prior

An alternative to the standard single-site Metropolis updating schemes (George and McCulloch, 1993; Hoeting et al., 1999), the Swendsen-Wang algorithm is often used as a means to decouple interactions between predictors and enabling better mixing in stochastic search variable selection procedures. In the Ising model, $\gamma$ represents a binary spatial process having joint distribution

$$
\pi(\gamma) \propto \exp \left( \sum_{i=1}^{n} \alpha_i(\gamma) + \sum_{j<k} \phi_{jk} I(\gamma_j = \gamma_k) \right)
$$

(3.2)

where $\phi_{jk} \geq 0$ is a user specified “potential function,” and auxiliary variables $u_{jk}$, $1 \leq j, k \leq p$ are introduced having uniform conditional distribution

$$
\pi(u_{jk}|\gamma) = \exp(-\phi_{jk} I(\gamma_j = \gamma_k)) I(u_{jk} \in [0, \exp(\phi_{jk} I(\gamma_j = \gamma_k))]).
$$

(3.3)

This specifies a valid joint distribution $\pi(\gamma, u)$, chosen specifically so that sampling from conditional distribution $\pi(\gamma|u) \propto \pi(\gamma, u)$ is easy. A consequence of this parameterization is that it constrains $\gamma_j = \gamma_k$ whenever $u_{jk} \in [1, \exp \phi_{jk}]$, resulting in clusters of “sites” (predictors) having the same value.

In the context of constructing a general purpose stochastic search variable selection algorithm, MCMC proceeds by drawing from the pair of full conditional distributions $\pi(u|\gamma, y)$ and $\pi(\gamma|u, y) \propto \pi(u|\gamma, y) \pi(\gamma|y)$. In particular,

$$
\pi(\gamma|u, y) \propto \frac{\pi(\gamma|y)}{\exp \left( \sum_{j,k} \phi_{jk} I(\gamma_j = \gamma_k) \right)} I(u_{jk} \in [0, \exp(\phi_{jk} I(\gamma_j = \gamma_k))], 1 \leq j, k \leq p).
$$

(3.4)

Here, one has the freedom to specify $\pi(u|\gamma, y)$, often chosen by default as (3.3). In addition, a way of choosing tuning parameters $\{\phi_{jk}\}$ is needed, noting that $\phi_{jk} < 0 \iff \gamma_j \neq \gamma_k$ and $\phi_{jk} > 0 \iff \gamma_j = \gamma_k$. Nott and Green (2004) provide an overview of how to choose these tuning parameters with application to a linear regression setting.
3.4 A framework for robust SSVS in high dimensions

Acknowledging the tension between local efficiency and mode-finding while maintaining MCMC reversibility, we adapt concepts from stochastic neighborhood search Hans et al. (2007) and generalize the multiple-try Metropolis (Liu et al., 2000) algorithm for efficient sampling of inclusion vectors. Key innovations address the challenges faced by variable selection sampling in high dimensions; in particular, a scalable MCMC sampler should

1. **effectively tradeoff exploration and exploitation**: in high dimensions, MCMC should employ a mechanism for adaptation as means of efficiently discovering regions of high posterior probability;

2. **have an efficient transitioning scheme**: poor mixing can result when good proposals are identified but rejected because of the reversibility constraint; and

3. **cut the computational budget**: when likelihood evaluations are expensive, rejections are wasteful. A flexible sampling scheme will allow for a host of proposals, allowing for an annealing process toward modal regions of the posterior.

### 3.4.1 Sparse priors for inclusion vectors

The standard product bernoulli prior conditional on common prior inclusion probability $\tau \in (0, 1)$ leads to an exchangeable prior over prediction inclusion vector $\gamma = (\gamma_1, \ldots, \gamma_p)$, namely

$$
\pi(\gamma | \tau) = \prod_{j=1}^{p} \tau^{\gamma_j} (1 - \tau)^{1-\gamma_j}.
$$

(3.5)

Fixing $\tau = 1/2$ leads to a distribution over model sizes depicted in Figure 3.1(b) having an average model size of $p/2$, which clearly does not favor sparse configurations for growing $p$. The “uniform indifference” prior $\pi(\gamma) = 2^{-p}$, $\gamma \in \{0, 1\}^p$ likewise fails to adequately penalize large models.
If $\tau$ is treated as a model parameter to be estimated from the data, the induced prior over model size has automatic adjustment properties for multiple testing, yielding strong control over the number of false predictors admitted into the model even as the number of spurious tests grows without bound; see Scott et al. (2010). We therefore model $\tau \sim \text{Beta}(\omega, \nu)$ parametrized by its prior mean and “sample size,” with $\omega = d^*/p$ and $\nu = p$, for a hyper-parameter $d^*$ on the expected model size. The standard Beta distribution parameterization is given by $\alpha = \omega \nu$, $\beta = (1 - \omega) \nu$.

When $p$ is large, $\text{var}(\tau) \approx 1/p^2$ and $\tau$ concentrates near its prior mean; see Figure 3.1(a). The induced marginal Beta-binomial distribution $\pi(|\gamma|; d^*)$ shown in Figure 3.1(c) places most of its mass on small models, with alternate choices of $d^*$ allowing for additional flexibility. In addition, closed form prior model probabilities may be computed as

$$\pi(\gamma; d^*) = \frac{B(\alpha + |\gamma|, \beta + p - |\gamma|)}{B(\alpha, \beta)} \quad (3.6)$$

where $B(\cdot, \cdot)$ is the normalizing constant for Beta distribution.

### 3.4.2 Reversible neighborhood sampling

We consider the following single-predictor changes to component inclusion vector $\gamma$, namely

1. Add neighbors: $N_A(\gamma) = \{\gamma' : \gamma = \gamma + 1_j, \ j \in [\gamma]^c\}$
2. Remove neighbors: $N_R(\gamma) = \{\gamma' : \gamma = \gamma - 1_j, \ j \in \gamma\}$
3. Swap neighbors: $N_S(\gamma) = \{\gamma' : \gamma = \gamma - 1_j + 1_k, \ (j, k) \in \gamma \times [\gamma]^c\}$.

Swap moves are equivalent to adding and then removing a (different) predictor, but doing so in one step. A proposal distribution

$$p(\gamma' | \gamma) = \frac{h(\gamma' | \gamma, -)}{\sum_{\tilde{\gamma} \in N(\gamma)} h(\tilde{\gamma} | \gamma, -)}, \quad \gamma' \in N(\gamma) = N_A(\gamma) \cup N_R(\gamma) \cup N_S(\gamma) \quad (3.7)$$
for a nonnegative and bounded score function $h(\cdot|\gamma, -)$ defined over the “one-away” neighbors may be used in a Markov chain sampler over predictor inclusion. A Metropolis step is completed by accepting proposals from (3.7) with probability

$$
\alpha(\gamma, \gamma') = \min \left\{ 1, \frac{\pi(\gamma'|\gamma') h(\gamma|\gamma', -) \sum_{\tilde{\gamma} \in N(\gamma)} h(\tilde{\gamma}|\gamma, -)}{\pi(\gamma|\gamma') h(\gamma'|\gamma', -) \sum_{\tilde{\gamma} \in N(\gamma')} h(\tilde{\gamma}|\gamma', -)} \right\}
$$

This leads to a Markov chain over states that is irreducible, aperiodic, and positive recurrent. It is easy to verify that transition kernel $T(\gamma \rightarrow \gamma') = p(\gamma'|\gamma) \alpha(\gamma, \gamma')$, it is easy to see that $\pi(\gamma|-) T(\gamma \rightarrow \gamma')$ satisfies detailed balance. The induced $\pi$-reversible Markov chain is referred to as reversible neighborhood sampler (RNS). RNS allows every predictor to be considered in the context of the current inclusion vector $\gamma$ and is biased towards configurations with large posterior probability.

Score function (3.7) used to construct the RNS proposal distribution is general, however we focus on the special case where $h(\tilde{\gamma}|\gamma, -) \propto \pi(\tilde{\gamma}|\gamma)$, the unnormalized posterior probability for $\tilde{\gamma} \in N(\gamma)$. The resulting per-iteration complexity for sampling component inclusion vectors scales linearly in the predictor dimension, with $|\gamma|$
remove, \( p-|\gamma| \) add, and \(|\gamma|(|p-|\gamma|)\) swap neighbors. This is cost is further exacerbated as each marginal likelihood evaluation scales at least as \( O(n) \). Although likelihood scores may be evaluated in parallel across separate processors, we note that most one-away neighbors offer little improvement to the model fit in high dimensions.

### 3.4.3 Paired-moved neighborhood search

In correlated and high dimensional settings, good inclusion configurations often have good (or better) neighbors which may cause transitions to be slow under the RNS reversibility constraint. Leveraging the disjoint structure of one-away neighborhood in terms of constituent add, remove and swap configurations, we propose a paired-move reversible neighborhood sampler (pRNS) with “Add-remove”, “Remove-add”, and “Swap-swap” forward-reverse neighborhoods. By allowing different moves to be proposed separately and in quick succession, pRNS can dramatically improve mixing in the space of single predictor changes to \( \gamma \). The pRNS proposal distribution is given by

\[
p(\gamma'|\gamma) = w_A p_A(\gamma'|\gamma) + w_R p_R(\gamma'|\gamma) + (1 - w_A - w_R) p_S(\gamma'|\gamma) \tag{3.8}
\]

with proposal distributions \( p_A, p_R, p_S \) as in (3.7) restricted to their respective one-away neighborhoods and \( 0 \leq w_A, w_R \leq 1 \). These probabilities are allowed to vary with inclusion size to encourage additions to smaller inclusion states, removal from larger inclusion states, and swaps for intermediately sized ones. We specify \( w_A(|\gamma|) \) to be monotonically decreasing in \(|\gamma|\) with \( w_A(0) = 1, w_A(p) = 0 \) and \( w_S(|\gamma|) > 0 \) whenever \(|\gamma|(|p-|\gamma|)| > 0 \). In addition, we recommend having \( w_S(|\gamma|) \) be unimodal at \( \bar{d} = \arg\min \Pr(|\gamma| > s; d^*) \leq 0.05 \) for a configuration size \( s > d^* \). The transition rule for sampling inclusion vectors under pRNS is as follows:

1. Select move \( m \in \{A, R, S\} \) with probability \( w_A, w_R, \) and \( w_S \), respectively

2. For move \( m \in \{A, R, S\} \), construct forward one-away neighborhood \( N_m(\gamma) \)
3. Propose $\gamma' \in N_m(\gamma)$ sampled according (3.8) restricted to this set

4. Construct reverse neighborhood $N_{m'}(\gamma')$ for $m' \in \{R, A, S\}$ (note: $\gamma \in N_{m'}(\gamma')$

5. Accept proposal $\gamma'$ with probability

$$
\alpha_m(\gamma, \gamma') = \min \left\{ 1, \frac{w_{m'}(|\gamma'|) \sum_{\tilde{\gamma} \in N_m(\gamma)} \pi(\tilde{\gamma}|-) \pi(\tilde{\gamma})}{w_m(|\gamma|) \sum_{\tilde{\gamma} \in N_{m'}(\gamma')} \pi(\tilde{\gamma}) \pi(\tilde{\gamma})} \right\}. \quad (3.9)
$$

**Lemma 3.1.** The paired-move neighborhood sampler (pRNS) with acceptance probability (3.9) satisfies detailed balance and samples have the desired target distribution $\pi(\gamma)$.

**Proof:** See Appendix B.

**Remark 3.2.** Recall that score function $h(\tilde{\gamma}|\gamma, -)$ need only be bounded and nonnegative. Proceeding sections take consider $h(\tilde{\gamma}|\gamma, -)$ proportional to the unnormalized posterior probability $\pi(\tilde{\gamma})$.

**Remark 3.3.** pRNS effectively reduces the number of likelihood evaluations per iteration for moderate $p$, though Add and Swap neighborhoods remain $O(p)$. As the predictor dimension $p$ grows, an additional mechanism for limiting the neighborhood size is essential (especially as evaluating score function $h(\cdot)$ may be costly in general).

3.4.4 A multiple-try Metropolis algorithm for inclusion

For a real valued random variable $x \in \mathcal{X} \subseteq \mathbb{R}$, let $T : \mathcal{X} \times \mathcal{X} \to [0, 1]$ be a transition kernel such that $T(x, x') > 0 \iff T(x', x) > 0$. In addition, let $\lambda(x, x')$ be some symmetric function and define weight function $\omega(x, x') = \pi(x) T(x, x') \lambda(x, x')$.

The Multiple-try Metropolis algorithm (MTM, Liu et al., 2000) produces draws from target distribution $\pi(x)$ by (1) sampling reference set $x_1^*, \ldots, x_M^* \sim T(x, \cdot)$; (2) calculating weights $\omega(x_i^*, x)$, $i = 1, \ldots, M$; (3) drawing $x' \in \{x_1^*, \ldots, x_M^*\}$ with probability proportional to $\omega(x', x)$; (4) sampling reverse set $\tilde{x}_1, \ldots, \tilde{x}_{M-1} \sim T(x', \cdot)$ having
weights $w(\tilde{x}_i, x')$, $i = 1, \ldots, M - 1$ and $\tilde{x}_M = x$; and (5) accepting proposal $x'$ w.p. $\alpha(x, x') = \min\left\{1, \frac{\sum_{i=1}^{M} w(x_i, x')}{\sum_{i=1}^{M} w(x_i, x')}\right\}$. The standard Metropolis-Hasting sampler results as a special case, taking $\lambda(x, x') = 1$ and $M = 1$. Pandolfi et al. (2010) extend this approach by considering RNS proposal distribution (3.7), along with $T(x, \cdot)$ which generates sets of candidate moves as before.

We adapt MTM to the discrete setting where transitions are confined to the one-away neighbors of inclusion vector $\gamma$. Reversibility is maintained and we propose a general framework for generating stochastic sets of “one-away” inclusion neighborhoods. Define $\eta_j \sim \text{Bern}(\omega_j)$ with weight function $\omega : \{0,1\} \times \mathbb{R}^+ \rightarrow [0,1]$ taking input $\gamma_j = I(j \in \gamma)$ and nonnegative predictor importance score $v_j$, $1 \leq j \leq p$. Let $\text{tog}(j, \gamma) : \gamma_j \leftarrow 1 - \gamma_j$, where predictor $j$ is “toggled” on/off corresponding to an Add(Remove) whenever $\gamma_j = 0$ and $\eta_j = 1$. The unique set of selected predictor toggles ensures no duplicates among the constructed neighborhood sets. We initialize $\gamma = (0, \ldots, 0)$ and set importance weights $v_j = 1$, $1 \leq j \leq p$ under an assumed prior predictor exchangeability.

For clarity, we focus attention on the construction of mixed one-away neighborhoods under Add/Remove toggles (n.b. Swap proposals are incorporated by considering $\tilde{\gamma} = \text{tog}((r,a), \gamma)$ for predictor pairs $(r,a) \in \{(j,k) : \gamma_j = 1, \gamma_k = 0\}$ subject to $\eta_r = \eta_a = 1$). The discrete MTM (dMTM) algorithm proceeds as follows:

1. For $j = 1, \ldots, p$, draw $\eta_j \sim \text{Bern}(\omega(\gamma_j, v_j))$ independently

2. For $k \in \{j : \eta_j = 1\}$, $\tilde{\gamma} = \text{tog}(k, \gamma)$ defines the forward set of mixed Add and Remove neighbors

3. Among the constructed forward neighbors, select proposal $\gamma' = \text{tog}(k^*, \gamma)$ with probability $\pi(\gamma')/\sum_{j: \eta_j = 1} \pi(\text{tog}(j, \gamma))$

4. For $j \neq k^*$, draw $\eta_j' \sim \text{Bern}(\omega(\gamma'_j, v_j))$ independently and set $\eta_{k^*}' = 1$. For
\( k \in \{ j : \eta'_j = 1 \} \), \( \tilde{\gamma} = \text{tog}(k, \gamma') \) defines the reverse set of mixed Add and Remove neighbors.

5. Accept proposal \( \gamma' \) with probability

\[
\alpha(\gamma, \gamma') = \min \left\{ 1, \frac{\omega(\gamma'_k, v_k)}{\omega(\gamma_k^*, v_k^*)} \sum_{j : \eta_j = 1} \pi(\text{tog}(j, \gamma)) \right\}. \quad (3.10)
\]

**Lemma 3.4.** The discrete MTM (dMTM) algorithm with acceptance probability (3.10) satisfies detailed balance and samples have the desired target distribution \( \pi(\gamma) \).

**Proof:** Follows from the proof of Lemma 3.5 in Appendix B.

### 3.4.5 A paired-move discrete MTM sampler

A paired-move multiple-try neighborhood sampler (pMTM) is obtained as a special case of the dMTM algorithm for a specific choice of weight function \( \omega \), namely

\[
\omega(\gamma_j, v_j; m) = f(v_j)I(m = A, \gamma_j = 0) + g(v_j)I(m = R, \gamma_j = 1) \quad (3.11)
\]

with choice of functions \( f, g : \mathbb{R}_+ \to [0, 1] \) controlling the inclusion probability for predictor \( \gamma_j \in N(\gamma; m) \) on move \( m \in \{ A, R, S \} \). The pMTM algorithm is given as:

1. Select move \( m \in \{ A, R, S \} \) with probability \( w_A(|\gamma|), w_R(|\gamma|), \) and \( w_S(|\gamma|), \) respectively.

2. (a) If move \( m \in \{ A, R \} \) : for \( j = 1, \ldots, p \), draw \( \eta_j \sim \text{Bern}(\omega(\gamma_j, v_j; m)) \). For \( k \in \{ j : \eta_j = 1 \} \), \( \tilde{\gamma} = \text{tog}(k, \gamma) \) defines the set of Add/Remove neighbors.

   (b) If move \( m = S \) : define \( (\eta_r, \eta_a) \in \{ 0, 1 \}^2 \) and consider predictor pairs \( (r, a) \in \{ (j, k) : \gamma_j = 1, \gamma_k = 0 \} \). For \( r \in \gamma \), draw \( \eta_r \sim \text{Bern}(\omega(\gamma_r, v_r, m = R)) \), and independently for \( a \in [\gamma]^c \) draw \( \eta_a \sim \text{Bern}(\omega(\gamma_a, v_a, m = A)) \). For \( (r, a) : \eta_r = \eta_a = 1, \tilde{\gamma} = \text{tog}((r, a), \gamma) \) defines the set of Swap neighbors.
3. Among the constructed forward neighbors, select proposal $\gamma' \in N_m(\gamma)$ with probability $\pi(\gamma')/\sum_{\tilde{\gamma} \in N_m(\gamma)} \pi(\tilde{\gamma})$. If $m \in \{A, R\}$, denote $\gamma' = \text{tog}(k^*, \gamma)$; otherwise denote $\gamma' = \text{tog}((r^*, a^*), \gamma)$ for $m = S$

4. (a) If move $m = A$ : for $j \neq k^*$, draw $\eta_j' \sim \text{Bern}(\omega(\gamma'_j, v_j, m' = R))$ and set $\eta_{k^*}' = 1$. Using the defined weight function, note that for $j \notin \gamma'$, $\omega_j = 0 \implies \eta_j' = 0$. For $k \in \{j : \eta_j' = 1\}$, $\tilde{\gamma} = \text{tog}(j, \gamma')$ defines the set of reverse (remove) neighbors

(b) If move $m = R$ : for $j \neq k^*$, draw $\eta_j' \sim \text{Bern}(\omega(\gamma'_j, v_j, m' = A))$ and set $\eta_{k^*}' = 1$. Using the defined weight function, note that for $j \in \gamma'$, $\omega_j = 0 \implies \eta_j' = 0$. For $k \in \{j : \eta_j' = 1\}$, $\tilde{\gamma} = \text{tog}(j, \gamma')$ defines the set of reverse (add) neighbors

(c) If move $m = S$ : define $(\eta_r', \eta_a') \in \{0, 1\}^2$ as before, with predictor pairs $(r, a) \in S(\gamma') = \{(j, k) : \gamma'_j = 1, \gamma'_k = 0\}$. For $(r, a) \in S(\gamma') : (r, a) \neq (a^*, r^*)$, draw $\eta_r' \sim \text{Bern}(\omega(\gamma'_r, v_r, m' = R))$ and independently draw $\eta_a' \sim \text{Bern}(\omega(\gamma'_a, v_a, m' = A))$, and set $(\eta_r'^*, \eta_a'^*) = (1, 1)$. For $(r, a) : \eta_r' = \eta_a' = 1$, $\tilde{\gamma} = \text{tog}((r, a), \gamma')$ defines the set of reverse (swap) neighbors

5. For $m \in \{A, R, S\}$, the corresponding reverse paired neighborhood is $m'^* \in \{R, A, S\}$. Let $s^* = k^*$ if $m \in \{A, R\}$; otherwise $s^* = (r^*, a^*)$ for $m = S$. With forward and reverse paired neighborhoods $N_m$ and $N_{m'}$ as defined above, accept proposal $\gamma' = \text{tog}(s^*, \gamma)$ with probability

$$\alpha_m(\gamma, \gamma') = \min \left\{ 1, \frac{w_{m'}(|\gamma'|)}{w_{m}(|\gamma|)} \frac{\omega(s^*, v; m') \sum_{\tilde{\gamma} \in N_m(\gamma)} \pi(\tilde{\gamma})}{\omega(s^*, v; m) \sum_{\tilde{\gamma} \in N_{m'}(\gamma)} \pi(\tilde{\gamma})} \right\}. \quad (3.12)$$

**Lemma 3.5.** The paired MTM (pMTM) algorithm with acceptance probability (3.12) satisfies detailed balance and samples have the desired target distribution $\pi(\gamma)$.

**Proof:** See Appendix B.
3.4.6 Adaptive predictor importance

Weight function $\omega(\gamma_j, v_j; m)$ (defined for pMTM in (3.11)) provides a mechanism to improve mixing and robustness for sampling predictor inclusion vectors in both low-signal and high dimensional settings. Effective strategies will enhance the chance of selecting predictors with large $v_j$ when $j \notin \gamma$ and suppress them when $j \in \gamma$. As importance scores are updated, predictors with large $v_j$ are promoted within Add neighborhoods and demoted in remove neighborhoods.

We propose an adaptive version of the Markov chain sampler where a predictor’s chance of being included in an (Remove) Add neighborhood is (inversely) proportional to its importance score, $v_j$. We consider

$$f(v_j; M) = \frac{M v_j}{M v_j + p}, \quad g(v_j) = \frac{1}{v_j} \quad (3.13)$$

for inclusion propensity functions $f, g$ in (3.11) which control the size and richness of one-away neighborhoods. $M$ is defined to be a target “neighborhood budget” (likelihood evaluations), noting the expected size of an Add neighborhood is

$$\sum_{j \notin \gamma} f(v_j = 1) = M(p - |\gamma|)/(p + M) \approx M$$

initially\(^1\). When true model size $|\gamma_0| \ll p$, a majority of the predictors retain $v_j \approx 1$, thus stochastic control over neighborhood size is maintained\(^2\). One may dynamically adjust $M(|\gamma|; \{v\})$ so that $|N_m(\gamma)| \leq B$ with high probability. By default, we propose setting $B = \max\{10, [p^{1/2}]\}$. Predictor propensity scores are increased for predictors frequently included in the past and adaptations designed to be modest. For $j = 1, \ldots, p$,

$$v_j(t + 1) = v_j(t) + \gamma_j (I(t \leq b_0) (t/b_0) + I(t > b_0) (t - b_0)^{-\zeta}) \quad (3.14)$$

for learning rate $\zeta \in (1/2, 1]$, burn-in period $b_0 = \max\{100, |T/10|\}$ and a MCMC chain of length $T \geq 100$. Following convention from stochastic gradient descent, we

\(^1\) Under assumed sparsity, $\mathbb{E}|\gamma| = d^* \ll p$; see Section 3.4.1.

\(^2\) For paired Swap moves, $f(v) \leftarrow f(v)/|\gamma|$ in (3.13) maintains the same limit.
fix $\zeta = 2/3$. Stationarity of the pMTM sampler is maintained subject to diminishing adaptation of predictor importance scores (Roberts and Rosenthal, 2007).

3.5 Examples of variable selection for GLMM

We focus on sampling a $p$-variate inclusion vector using a Metropolis Hastings step imbedded within a larger Gibbs sampler. Applications to linear, binary and log-linear regression are considered, with a brief outline of the sampling scheme in each. The first two permit conjugate updating, while data augmentation is used for count data, with priors appropriately chosen to accommodate over dispersion. In Chapter 4, pMTM sampling is used in a nonparametric regression model, with simulated and real data studies demonstrating state-of-the-art inference and predictive performance.

3.5.1 Variable selection in conjugate settings

For predictor-response pairs $(x, y) \in \mathbb{R}^p \times \mathbb{R}$, the linear regression model augmented with variable selection priors models response as $y|x, \gamma, \beta, \sigma^2 \sim N(x' \beta, \gamma, \sigma^2)$. A hierarchical model specification is completed by assigning prior (3.5) to $\gamma = (\gamma_1, \ldots, \gamma_p)$ along with a conditionally conjugate prior on $(\beta, \sigma^2)|\gamma \sim N(0, \sigma^2 \Sigma_\beta) \times IG(a, b)$. In fact, the posterior over predictor inclusion factorizes in this setting, with marginal inclusion probabilities available in closed form (Clyde, 1999; Ghosh and Clyde, 2011). Nevertheless, a basic MCMC scheme alternates between (1) sampling $\gamma \sim \pi(\gamma|y) \propto \pi(\gamma) p(y|\gamma)$ via a Metropolis step, and (2) drawing $(\beta, \sigma^2)|\gamma, y$ from the analytic NIG conditional posterior distribution.

For binary response data $y \in \{0, 1\}$, the Albert and Chib (1993) data augmentation can be used to obtain a Gibbs sampling procedure via the probit link. Here $\Pr(y = 1|x, \beta) = \Phi(x' \beta)$ under the standard Gaussian distribution function $\Phi$, induced by setting $y = I(z > 0)$ for latent variable $z|x, \beta \sim N(x' \beta, 1)$. Non-zero coefficients are given the standard normal prior $\beta|\gamma \sim N(0, \Sigma_\beta)$, and a Gibbs sam-
pler alternates between (1) sampling \( \gamma \sim \pi(\gamma|z,y) \propto \pi(\gamma)p(z|\gamma) \) conditional on the latent variables \( z = (z_1, \ldots, z_n)' \) via a Metropolis step; (2) imputing latent variables \( z_i|\beta, \gamma, y \) from their truncated normal full conditionals in parallel for \( i = 1, \ldots, n \); and (3) sampling \( \beta|\gamma, z, y \) from its \( p \)-variate Gaussian full conditional.

3.5.2 Variable selection in a non-conjugate setting

We consider an application of log-linear regression for modeling count data with observation-specific random effects. The model is given as

\[
y_i|\lambda_i \sim \text{Po}(\lambda_i), \quad z_i = \log \lambda_i = x_i^T \beta + \epsilon_i
\]

\[
\epsilon_i|\nu_i \sim \text{N}(0, \nu_i^2), \quad \nu_i = \sigma \xi_i, \quad 1/\sigma^2 \sim \text{Gamma}(a, b)
\]

\[
\xi_i|\rho \sim \text{Bernoulli}(\rho), \quad \rho \sim \text{Beta}(c, c), \quad c < 1.
\]

The introduction of observation-specific random effects in the Poisson regression model accommodates overdispersion frequently encountered in count data and is a standard technique used for robustness. The sampling model reduces to standard form when \( \xi_i = 0 \), i.e., \( y_i|x_i, \beta \sim \text{Po}(\exp(x_i^T \beta)) \). For variable selection, we place spike-slab priors on each regression coefficient as

\[
\beta_j|\gamma_j \sim (1 - \gamma_j) \delta_0(\beta_j) + \gamma_j N(0, \sigma^2) \quad \gamma_j|\tau \sim \text{Bernoulli}(\tau), \quad j = 1, \ldots, p.
\]

The posterior distribution over predictor coefficients is not analytically available due to non-conjugacy of the standard normal prior with respect to the Poisson likelihood. Clyde and Desimone (1997) use reversible-jump MCMC (RJ-MCMC) (Green, 2003) to sample from a posterior over \( (\beta_j, \gamma) \) using approximate predictor inclusion probabilities derived under a variable stabilizing transform of the response. It should be noted that the approximate inclusion probabilities are based on orthogonality in the data design matrix (e.g., using PCA). These estimates may be transformed back to obtain inclusion estimates for the original predictors, however these are no longer
necessarily sparse. Similar approximate probabilities may be incorporated into the 
pMTM algorithm as an initialization for weight function $\omega$ (see Section 3.4.5).

Instead of resorting to the general RJ-MCMC scheme, we allow for multiple local 
add/remove/swap (birth-death) moves within the context of our proposed stochastic 
neighborhood sampler. This avoids using the derived approximate normal-product 
Bernoulli proposal distribution (Clyde and Desimone, 1997) which may degrade mix-
ing in the presence of over dispersion. For simplicity we set $\tau = 1/p$, though learning 
of this hyper-parameter may be easily incorporated within the following blocked 
MCMC sampling scheme:

1. Update $\gamma, \beta, \sigma^2 | \{\xi_i, z_i\}, y$: This step proceeds by (1) sampling $\gamma \sim \pi(\gamma|\xi, z) \propto \pi(\gamma|1) t_{2a}(z|0, \text{diag}(\xi_1, \ldots, \xi_n) + X_\gamma X_\gamma^T, b/a)$ via a Metropolis step, and (2) sam-
pling $\beta, \sigma^2 | \gamma, \xi, z$ from its conjugate normal-inverse gamma distribution

2. Update $\{\xi_i, z_i\} | \beta, \sigma^2, y$: Draws from this conditional distribution are made 
in parallel for $i = 1, \ldots, n$: (1) sample $\xi_i|y_i, \beta, \sigma^2 \sim \text{Bernoulli}(\rho_i)$ with $\rho_i = \frac{1-p}{p} \times \frac{\text{Po}(y_i|\lambda_i=\exp(x_i^T \beta))}{\int \text{Po}(y_i|\exp(z_i)) \pi(z_i|1, \beta, \sigma^2) dz_i}$, and (2) sample $z_i \sim \pi(z_i|\xi_i, \beta, \sigma^2, y_i) \propto p(y_i|z_i) \pi(z_i|\xi_i, \beta, \sigma^2)$ via a Metropolis step (n.b. if $\xi_i = 0$ 
then $z_i = x_i^T \beta$)

3. Sample $\rho \sim \pi(\rho|\cdot) = \text{Beta}(1 + \#\{\xi_i = 1\}, 1 + \#\{\xi_i = 0\})$.

Sampling in step 2 from collapsed conditional distribution $\xi_i|y_i, \beta, \sigma^2$ depends on an 
integral which can be approximated on the fly via numerical quadrature.

Alternatively, consider a local-global shrinkage approach wherein $z_i|\lambda_i, \beta, \sigma \sim 
\text{N}(x_i^T \beta, \nu_i^2 = \sigma^2 \lambda_i), \lambda_i \sim \text{Exp}(s_i^2/2), s_i \sim \text{Gamma}(\alpha, \eta)$ independently for $i = 1, \ldots, n$ 
and $1/\sigma^2 \sim \text{Gamma}(\alpha, \eta)$ as before. Marginally, $(z_i - x_i^T \beta)|\beta, \sigma \sim \text{GDP}(\sigma\eta/\alpha, \alpha)$ 
(Armagan et al., 2013). Here, state-of-the-art shrinkage prior is used to ensure that 
the relaxation is “soft” and we may set $\alpha = \eta = 1$. A blocked Metropolis-within-
Gibbs sampler iterates between sampling $\gamma, \beta, \sigma | \{\lambda_i, z_i\}, y$ and $\{\lambda_i, z_i\} | \gamma, \beta, \sigma, y$ as:
1. Update $\gamma, \sigma \mid \lambda_i, z_i, y$ by (1) sampling $\gamma \sim \pi(\gamma \mid z, y)$ via a Metropolis step, (2) drawing $\sigma \mid \gamma, z, \lambda, y$ from its conjugate inverse-gamma distribution, and (3) sample $\beta \mid \gamma, \sigma, \{\lambda_i, z_i\}, y$ from its normal conditional distribution.

2. Update $\lambda_i, z_i$— in parallel for $i = 1, \ldots, n$ by (1) sampling $z_i \mid \gamma, \beta, \sigma, y$ via a Metropolis step, and (2) sampling $s_i \mid z_i, \beta, \sigma, y \sim \text{Gamma}(\alpha + 1, \eta + |z_i - x'_i \beta|/\sigma)$ and $\lambda_i \mid s_i, z_i, \beta, \sigma, y \sim \text{gIG}(1/2, s_i^2, ((z - x'_i \beta)/\sigma)^2)$.

This leads to an efficient data augmentation sampling scheme. Iterating between these steps yields posterior draws for model parameters $(\beta, \gamma)$ that target the corresponding stationary distribution over this parameter subspace.

### 3.6 Analysis of pMTM

We compare pMTM with two popular and widely used MCMC variable selection methods on four well-known real datasets of moderate dimension. SSVS denotes a spike-slab inclusion/exclusion of predictors drawing from the sequence of full conditionals $\gamma_j \mid \beta_{-j}, y$ and $\beta_j \mid \gamma_j, \beta_{-j}, y$, $1 \leq j \leq p$ at every MCMC iteration. Sampling via pMTM follows the procedure outline in Section 3.5, and ‘Toggle’ uses a simple MH step with the same add/remove/swap proposals to traverse the model space. Specifically, if ‘add’, propose $\gamma' = \gamma + 1_j, j \in [\gamma]^c$ proportional to propensity score $v_j$; if ‘remove’, propose $\gamma' = \gamma - 1_j, j \in [\gamma]$ proportional to $1/v_j$; otherwise propose $\gamma' = \gamma + 1_j - 1_k, (j, k) \in [\gamma] \times [\gamma]^c$ proportional to $v_j/v_k$. The latter uses adaptive predictor propensity scores that are updating as described in Section 3.4.6.

Out-of-sample predictive root-mean-squared error is reported as ‘pred r-MSE’; in addition, ‘log0.95 post’ denotes the average 95-th percentile of the negative log posterior probability (smaller is better), whereas $\alpha_{\text{Accept}}$ denotes the averaged acceptance probability (each computed over the entire MCMC chain post burn-in). Individual MCMC chains were run for 3000 iterations, with the initial 20% discarded as burn-in
and subsequent draws thinned by 5. In real data experiments, ‘r-MSE’ is reported
for regression coefficient estimates against least-square values for non-zero predictors
selected by the Lasso. Tables report averaged performance across 10 cross validated
(CV) splits with standard errors appearing as subscripts. An average standard-error
for marginal inclusion probability estimates is also reported: for each $j : |\beta_{0j}| > 0,$
compute $s(\hat{p}_j) = \sqrt{\text{var}(\hat{p}_{1j}, \ldots, \hat{p}_{rj})}$ and set $sd(\hat{p}_{\text{incl}}) = \text{mean}(s(\hat{p}_j)).$

A basic implementation evaluates pMTM neighborhood proposals serially and
we allow Toggle to evaluate $[0.5(1 + \sqrt{p})]$ draws from $\pi(\gamma | y)$ during each MCMC
iteration, which amounts to approximately equal runtime for the predictor dimension(s) considered. For the ‘cement’ dataset, likelihood evaluations are more ex-
pensive as the training data size is relatively large, hence Toggle proposes a sin-
gle update to $\gamma$ at each MCMC iteration. Unless stated otherwise, pMTM targets
$M = \min\{10, [0.5(1 + \sqrt{p})]\}$ neighborhood moves and computes their approximate
likelihood scores as discussed in Section 3.7.

3.6.1 Real data comparisons

<table>
<thead>
<tr>
<th></th>
<th>sample size</th>
<th>predictors</th>
<th>response</th>
<th>transformation(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cement</td>
<td>1030</td>
<td>8 (36)</td>
<td>CCS</td>
<td>log(age)</td>
</tr>
<tr>
<td>ozone</td>
<td>330</td>
<td>9 (44)</td>
<td>OC</td>
<td>log(OC)</td>
</tr>
<tr>
<td>cookie</td>
<td>72</td>
<td>700</td>
<td>fat</td>
<td>-</td>
</tr>
<tr>
<td>boston</td>
<td>506</td>
<td>13 (91)</td>
<td>medv</td>
<td>log(medv)</td>
</tr>
</tbody>
</table>

Real data comparisons demonstrate that sampling via pMTM produces results
that are consistently best among MCMC competitors for a fixed runtime. pMTM
outperforms Toggle on the ‘cement’ dataset in both coefficient and predictive r-MSE
(see Table 3.2). In addition, the log-posterior plot in Figure 3.2 suggests that pMTM
may be more efficient (on per iteration basis) than Toggle, though further analysis is necessary to better quantify the accuracy vs. speed tradeoff. For the biscuit-dough ‘cookie’ data, predictors exhibit a high degree of correlation and while no competitor is significantly different, pMTM fairs better on predictive performance, on average. The average log-posterior score for SSVS here is smaller due to the fact that it includes many fewer predictors compared to the other two methods. Figure 3.2 provides evidence that pMTM and Toggle produce similar inclusion probability estimates, whereas SSVS often departs significantly from the other two. In addition, variability for SSVS inclusion probability estimates over replicate runs (see $\hat{sd}(\hat{p}_{incl})$ in Tables 3.4 - 3.7) appear to be higher than the other two competitors.

Table 3.2: Summary performance statistics. All methods run for approximately an equal amount of CPU time. Median values of summary statistics and standard-errors reported are obtained across 10 cross validated splits. $\hat{sd}(\hat{p}_{incl})$ is reported as an average over all $p$ predictors.
3.6.2 Simulation comparisons

For simulation experiments we use predictors from the ‘communities and crime’ and ‘boston housing’ UCI datasets to retain realistic correlation structure. As before, we include all two-way interaction effects for the latter. Predictors are standardized to be mean zero and variance 1. Response data is generated as $y \sim N(x'\beta, \sigma_0^2)$, $x \in \mathbb{R}^{p+1}$ (a column of one-s is appended to standardized design matrix for an intercept). We set $|\beta_{0j}| = 1$ for predictors included in the model, see Table 3.3.

Table 3.3: Simulated experiments

| data  | signal ($|\beta_{0j}|$, $\sigma_0$) | correlation | chosen | predictors |
|-------|-----------------------------------|-------------|--------|------------|
| 1 boston | (1,1)                            | high        | 12 / 92 | \{(1,19,24), (10,60,85), (30,35,65), (46,50,79)\} |
| 2 crime  | (1,1)                            | high        | 12 / 102 | \{(3), (1,11,28), (10,17), (20,21,85), (50,72,93)\} |
| 3 crime  | (1,2)                            | high        | 12 / 102 | \{(3), (1,11,28), (10,17), (20,21,85), (50,72,93)\} |
| 4 crime  | (1,1)                            | low         | 10 / 102 | \{(1, 12, 22, 34, 43, 61), (10, 48, 89, 100)\} |

As before, Toggle evaluates $[0.5(1 + \sqrt{p})]$ draws per iteration to match with the serial compute-time for pMTM with target neighborhood size $M = 10$ (see Tables 3.4 - 3.7). Experiments 1-3 select groups of highly correlated predictors. Experiments 2 and 3 enable performance measure as the signal-to-noise ratio (SNR) changes for fixed design and predictor truth. Experiments 2 and 4 allow us to compare the effect of correlation amongst true non-zero coefficients on inferential and predictive performance. Tables report medians and standard deviations on summary statistics across $r = 10$ CV splits. Table 3.3 reports average coverage and ‘r-MSE’ on the true non-zero predictors. A coefficient of variation statistic $c\text{.var} = \frac{1}{p} \sum_{j=1}^{p} \hat{\sigma}_j/\hat{\mu}_j$, $\hat{\sigma}_j = \text{stdev}(\{\hat{\mu}_j\})$, $\hat{\mu}_j = \text{mean}(\{\hat{\mu}_j\})$, is also reported on marginal inclusion probabilities for the true non-zero predictors.
Table 3.4: Simulation 1, ‘boston’; high correlation; SNR (1,1).

<table>
<thead>
<tr>
<th>Method</th>
<th>r-MSE*</th>
<th>pred r-MSE</th>
<th>log_{0.95} post</th>
<th>( \alpha_{\text{Accept}} )</th>
<th>coverage*</th>
<th>c.var*</th>
</tr>
</thead>
<tbody>
<tr>
<td>pMTM-10</td>
<td>0.565_{0.03}</td>
<td>1.049_{0.11}</td>
<td>1462.5_{6.3}</td>
<td>0.345_{0.01}</td>
<td>0.692_{0.06}</td>
<td>0.305</td>
</tr>
<tr>
<td>SSVS</td>
<td>0.700_{0.14}</td>
<td>1.057_{0.11}</td>
<td>1465.9_{6.0}</td>
<td>1.000_{0.00}</td>
<td>0.458_{0.16}</td>
<td>2.995</td>
</tr>
<tr>
<td>Toggle</td>
<td>0.582_{0.05}</td>
<td>1.042_{0.11}</td>
<td>1461.7_{6.0}</td>
<td>0.194_{0.03}</td>
<td>0.675_{0.11}</td>
<td>0.483</td>
</tr>
<tr>
<td>Lars</td>
<td>0.553_{0.05}</td>
<td>1.073_{0.11}</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.5: Simulation 2, ‘crime’; high correlation; SNR (1,1).

<table>
<thead>
<tr>
<th>Method</th>
<th>r-MSE*</th>
<th>pred r-MSE</th>
<th>log_{0.95} post</th>
<th>( \alpha_{\text{Accept}} )</th>
<th>coverage*</th>
<th>c.var*</th>
</tr>
</thead>
<tbody>
<tr>
<td>pMTM-10</td>
<td>0.334_{0.08}</td>
<td>1.001_{0.13}</td>
<td>1146.1_{5.4}</td>
<td>0.270_{0.01}</td>
<td>0.942_{0.10}</td>
<td>0.190</td>
</tr>
<tr>
<td>SSVS</td>
<td>0.570_{0.14}</td>
<td>0.966_{0.20}</td>
<td>1150.0_{4.8}</td>
<td>1.000_{0.00}</td>
<td>0.641_{0.24}</td>
<td>0.251</td>
</tr>
<tr>
<td>Toggle</td>
<td>0.365_{0.08}</td>
<td>0.986_{0.11}</td>
<td>1148.5_{5.0}</td>
<td>0.206_{0.03}</td>
<td>0.925_{0.14}</td>
<td>0.197</td>
</tr>
<tr>
<td>Lars</td>
<td>0.715_{0.10}</td>
<td>1.065_{0.20}</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.6: Simulation 3, ‘crime’; high correlation; SNR (1,2).

<table>
<thead>
<tr>
<th>Method</th>
<th>r-MSE*</th>
<th>pred r-MSE</th>
<th>log_{0.95} post</th>
<th>( \alpha_{\text{Accept}} )</th>
<th>coverage*</th>
<th>c.var*</th>
</tr>
</thead>
<tbody>
<tr>
<td>pMTM-10</td>
<td>0.861_{0.05}</td>
<td>2.020_{0.21}</td>
<td>1409.0_{7.3}</td>
<td>0.220_{0.02}</td>
<td>0.325_{0.09}</td>
<td>1.688</td>
</tr>
<tr>
<td>SSVS</td>
<td>0.861_{0.07}</td>
<td>2.072_{0.20}</td>
<td>1411.2_{7.5}</td>
<td>1.000_{0.00}</td>
<td>0.325_{0.20}</td>
<td>3.642</td>
</tr>
<tr>
<td>Toggle</td>
<td>0.879_{0.05}</td>
<td>1.990_{0.22}</td>
<td>1408.9_{7.5}</td>
<td>0.112_{0.02}</td>
<td>0.325_{0.07}</td>
<td>2.422</td>
</tr>
<tr>
<td>Lars</td>
<td>0.824_{0.02}</td>
<td>2.081_{0.26}</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 3.7: Simulation 4, ‘crime’; low correlation; SNR (1,1).

<table>
<thead>
<tr>
<th>Method</th>
<th>r-MSE*</th>
<th>pred r-MSE</th>
<th>log_{0.95} post</th>
<th>( \alpha_{\text{Accept}} )</th>
<th>coverage*</th>
<th>c.var*</th>
</tr>
</thead>
<tbody>
<tr>
<td>pMTM-10</td>
<td>0.289_{0.05}</td>
<td>0.981_{0.25}</td>
<td>1172.2_{8.5}</td>
<td>0.291_{0.02}</td>
<td>0.860_{0.08}</td>
<td>0.077</td>
</tr>
<tr>
<td>SSVS</td>
<td>0.328_{0.14}</td>
<td>0.989_{0.24}</td>
<td>1173.7_{8.3}</td>
<td>1.000_{0.00}</td>
<td>0.830_{0.11}</td>
<td>0.155</td>
</tr>
<tr>
<td>Toggle</td>
<td>0.286_{0.04}</td>
<td>0.981_{0.23}</td>
<td>1172.0_{8.6}</td>
<td>0.230_{0.02}</td>
<td>0.860_{0.10}</td>
<td>0.058</td>
</tr>
<tr>
<td>Lars</td>
<td>0.284_{0.06}</td>
<td>1.030_{0.27}</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

In high correlation settings (Experiments 1-3), pMTM is consistently best among the variable selection samplers. Toggle (with multiple draws per iteration) is competitive with pMTM. SSVS suffers on coverage in Experiments 1-2 even with relatively high SNR \((\sigma_0 = 1)\), and results in significantly higher coefficient r-MSE in Experiments 1-3. In the high signal and low correlation setting (Experiment 4), all the sampling competitors essentially perform equally well. Consistent with findings on real data experiments, Figure 3.3 demonstrates that pMTM and Toggle quickly stabilize in a region of high posterior probability in contrast to SSVS, which consistently takes longer to do so.
3.7 Extensions and future work

The pMTM of Section 3.4.5 delivers state-of-the-art predictive and inferential performance across a variety of simulated and real data settings. Trace-plots for the log-posterior score suggest that pMTM quickly identifies regions of high posterior probability with improved acceptance rates over Toggle. Future work will

1. investigate the number of models explored by the variable selection schemes post burn-in and better quantify the reliability with which marginal predictor inclusion probabilities are estimated;

2. quantify the accuracy vs. speed tradeoff as a function of the predictor and data dimension. Compute-time in initial experiments was kept equal across sampling competitors, however, pMTM neighborhoods can instead be distributed across cores/machines, with subsampling allowing for lightweight score (likelihood) evaluations (see Section 3.7.2); and

3. extend pMTM to more complex settings (e.g., the GLMM setting; see Chen and Dunson (2003), Cai and Dunson (2006)) and study asymptotic efficiency of MCMC output under the stochastic multiple-try neighborhood (pMTM) scheme as compared to simpler ‘Toggle’ samplers (Peskun, 1973; Liu, 1996; Tierney, 1998).

3.7.1 Improved adaptivity

Learning predictor propensity scores \( v_j, j = 1, \ldots, p \) as described in Section 3.4.6 can dramatically improve the stability and robustness of an MCMC sampler in estimating predictor inclusion probabilities and predictive MSE in both high dimensional and correlated predictor settings. In spectrometry or gene expression data, for example, predictors are often highly correlated because of their spatial proximity, and therefore “exchangeable” with respect to their explanatory power. It is well known that penalized methods such as the Lasso (Tibshirani, 1996) often simply selects one
out of a set of highly correlated predictors, and the elastic-net penalty (Zou and Hastie, 2005) is often a more robust shrinkage method to use in such settings. A regularized estimate for the correlation matrix (Schäfer and Strimmer, 2005; Bickel and Levina, 2008) or other similarity measures between predictors may be used to efficiently update propensity scores. Define $\widehat{C}_{ij} = |\rho_{ij}| I(|\rho_{ij}| \geq \epsilon)$ as a thresholded absolute correlation matrix, where $\rho_{ij} = \text{Corr}(x_i, x_j)$ is the empirical correlation between predictors $1 \leq i, j \leq p$ and $\epsilon \in (0, 1)$ is a pre-specific threshold. The propensity score adaptation (3.14) may be modified as

$$v_j(t + 1) = v_j(t) + z(\{\widehat{C}_{i,j}\}) (I(t \leq b_0) (t/b_0) + I(t > b_0) (t - b_0)^{-\zeta})$$

(3.15)

for $z : [0, 1]^d \rightarrow [0, 1]$; in particular, $z = (1 - \gamma_j) (\sum_{k=1}^{p} \gamma_k \widehat{C}_{kj} / \sum_{k=1}^{p} \gamma_k) + \gamma_j$.

3.7.2 Speeding up pMTM via subsampling

Marginal likelihood evaluations in the linear regression setting require $O(n d^3)$ flops, $d = |\gamma|$, since inverting the $n \times n$ covariance matrix only requires inverting a $d \times d$ matrix by the Sherman-Woodbury formula. Evaluations in more general model setting (e.g., nonparametric regression) often can be reduced to scale linearly in the number of data points as well (see Section 4.5.2). When the predictor dimension is very large (e.g., $p \approx 10^6$), one faces the need to explore a large number of one-away neighbors to effectively search the model space and learn which predictors are important. When scoring inclusion vectors in (3.7) proportional to their posterior probability, $h(\hat{\gamma}|\gamma, -) \propto \pi(\hat{\gamma}|-)$, pMTM requires on average $M(1 + w_S(|\gamma|))$ likelihood evaluations per iteration.

Several recent works attempt to reduce the computational bottleneck in processing very large datasets by considering fixed, stochastic or streaming shards (Quiroz et al., 2014; Maclaurin and Adams, 2014; Guhainiyogi et al., 2014; Korattikara et al., 2013). One possible direction for scaling pMTM is to use proposal distributions based
on data subsampling wherein a random subset of points is used to score neighborhood proposals. That is, one may take score function $h(\gamma'|\gamma)$ as the unnormalized posterior probability for an inclusion proposal $\gamma'$ evaluated on a set of randomly chosen indices at each MCMC iteration; in particular, $h_S(\gamma') = p(y_S|\gamma')\pi(\gamma')$ for a random index set $S \subset \{1, \ldots, n\}$. This allows the $M$-budgeted pMTM sampler to update inclusion vectors at a $O(M|S|)$ per-iteration cost instead of $O(Mn)$.

**Lemma 3.6.** For paired-moves $(m, m') \in \{(A, R, S), (R, A, S)\}$, a modified pMTM sampler that uses a random subset of data, $S \subset \{1, \ldots, n\}$, to score proposals $\gamma' \in N_m(\gamma)$ as $h_S(\gamma') = p(y_S|\gamma')\pi(\gamma')$ satisfies detailed balance with the original pMTM acceptance probability (3.12) corrected by scaling factor $\frac{\pi(\gamma')h_S(\gamma)}{\pi(\gamma)h_S(\gamma)}$.

**Proof.** We show reversibility for paired-move $(m, m') \in \{(A, R), (R, A)\}$ is maintained, with the proof for paired Swap moves following trivially.

$$\pi(\gamma)T(\gamma, \gamma')$$

$$= \pi(\gamma)w_m(|\gamma|) \sum_{\eta, \eta_i \in \{0, 1\}^p \atop \eta_k = \eta'_k = 1} \omega_k \left\{ \prod_{j \neq k} \omega_j^{\eta_j} (1 - \omega_j)^{1 - \eta_j} \omega_j'^{\eta_j'} (1 - \omega_j')^{1 - \eta_j'} \right\}$$

$$\times \frac{h_S(\gamma')}{\sum_{j, \eta_j = 1} h_S(\text{top}(j, \gamma))} \min \left\{ 1, \frac{\pi(\gamma')h_S(\gamma)}{\pi(\gamma)h_S(\gamma')}, \frac{w_{m'}(|\gamma'|)}{w_m(|\gamma|)} \frac{\omega_k \sum_{j: \eta_j = 1} h_S(\text{top}(j, \gamma))}{\omega_k \sum_{j: \eta_j' = 1} h_S(\text{top}(j, \gamma'))} \right\}$$

$$= \sum_{\eta, \eta_i \in \{0, 1\}^p \atop \eta_k = \eta'_k = 1} \left\{ \prod_{j \neq k} \omega_j^{\eta_j} (1 - \omega_j)^{1 - \eta_j} \omega_j'^{\eta_j'} (1 - \omega_j')^{1 - \eta_j'} \right\}$$

$$\times \min \left\{ \frac{w_m(|\gamma|) \pi(\gamma)h_S(\gamma')}{\sum_{j, \eta_j = 1} h_S(\text{top}(j, \gamma))}, \frac{w_{m'}(|\gamma'|) \omega_k \pi(\gamma')h_S(\gamma)}{\sum_{j, \eta_j' = 1} h_S(\text{top}(j, \gamma'))} \right\}.$$

The expression is symmetric and therefore satisfies detailed balance, leaving the stationary distribution unchanged. \qed
Figure 3.2: Marginal inclusion probabilities and log-posterior trace plots shown averaged over 10 cross validated replications for datasets considered in Table 3.1.
Figure 3.3: Simulated data with setup given in Table 3.3.
Modern regression methods for *large-p small-n* data often impose overly restrictive modeling assumptions and suffer considerably in terms of statistical estimation and computational scalability. Additive-interactive regression has recently been shown to offer attractive minimax error rates over traditional nonparametric multivariate regression in a wide variety of settings. Within this framework, we develop an efficient and scalable Bayesian implementation for the modeling of potentially complex predictor-response relations using an additive Gaussian process (aGP) prior. Careful prior specification is developed in light of performance and computational considerations. Key innovations address challenges in exploring the joint posterior over multiple subsets of high dimensional predictor inclusion vectors by employing an efficient neighborhood stochastic-search Markov chain sampler. aGP offers state-of-the-art support and interaction recovery while improving dramatically over competitors in terms of prediction accuracy on a diverse set of simulated and real data. Results from real data studies provide strong evidence that the additive-interactive framework is an attractive modeling platform for high-dimensional nonparametric regression.
4.1 Introduction

The collection of large volumes of rich and complex data has become ubiquitous in recent years, and there is a need for multivariate regression techniques that efficiently capture a wide range of naturally occurring predictor-response relations, identify important predictors and their interactions and doing so even when the number of predictors is large but the sample size remains limited. In genomics, modern sequencing techniques can measure hundreds of thousands of genetic markers simultaneously from individuals, broadening the scope of early detection of complex diseases and identification of important causative mutations. Elsewhere, association discovery and intervention effect quantification (e.g., in observational studies and survey data) are routinely required in high impact scientific research (Hill, 2011).

Much of the high dimensional regression literature focuses on parametric linear models regularized with sparsity and shrinkage (Tibshirani, 1996; Candes and Tao, 2007; Hastie et al., 2009). By requiring the parametric relation to hold globally, such methods remain prone to introducing bias when quantifying treatment effects in non-randomized observational studies. In addition, these methods run the risk of over-parametrization when attempting to adjust for non-linearity or to uncover predictor interaction. Restricted by their linearity and additivity assumptions, these methods often fail to adequately model many naturally occurring predictor-response relations. Several smoothing based nonparametric regression methods (O’Hagan and Kingman, 1978; Lafferty and Wasserman, 2008; Bertin and Lecué, 2008; Williams and Rasmussen, 1996) accommodate a wider range of predictor-response relations and come with mathematical guarantees of delivering good performance in various settings (Lafferty and Wasserman, 2008; van der Vaart and van Zanten, 2008, 2009; Bhattacharya et al., 2011; Tokdar, 2012). However, the computational demands of these methods scale poorly with predictor dimension due to costly likelihood and
score function evaluations. More importantly, with a traditional sparse nonparamet-
ric regression model in which $f$ is assumed to depend on $d$ of the original $p$ predictors,
statistical estimation greatly suffers from the curse of dimensionality in the high di-
menional setting. In this setting, the minimax $L_2$ estimation risk $r_{n}^{1}$ based on $n$
observations is of the order $r_{n}^{2} \asymp n^{-2\alpha/(2\alpha+d)} + d/n \log(p/d)$, where $\alpha$
denotes the degree of smoothness of $f$ (Yang and Tokdar, 2014). The second term is the penalty
paid for variable selection, and remains small even in $large \; p \; small \; n$ situations, i.e.,
when $p$ is as large as $\exp(n^{\beta})$, $\beta \in (0, 1)$. For $p$ of this order, the first term gives
the standard fixed-dimension minimax risk for a $d$-variate $\alpha$-smooth function (Stone,
1982), and is small only when $d = o(\log n) = o(\log \log p)$, i.e., $f$ is extremely sparse
in the observed predictors.

A generalization of the sparse nonparametric model with attractive minimax
properties is the additive-interactive model where the unknown regression function
$f$ is theorized to decompose as $f = f_1 + \cdots + f_k$ for some $k \geq 1$, with each component
function $f_s$ depending nonparametrically on a small number of predictors (Ouyang,
2008; Chipman et al., 2010). Yang and Tokdar (2014) show that the $L_2$ minimax
risk of estimating $f$ remains small in the $p \gg n$ setting even when $f$ involves many of
the predictors, as long as the number of predictors included in any single component
is bounded. By restricting the size of each component, which restricts the maximum
order of interaction, one can therefore learn the effect of many more predictors by
increasing the number of additive components. In particular, if every component
function is $\alpha$-smooth and includes $d$ predictors, the minimax risk remains small
even with $k = o(n^{\zeta})$, $\zeta \in (0, 1)$ components. This corresponds to the inclusion of
d$n^{\zeta} \asymp (\log p)^{\zeta/\beta}$ predictors.

Although ensemble learning methods (Friedman, 1991; Freund and Schapire,
1997; Breiman, 2001; Chipman et al., 2010) also employ additive constructions, the

---

1 Minimax risk of estimation defined in Section 4.2
The overarching goal of using many “weak learners” is to boost overall efficiency and avoid the risk of overfitting, the merit of which remains suspect when $p \gg n$. More importantly, additivity is not used to address the large $p$ small $n$ curse of dimensionality by uncovering simpler structures, and the methods behave as black-box forecasting machines offering little in terms of inference on predictor importance and interaction. In contrast to these approaches, the additive-interactive framework makes a clear modeling assumption that a high dimensional $f$ divides into low dimensional pieces added together, allowing statistical methods to estimate each piece with an efficient low dimensional learner. Predictors are treated exchangeably and learning their importance and interaction patterns may be considered a primary objective, or one that facilitates better prediction. Both Bayesian additive regression trees (BART, Chipman et al., 2010) and Bayesian additive regression kernels (BARK, Ouyang, 2008) could be viewed as implementations of this framework. However, BART’s use of a large number of heavily regularized trees places it closer to the ensemble approach, and BARK is yet to be developed in the high dimensional setting.

4.2 GP regression: a brief overview

Recall that a stochastic process $w = (w(x) : x \in \mathcal{X})$ on a Euclidian domain $\mathcal{X}$ is called a Gaussian process (GP) if every finite collection of elements $(w(x_1), \ldots, w(x_n))$, $n \geq 1$, $\{x_1, \ldots, x_n\} \in \mathcal{X} \subset \mathbb{R}^p$, has a joint Gaussian distribution. Such a process is completely determined by its mean $\mu(x) = \mathbb{E}[w(x)]$ and covariance function $C(x, \tilde{x}) = \mathbb{E}[(w(x) - \mu(x))(w(\tilde{x}) - \mu(\tilde{x}))]$, with the latter being non-negative definite over $\mathcal{X} \times \mathcal{X}$. For any real function $\mu(x)$ and non-negative definite function $C(x, \tilde{x})$, there exists a GP with these functions as its mean and covariance. We refer to the probability law of such a process as GP($\mu, C$). When $\mu$ and $C$ are continuous, $w \sim \text{GP}(\mu, C)$ may be viewed as a random element of $C(\mathcal{X})$, the space of continuous functions over $\mathcal{X}$ equipped with the supremum norm. The smoothness of function
valued realizations of such a random element depends on the smoothness of $C$. For the squared exponential covariance function, $C = C^{SE}(x, \tilde{x}) := \exp(-\|x - \tilde{x}\|^2)$, realizations of $w$ are infinitely smooth elements of $C(\mathcal{X})$.

4.2.1 Gaussian process regression: notation and setup

For a continuous response $y \in \mathbb{R}$ measured with homogenous Gaussian error, the conditional distribution of the response given its associated $p$-dimensional predictor $x \in \mathcal{X}$ is modeled as

$$y = f(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, \sigma^2) \tag{4.1}$$

where $f(x) : \mathcal{X} \to \mathbb{R}$ is the unknown regression function. The linear model assumes $f(x) = x'\theta$ for an unknown $p$-dimensional coefficient vector $\theta$, and Bayesian inference on $f$ proceeds by placing prior $\theta \sim \pi_\theta$ and computing its associated posterior distribution having observed data $\{(x_i, y_i) : 1 \leq i \leq n\}$. The strong parametric assumption, however, often leads to poor statistical estimation and performance as previously discussed.

Instead, we model the data nonparametrically by assuming a Gaussian process prior over the unknown function in (4.1), i.e., $f \sim \text{GP}(\mu, K)$. In particular and without loss of generality, we consider the zero mean and stationary Gaussian process with mean function $\mu = 0$ and squared exponential covariance function

$$K = \tau^2 C^{SE}(x, \tilde{x}; \lambda) = \tau^2 \exp(-\lambda^2 \|x - \tilde{x}\|^2)$$

indexed by nonnegative scaling parameter $\tau$ and inverse length-scale parameter $\lambda$. It is convenient to re-parametrize the covariance scaling as $\tau = \sigma \rho$, with $\rho$ having the interpretation as a signal-to-noise ratio (SNR). Moreover, this leads to important efficiencies under a Bayesian model fitting procedure. The posterior distribution of $f$ given the observed set of data (and $\sigma, \rho, \lambda$) is again a Gaussian process, with a finite-dimensional posterior at the observed set of locations $x_1, \ldots, x_n$ given by

$$f|\sigma, \rho, y \sim \mathcal{N}(\rho^2 C(I_n + \rho^2 C)^{-1} y, (\sigma \rho)^2 C(I_n + \rho^2 C)^{-1}) . \tag{4.2}$$
Expressions for the posterior mean and covariance in (4.2) are scaled by the square of SNR parameter $\rho$, which shrinks toward the prior mean sharply for $\rho \ll 1$. Since $\rho, \lambda$ are typically not known a priori, this observation and the form of (4.2) turn out to be useful for prior elicitation; see Section 4.4.3.

4.2.2 Existing theory guarantee for GP regression

Given $n$ independent observations, an estimator $\tilde{f}$ for an unknown function $f$ has risk $r_n(\tilde{f}, f) = \mathbb{E}\{\ell(\tilde{f}, f)\}$ with respect to a class $A_n$, the space of all measurable functions of data to $C_f$. Here, expectation is taken over data model (4.1) and an $L_2$ loss may be assumed. A minimax estimator $\hat{f} \in A_n \subset C_f$ is defined to be the minimizer of the worst-case risk over all possible truths $f$, i.e.,

$$\sup_{f \in C_f} r_n(\hat{f}, f) = \inf_{f \in A_n} \sup_{f \in C_f} r_n(\tilde{f}, f).$$

In addition, recall that convergence of the posterior distribution for $f$ around the true function $f_0$ at rate $\epsilon_n$ is defined as

$$\lim_{n \to \infty} \Pi \left( \|f - f_0\|_{L_2}^2 \geq M \epsilon_n |Y_1, \cdots, Y_n| \right) = 0 \quad (4.3)$$

whenever $y_i|x_i \sim N(f_0(x_i), \sigma_0^2)$, $1 \leq i \leq n$, and a constant $M$ which depends only on the set of assumptions (e.g., smoothness and sparsity of the assumed function class).

When $f_0$ is known to be $\alpha$-smooth, the minimax rate of estimation is $n^{-\alpha/(2\alpha+p)}$ with dimension $p$ held fixed as shown by Stone (1982). Seminal work by van der Vaart and van Zanten (2008) showed that a rescaled GP prior placed on the space of functions offers adaptive asymptotic posterior convergence at near optimal minimax rates. In particular, a GP prior $w_{A_x}$ having a squared exponential covariance function with inverse length-scale parameter $A$, namely

$$w_A|A \sim \text{GP}(0, C_A), \quad A^p \sim \text{Gamma}(a, b) \quad (4.4)$$

has a posterior which contracts to $f_0$ at the minimax rate up to log factors and is adaptive as the smoothness need not be known a priori\(^2\). However, this rate is

\(^2\) Theoretical treatments of GP regression often assume $\mu = 0$ and squared exponential covariance
completely unsatisfactory in settings where \( p \) is large. This has lead to a number of efforts to establish improved rates for GP regression. Bhattacharya et al. (2011) and Tokdar (2012) showed that if the regression can be cast in a lower dimensional subspace that does not amount to any loss of information about the true function, then a GP model can attain a posterior convergence rate of order \( n^{-\alpha/(2\alpha+d)} \) with \( d \ll p \). Such a subspace is not known a priori, but can be learned adaptively by endowing the GP model with appropriate variable selection priors (see Section 4.3). This has lead to additional theoretical and methodological work for Gaussian processes in recent years (Yang and Tokdar, 2014; Qamar and Tokdar, 2014).

4.2.3 A simple illustrating example

GP regression, in addition to being conceptually simple, is an extremely powerful and versatile modeling framework. Notable among many of its salient features is GP’s ability to (a) smoothly interpolate between regions of the predictor space and borrow information from “nearby” regions (as determined by inverse length-scale \( \lambda \)) to improve estimation where data is sparse; (b) estimate arbitrary shapes and provide reliable posterior credible intervals unlike many parametric and nonparametric counterparts (as measured by frequentist coverage on 95% credible intervals for prediction); and (c) efficiently adapt to the underlying smoothness of the regression function without the need for external tuning of penalty parameters that regularize model complexity.

We consider modeling simple regression functions in the 1D predictor setting to illustrate some of these features. Here GP regression (GPR) is fit using a covariance kernel parametrized as discussed in Section 4.2.1. For various values of inverse length-scale \( \lambda \) (as determined by \( c(\Delta) \) in (4.10)), Figure 4.1 plots point-wise posterior means function \( C(x, \tilde{x}) = \exp(-\|x - \tilde{x}\|^2) \), with additional hyper-parameters placed inside the covariance function; both isotropic and anisotropic kernels have been studied.
and 95% credible intervals for are also plotted for the following functions:

\[
f(x) = \begin{cases} 
1 - 3x & 0 \leq x < 1/3 \\
(3|x - 0.5| + \frac{1}{5}) \sin(9\pi(x - 1/3)) & 1/3 \leq x < 2/3 \\
0 & \text{otherwise},
\end{cases}
\]

\[
h(x) = 2x \sin(2\pi x) \quad 0 \leq x \leq 1.
\]

Larger \(c(\Delta)\) correspond to larger length-scale values for which GPR models global trends in the regression surface as depicted in the far right column of Figure 4.1. In contrast, smaller values for this parameter constrain GPR to borrow information across points at a smaller length-scale, resulting in increased nonlinearity in the reconstructed response surface.

Data are generated from \(y \sim N(\mu(x), \sigma_0 = 0.1)\) for a chosen regression function \(\mu = \{f, h\}\) based on \(n = 50\) fixed locations\(^3\), \(x \in (0,1)\). Reported marginal log-likelihood scores provide a goodness-of-fit diagnostic, demonstrating that the learning of inverse length-scale \(\lambda\) is crucial for GPR to adapt to the complexity (nonlinearity) of the response surface.

Figure 4.1 also depicts the mean predictive surface for BART (Chipman et al., 2010) which serves as a baseline nonparametric competitor and allows us to contrast some of the salient features of the GPR method. In particular, by failing to exploit smoothness in estimating the response surface BART suffers on estimation accuracy. The predictive coverage on the test data for \(f(x)\) was 75% ± 4% for GPR and 72 ± 4% for BART; similarly for \(h(x)\), 72 ± 4% for GPR and 57 ± 4% for BART. An improved model fit over BART (not shown) can be obtained using BTGP (Gramacy and Lee, 2008), however, the latter results in overly conservative (wide) predictive intervals over large regions of the predictor space.

\(^3\) We select 25 uniformly spaced points on \((0, 1/3)\), 15 uniformly spaced points on \((1/3, 1/2)\), and the remaining 10 points uniformly spaced on \((1/2, 1)\).
### 4.3 Variable selection and structured GP regression

In high dimensions when the ambient predictor dimension $p$ is large or ultra large, good recovery of the regression function hinges on the ability to learn important dimensions which are predictive of the response. Automatic relevance determination (ARD; Neal, 1997) is a classic approach used for variable selection in the GP regression setting. Here, one considers anisotropic squared exponential covariance function

$$C^{\text{ARD}}(x, \tilde{x}) = \text{Corr}(f(x), f(\tilde{x})) = \exp\left(- (x - \tilde{x})' P (x - \tilde{x}) \right)$$  \hspace{1cm} (4.5)

with $P = \text{diag}(\lambda_1^2, \ldots, \lambda_p^2)$ for inverse length-scale parameters $\lambda_j$, $1 \leq j \leq p$. Predictor importances are assessed based on inferred values for these hyper-parameters,
and those having \( \lambda_j < 1 \) are effectively excluded from the model.

Bayesian inference on \( \lambda = (\lambda_1, \ldots, \lambda_p) \) using draws from full conditional distributions \( \lambda | f, y \) and \( f | \lambda, y \) are useless in practice due to strong coupling between \( \lambda \) and the latent GP realizations. Evidence maximization (Type-II maximum likelihood) is often used as a means of bypassing sampling difficulties, solving

\[
\hat{\lambda} = \text{argmax}_{(\lambda_1, \ldots, \lambda_p)} p(y|x, \lambda_1, \ldots, \lambda_p)
\]

and subsequently reporting \( f | \hat{\lambda}, y \); see Williams and Rasmussen (2006) for GP model fitting strategies along these lines. However, in high dimensions or correlated predictor settings, optimization-based approaches can be unreliable as they fail to (a) tackle multi-modality in the parameter space, and (b) convey relevant information on relative predictor importance.

Extensions of traditional stochastic search variable methods have been proposed in the GP setting (Linkletter et al., 2006; Savitsky et al., 2011), with re-parameterized GP covariance function enabling MC³ or sequential scan variable selection strategies. These approaches instead sample from \( \lambda | y \) and \( f | \lambda, y \), which conceptually solves the coupling issue but requires constructing clever proposal distributions to update \( \lambda \) which is challenging when \( p \) is large. One approach sets \( \lambda^2_j = -\log \rho_j, \rho_j \in (0, 1], \) endowing each parameter with a spike-slab mixture prior \( \rho_j | \gamma_j \sim \gamma_j I(0 \leq \rho_j \leq 1) + (1 - \gamma_j) \delta(\rho_j - 1), \gamma_j = \{0, 1\} \) independently for \( 1 \leq j \leq p \). A Metropolis Hastings update appears to work reasonably well in moderate dimensions, but tends to suffer increasingly as the predictor dimension grows. In practice, collinearity amongst predictors as well as departures from full additivity (e.g., when the regression surface depends on few or several low-order interaction effects) can hamper mixing efficiency and lead to instability in sampling of the variable selection parameters. While several methods have focussed on scaling GP regression in the large \( n \) setting (Banerjee et al., 2008; Titsias and Lawrence, 2010; Banerjee et al., 2012; Damianou et al., 2014), few
adequately address the challenges of posterior computation for effective dimension reduction in the large-\(p\) setting.

A richer class of covariance functions can be constructed via summation, multiplication or convolution of known stationary and non-stationary elementary kernel functions, while maintaining positive definiteness of the resulting covariance matrix. This gives rise to a tremendous flexibility for modeling plausible relationships between the predictors and the response. In the machine learning literature, the multiple kernel learning framework (Gönen and Alpaydın, 2011) has spurred recent approaches to learn more flexible covariance functions for Gaussian processes, see for example (HKL, Bach, 2009); (GPAdd, Duvenaud et al., 2011). These methods propose a GP covariance function written as the sum over a product of individual univariate basis kernels, \(k_j(x_j, \tilde{x}_j), 1 \leq j \leq p\). A regularized optimization framework is adopted to learn a weighted sum over an exponential number of kernels, showing that efficient “kernel selection” can be performed in polynomial time. In particular, they encode a \(d\)-th order covariance contribution via a product kernel

\[
k_{d}(x, \tilde{x}) = \sigma_d^2 \sum_{j_1, \ldots, j_d = 1}^{d} \prod_{l=1}^{d} k_{j_l}(x_{j_l}, \tilde{x}_{j_l})
\]

where \(1 \leq j_l \leq p\) and \(1 \leq j_1 < \cdots < j_d \leq p\). As before, the final covariance function is specified as \(C(x, \tilde{x}) = \sum_{d=1}^{D} k_{d}(x, \tilde{x})\). When univariate \(k_j(\cdot, \cdot)\) functions are chosen as Gaussian kernels with predictor specific length-scale parameters, then the additive term \(k_p(x, \tilde{x})\) corresponds precisely to the ARD covariance function (4.5). This makes explicit a key flexibility of kernel compositions over the ARD approach to nonparametric variable selection. Whereas the latter inherently learns an all-way interaction between selected predictors, the kernel learning approach enables the modeling of local and non-local effects. In particular, the set of \(k_{d}(\cdot, \cdot)\) are equivalent to a generalized additive model with individual \(f(x_j) \sim \text{GP}(\mu, k_j)\) when \(d = 1\).
While these optimization-based approaches show promise, they intrinsically lack the ability to provide inference on key functionals of interest, e.g., marginal predictor inclusion probabilities.

HKL reduces the complexity of summing over an exponentially large number of kernels by constraining selected kernels to lie in a convex hull\(^4\), allowing individual additive kernel evaluations to be computed at an \(O(d)\) cost, \(1 \leq d \leq p\). GPAdd makes clever use of recursive formulae that allow them to remove this requirement while still computing the sum in \(O(d^2)\) time. However, both methods make fairly strong assumptions on the parametrization of the ‘base kernel’, \(k_{ji}\), for the individual predictor dimensions. More specifically, in addition to requiring all \(d\)-way interactions to share a common scaling \(\sigma_d\), GPAdd associates each predictor dimension with a single smoothness parameter, \(\lambda_{ji}\). Such a ‘dimension reduction’ is presumably crucial for efficient optimization but quite limiting in practice. In particular, there is no reason at all to assume that the smoothness for a main-effect involving predictor \(k\) is in any way \textit{optimal} for higher order interactions involving \(k\).

This chapter seeks to address these inadequacies as well as the fundamental lack of flexible and reliable posterior computation for high dimensional GP regression. Though methods for incorporating lower dimensional structure has been a burgeoning area of applied research over the past two decades, theoretical guarantees and justifications for many approaches have only recently been understood\(^5\). We propose a fully Bayesian implementation of additive-interactive regression using an additive Gaussian process (aGP) prior, combining local multivariate smoothing, sparsity, additivity, and controlled interaction in the large-\(p\) small-\(n\) setting. Careful prior specification is developed in light of performance and computational considerations,

\(^4\) See HKL, Bach (2009) for details.

\(^5\) Appendix C details recent results that have established minimax rates of estimation in the additive regression setting.
with innovations addressing key challenges in exploring the joint posterior over multiple subsets of high dimensional inclusion vectors. One must also contend with an $O(n^3)$ cost for GP likelihood evaluations which, in the context of MCMC sampling, necessitates a procedure that mixes well and minimizes inefficiencies (e.g., frequently rejected proposals).

4.4 Additive GP regression

4.4.1 An Additive-interactive framework for structured sparsity

Let $C(\alpha, d, L)$ denote the space of $d$-variate $\alpha$-smooth functions such that $\|f\|_{C^\alpha} \leq L < \infty$. Sparsity is encoded through the use of binary inclusion vectors $\gamma \in \{0, 1\}^p$, $|\gamma| = \sum_{j=1}^p \gamma_j \leq d$. Let $x_\gamma = \{x_j : \gamma_j = 1\}$ denote the predictors chosen by inclusion vector $\gamma$ and $x \in \mathcal{X}$. We consider “additive” function spaces indexed by $k, p, \bar{d} \in \mathbb{N}$, with $\alpha, L \in (0, \infty)^k$ and $d \in \mathbb{N}^k$. Suppose unknown regression function $f \in \mathcal{M}_A(\alpha, d, L, k, p, \bar{d})$, the class of $k$ irreducible components, each having smoothness $\alpha_h$ and interaction order $|\gamma_h| \leq \bar{d}, 1 \leq l \leq k$. Notationally,

$$\mathcal{M}(\alpha, d, L, k, p, \bar{d}) = \left\{ x \mapsto f = f_1(x_{\gamma_1}) + \cdots + f_k(x_{\gamma_k}) : f_h \in C(\alpha_h, |\gamma_h|, L_h), \right.$$  \hspace{1cm}  \left. \gamma_h \in \{0, 1\}^p, |\gamma_h| \leq |\gamma_h|, \sum_{h=1}^k b_{h,j} \leq \bar{d}, \gamma_h \neq \gamma_m, \right.$$  \hspace{1cm}  \left. l, m = 1, \ldots, k; l \neq m; j = 1, \ldots, p \right\} \hspace{1cm} (4.6)

where each predictor is restricted to appear in at most $\bar{d} < k$ additive components, and no component inclusion vector is a subset of another. Without loss of generality, each component function $f_h$ mean be assumed to have mean zero with respect to some underlying probability measure $Q$ on $\mathcal{X}$.

Minimax rates of estimation have recently been established for the additive-interactive framework, along with the existence of a GP regression model equipped

---

6 Here, $\alpha$ denotes the number of derivatives plus the Hölder coefficient of the regression function.
with appropriate variable selection priors that offer adaptive and near optimal performance in this setting (Yang and Tokdar, 2014); see Appendix C.

4.4.2 A model for additive GP regression

Under an additive-interactive assumption, the true regression function \( f_0(x) \) decomposes into the sum of \( k_0 \) irreducible components, each having smoothness \( \alpha_h \) and relying on \(|\gamma_h| \leq \bar{d}\) predictors parametrized by inclusion vector \( \gamma_{0,h} \in \{0,1\}^p \), \( 1 \leq h \leq k_0 \). Then, \( f_0(x) = \sum_{h=1}^{k_0} f_{0,h}(x_{\gamma_{0,h}}) \), and the proposed additive method uses a sum of \( k \) components functions to model \( f_0(x) \). Under a Gaussian error model, points \((x,y) \in \mathbb{R}^p \times \mathbb{R}\) are modeled as

\[
y = \sum_{h=1}^{k} f_h(x) + \epsilon, \quad \epsilon \sim \mathcal{N}(0,\sigma^2). \tag{4.7}
\]

Since \( k_0 \) is unknown, we define \( k_{\text{max}} : k_0 \leq k_{\text{max}} \) as a fixed upper bound on the number of additive components used to estimate \( f_0(x) \). Components are allowed to be empty (i.e., \(|\gamma_h| = 0\)) to allow the effective number of components be random within this bound. The set of active component indices is denoted \( \mathcal{A} : |\mathcal{A}| \leq k_{\text{max}} \) for which \( f = \sum_{h \in \mathcal{A}} f_h \). The adaptation scheme described in Section 4.4.4 effectively eliminates the need to implement a reversible jump MCMC over a trans-dimensional parameter space.

Following discussions for the GPR model in Section 4.2.1, individual component functions in (4.7) receive prior \( f_h(x) \sim \text{GP}(0,\sigma^2C) \), for rescaled squared exponential covariance function \( C = \rho^2C_{\text{SE}}(x,\tilde{x};\rho,\lambda,\gamma) = \rho^2 \exp(-\lambda^2 \|x_\gamma - \tilde{x}_\gamma\|^2) \) indexed by positive scalars \( \rho,\lambda \), and \( p\)-dimensional inclusion vector \( \gamma = (\gamma_1,\ldots,\gamma_p) \in \{0,1\}^p \) with \( x_\gamma = \{x_j : \gamma_j = 1, 1 \leq j \leq p\} \). The following hierarchical prior for aGP parameters then completes our model specification:

\[
f_h|\sigma, (\rho_h,\lambda_h,\gamma_h) \sim \text{GP}(0,(\sigma\rho_h)^2C_{\text{SE}}^\gamma(\cdot,\cdot;\gamma_h,\lambda_h)) \\
\rho_h \sim \pi_\rho, \lambda_h \sim \pi_\lambda, \gamma_h \sim \pi_\gamma, \sigma^2 \sim \pi_\sigma, \quad 1 \leq h \leq k. \tag{4.8}
\]
Careful prior specification is crucial to allow aGP to adapt to the degree of additiveness, sparsity and interaction in the high dimensional $p \gg n$ setting. Parameters $\gamma_h$ and $\lambda_h$ inserted in component covariance functions, $C_h$, facilitate selective predictor inclusion and local smoothing, encouraging “herding” of interacting predictors together. Sparse priors on component inclusion vectors regularize model complexity in terms of the number of active components, in addition to limiting the order of interaction in each. Predictors are treated exchangeably having prior inclusion probability $\tau \sim \text{Beta}(\omega, \nu)$, leading to a Beta-binomial prior (3.5) over component inclusion size as discussed in Section 3.4.1 (see Figure 3.1). Priors $\pi_p, \pi_\lambda$ improve the adaptivity of the aGP model to data by favoring smoothness, reliably borrowing information across unobserved regions of the predictor space, and leading to well-calibrated predictive intervals\(^7\) (see Figure 4.1).

4.4.3 Prior specification for aGP

The predictors and response are assumed standardized to elicit meaningful grid points and default specification for prior hyper-parameters. The noise variance is be modeled using a conjugate inverse-gamma prior, $\sigma^2 \sim \text{IG}(\nu/2, \nu s_0^2/2)$, with $s_0^2$ chosen depending on $\nu$ so that $\Pr(\sigma^2 \leq 1) = 0.95$. Choosing $\nu = 2.5$ accommodates heavier tails to allow for additional robustness to outliers while maintaining finite variance for $p(\theta | \{f_h, \rho_h, \lambda_h, \gamma_h\})$ having marginalized over $\sigma^2$.

Scale parameters of the GP covariance function (4.8) play a key role in the ability to learn lower-dimensional interaction effects in the additive-interactive framework, and aGP’s performance depends crucially on good mixing of the component inclusion vectors. However, it is well-known that coupling between the $n$-vector of GP realizations and its hyper-parameters can lead to poor mixing, and variable selection adds

\(^7\) Estimates of the regression surface $f$ will be smoothly interpolated in regions of $\mathcal{X}$ where data is sparse by borrowing information from nearby points, and having wider credible intervals.
additional layer of complexity. Consequently, poor mixing of the scale parameters can hamper MCMC efficiency if updates to $\gamma$ are rejected because the correct $(\rho, \lambda)$ scaling was not proposed. We propose a griddy-Gibbs (Ritter and Tanner, 1992) discretization of component scale parameters as an effective strategy for efficient inference over the model space. This enables marginalizing over scale parameters when sampling component inclusion vectors and avoids updating these parameters using a metropolis step which can lead to slower MCMC mixing.

Define $S_\rho$ as the support grid for signal-to-noise ratio $\rho$, and $S_\lambda$ for inverse length-scale $\lambda$. Associated prior weight vectors are denoted $W_\rho = \{ \bar{w}_i : \rho_i \in S_\rho, \sum_i \bar{w}_i = 1 \}$ and $W_\lambda = \{ \bar{w}_i : \lambda_i \in S_\lambda, \sum_i \bar{w}_i = 1 \}$, respectively. Discrete priors over scale parameters $\rho, \lambda$ are specified as

$$
\pi(\rho) = \sum_{\rho_i \in S_\rho} \bar{w}_i \delta(\rho - \rho_i), \quad w_i \propto \text{Beta}(\rho(R)|a_\rho, b_\rho)
$$

$$
\pi(\lambda) = \sum_{\lambda_i \in S_\lambda} \bar{w}_i \delta(\lambda - \lambda_i), \quad w_i \propto \text{Gamma}(\lambda^a|a_\lambda, b_\lambda)
$$

(4.9)

where $\bar{w}_i = w_i / \sum_{w_i \in S_\rho} w_i$ are the normalized weights for each scale parameter and $\delta(x-a)$ is the degenerate Dirac delta distribution at $x = a$. A careful choice for grid support $S_\rho$ and $S_\lambda$ is necessary to allow aGP to adapt to additive structures having varying degrees smoothness and signal strength.

**Discretization for signal-to-noise ratio parameter $\rho$**

The GP covariance scaling in (4.8) is parameterized as the product of noise variance $\sigma^2$ with SNR $\rho^2$. Components with smaller SNR concentrate increasingly on their prior mean, whereas those with larger values capture important variation in the response. The marginal posterior variance for a single GP component in (4.2), $R^2 := \rho^2/(1 + \rho^2)$, is therefore akin to the R-squared model fit diagnostic in least-squares regression. This provides a useful tool inducing the prior on $\rho$ by specifying our prior belief on $R^2 \in [0, 1]$. Grid values for $S_\rho$ are chosen so that $R^2$
ranges over \( \{0, 0.25, 0.50, 0.70, 0.85, 0.99\} \). Setting \( \bar{w}_0 = \Pr(\rho = 0) \) is discussed in Section 4.4.4. For prior \( R^2 \sim \text{Beta}(a_\rho, b_\rho) \), uniform weights over \( S_\rho \setminus \{0\} \) may be specified by setting \( a_\rho = b_\rho = 1 \). Final weights are obtained by re-normalizing, 
\[
\bar{w}(\rho) = (1 - \bar{w}_0) w(\rho) / \sum_{\rho \neq 0} w(\rho) \quad \text{for} \quad \rho \in S_\rho \setminus \{0\}.
\]

**Discretization for inverse length-scale parameter \( \lambda \)**

Correlation between nearby points is characterized by inverse length-scale \( \lambda \). Components with larger length-scales capture smoother patterns in the regression surface, while those with smaller length-scales characterize finer shapes and features. Weights for the discrete prior on \( \lambda \) are chosen to correspond to its associated powered gamma prior with expected prior component size \( d^* = p \mathbb{E}\{\tau|\omega, \nu\} \) (see Section 3.4.1). Let \( c(\Delta) \) denote the correlation between two \( p \)-dimensional points \( x, \tilde{x} \) at a distance of \( \Delta \). In particular, for \( \tilde{x} = x + \Delta b, \ b : \|b\|_2 = 1, \)
\[
c(\Delta) := \text{Corr}(f_h(x), f_h(\tilde{x})) = \exp(-\lambda_\Delta^2 \Delta^2). \tag{4.10}
\]

Restricting to higher values of correlation imposes regularity on the degree of smoothness of the regression surface within a local euclidean neighborhood. End points for \( S_\lambda \) are chosen such that \( \Pr(0.70 \leq c(\Delta) \leq 0.95) = 0.95 \). For \( \Delta = 0.1 \) and prior expected component size \( d^* = 1 \), one feasible solution is obtained by setting \( a_\lambda = 6 \) and \( b_\lambda = 2 \). By default, we consider \( c(\Delta) = \{0.50, 0.70, 0.80, 0.95, 0.99\} \), which from (4.10), correspond to inverse-length scales \( S_\lambda = \{8.33, 5.97, 4.72, 2.26, 1.00\} \). In particular, we note that \( \lambda \approx 1 \) characterizes a near-linear relationship for nearby points whereas larger values correspond to higher degrees of non-linearity (see Figure 4.1).

With interpretations for scale parameters \( \rho, \lambda \) as discussed, the chosen discretizations cover a range reasonable SNR and smoothness values and works well as a default specification across settings of varying signal strength, predictor correlation and dimension (see simulated and real data studies in Section 4.6.2 and 4.6.3).
4.4.4 Stochastic component adaptation

Define $k_A = \sum_{h=1}^{k_{\text{max}}} I(\max\{\rho_h, |\gamma_h| \} > 0)$ as the number of active components, where a component is deemed active whenever $\max\{\rho, |\gamma| \} > 0$. This allows for a few components with no predictors to remain active. In addition, non-empty components explaining little variation in the response will often have signal-to-noise ratio $\rho = 0$, but remain active to allow aGP to include new predictors into active components as they are found. Individual components are assumed to be sparse with $d^* \ll p$, and hence $|\gamma_h| \sim \text{Poisson}(d^*)$ with $\mathbb{E}k_A \approx k_{\text{max}}(1 - c_0)$, $c_0 = w_0 \exp(-d^*)$. The minimum and maximum number of active components are set as $k_{\text{min}} = \lfloor \log(p) \rfloor$ and $k_{\text{max}} = \lceil p^{1/2} \rceil$ in Algorithm 3 in Section 4.5.2.

In addition, define $B$ as a fixed computational budget (target likelihood evaluations) per iteration. By default, we set $B = 2k_{\text{max}}$. Active set $\mathcal{A}$ is updated by Algorithm 2 in each MCMC iteration, with baseline probability $\theta_0$ initialized so that, on average, one inactive component is “switched on,” and component inclusion vectors are updated with probability $\varrho_h = \max\{\theta_0, I(h \in \mathcal{A})\}$. As the number of active components adapt in a data dependent way, one may trade-off a large number of small scans with more detailed neighborhood scans with component-specific budgets of $M = \lfloor B/k_A \rfloor$. Such detailed scans become more effective as propensity scores are learned, enabling good paired neighborhoods to be considered; see Section 3.4.6. Setting $c_0 = 0.05$ ($w_0 = 0.14$) start aGP off with a large fraction of active components, enabling multiple attempts at identifying important predictors across the active components.
Algorithm 2 Updating active component set

**Input:** (i) Active index set $\mathcal{A}$; (ii) Parameters $\Theta = \{(\gamma_h, \rho_h, \lambda_h) : h \in \mathcal{A}\}$, $1 \leq h \leq k_{\text{max}}$

**Output:** (i) Updated active index set $\mathcal{A}$

1: function UPDATE.ACTIVESET($\Theta, \mathcal{A}$)
2: Initialize $\theta_0 = 0$
3: Compute $k_A = \sum_{h=1}^{k_{\text{max}}} I(\max\{\rho_h, |\gamma_h|\} > 0)$
4: if $k_A < k_{\text{max}}$ then
5:   set $\theta_0 = 1/(k_{\text{max}} - k_A)$
6: end if
7: for $h = 1 : k_{\text{max}}$ do
8:   $\varrho_h = \max\{\theta_0, I(h \in \mathcal{A})\}$
9:   if $(h > k_{\text{max}} - k_{\text{min}}, |\mathcal{A}| < k_{\text{min}})$ or $U(0,1) < \varrho_h$ then
10:      $\mathcal{A} \leftarrow \mathcal{A} \cup \{h\}$
11:   else
12:      $\mathcal{A} \leftarrow \mathcal{A} \setminus \{h\}$
13: end if
14: end for
15: end function

4.5 Posterior computation for aGP regression

We propose a fast, efficient, and reproducible Metropolis-within-Gibbs sampler for posterior computation in aGP model (4.8). Posterior computation is performed over the joint space of component parameters $\Theta = \{ (\rho_h, \lambda_h, \gamma_h) : h \in \mathcal{A}\}$, and sampling proceeds with sequential back-fitting updates to component-specific parameters (Friedman and Stuetzle, 1981). For each active component, the $n$-vector of GP realizations $(f_h(x_1), \ldots, f_h(x_n))^T$ is conjugate and may be analytically integrated out along with the error variance. The marginal likelihood in terms of component covariance matrices $C^{\text{SE}}_h$ (see Section 4.4.3) is given as $y|\Theta \sim t_{\nu}(0, I_n + \sum_{h \in \mathcal{A}} \rho_h^2 C^{\text{SE}}_h)$, where $t_{\nu}(\mu, \Sigma)$ denotes a $t$ distribution with $\nu$ degrees of freedom, mean $\mu$ and scaling $\Sigma$. A complete posterior computation algorithm for aGP is given in Algorithm 3, where (pMTM) sampling for each active component proceeds in three basic steps:

1. For the discrete $\mathcal{G} = S_\rho \otimes S_\lambda$ obtained by choosing $S_\rho$ and $S_\lambda$ as described in Section 4.4.3, sample component inclusion vector $\gamma_h^T \sim \pi(\cdot | \tau, \Theta_{-h}, y)$ where

$$\pi(\gamma_h | \tau, \Theta_{-h}, y) \propto \pi(\gamma_h | \tau) \sum_{(\rho, \lambda) \in \mathcal{G}} \pi(\rho, \lambda) p(y | (\rho, \lambda, \gamma_h), \Theta_{-h}). \quad (4.11)$$
Even when $p$ is small, only a tiny fraction of the possible $2^p - 1$ neighbors are “good” posterior configurations. We employ the stochastic neighborhood sampler (pMTM) developed in Section 3.4.5 to efficiently update component inclusion based on a set of “one-away” neighbors. Proposals are accepted with probability (3.12) to ensure validity of the samples drawn.

(2) Sample scale parameters $(\rho_h, \lambda_h) \in \mathcal{G}$ with weights given by

$$
\pi(\rho_h, \lambda_h | \gamma_h, \Theta_{-h}, y) = \frac{\pi(\rho_h, \lambda_h) p(y | (\rho_h, \lambda_h, \gamma_h), \Theta_{-h})}{\sum_{(\rho, \lambda) \in \mathcal{G}} \pi(\rho, \lambda) p(y | (\rho, \lambda, \gamma_h), \Theta_{-h})}
$$

(4.12)

(3) Auxiliary draws for $\sigma^2$ and $f = (f_h(x_1), \ldots, f_h(x_n))'$ may be drawn as

$$
\pi(\sigma^2 | \Theta, y) = IG(a + n/2, b + y^T \Sigma^{-1} y / 2)
$$

$$
\pi(f | \sigma^2, (\rho_h, \lambda_h, \gamma_h), \Theta_{-h}, y) = N(\Sigma_h \Sigma_{(h)}^{-1} y, \sigma^2 \Sigma_h)
$$

(4.13)

where $\Sigma_{(h)} = I_n + \sum_{s \neq h} \rho_s^2 C_s$ and $\Sigma_h = (\Sigma_{(h)}^{-1} + (\rho_h^2 C_h)^{-1})^{-1}$. In addition to being used to obtain samples from the predictive distribution at new covariate locations (see Section 4.5.3), posterior draws for variance parameter $\sigma^2$ also serve as a useful measure of MCMC mixing and stability.

4.5.1 Ensemble-wide moves

Inter-component moves (ICM) may be introduced to facilitate predictors being rearranged between active additive components. ICM moves propose updates to inclusion state $\Theta = \{(\gamma_h, \theta_h = (\rho_h, \lambda_h)) : \theta_h \in \mathcal{G}, h \in \mathcal{A}\}$ via transitions $((\gamma_m, \theta_m), (\gamma_n, \theta_n)) \rightarrow ((\gamma'_m, \theta'_m), (\gamma'_n, \theta'_n))$ between components $(m, n) \in \mathcal{A}$. These moves are complementary to the sequential updating of component inclusion vectors using the back-fitting MCMC scheme of Section 4.59. The following transitions are considered:

---

8 The function $h$ (see Section 3.4.2) chosen to score proposed neighbors to component inclusion vector $\gamma_h$ is proportional to its unnormalized posterior $\pi(\gamma | \cdot), \gamma' \in N_m(\gamma_h)$, as given in (4.11).

9 While paired neighborhood moves (pMTM) update component inclusion one at a time, adaptive learning of propensity scores (see Section 3.4.6) allows for shared learning of predictor importance across additive components, and is thus a viable alternative to the use of ICM moves.

93
(1) Paired donate: randomly choose two distinct active component indices \((m, n) \in \mathcal{A}\). Take \(\gamma_n\) to be the non-empty donor component by convention. For \(j \in \gamma_n\), define \(\tilde{\gamma}_{jm} = \{\ldots, \gamma_m + 1_j, \gamma_n - 1_j, \ldots\}\) and set \(N_{PD}(\gamma; (m, n)) = \{\tilde{\gamma}_{jm}: j \in \gamma_n\}\).

(2) Paired swap: randomly choose two distinct active and non-empty component indices \((m, n) \in \mathcal{A}\). For predictor pair \((j, k) \in S(\gamma_m, \gamma_n) = \{(p, q) \in \gamma_m \times \gamma_n: p \neq q\}\), define \(\tilde{\gamma}_{jk} = \{\ldots, \gamma_m - 1_j + 1_k, \gamma_n - 1_k + 1_j, \ldots\}\) and set \(N_{PS}(\gamma; (m, n)) = \{\tilde{\gamma}_{jk}: (j, k) \in S(\gamma_m, \gamma_n)\}\).

(3) Cross-component donate: randomly choose an active non-empty component index \(n \in \mathcal{A}\). For \(j \in \gamma_n\), define \(\tilde{\gamma}_{jm} = \{\ldots, \gamma_m + 1_j, \gamma_n - 1_j, \ldots\}\) for \(m \in \mathcal{A}\setminus\{n\}\). Set \(N_{CD}(\gamma; n) = \{\tilde{\gamma}_{jm}: j \in \gamma_n, m \neq n\}\).

Cross-component moves in (3) facilitate a rapid exchange of predictors from a selected active component to any other active component holding fixed all component scalings (i.e., transitions for \(\theta\) occur via the identity map). In contrast, the proposal neighborhood for (1) and (2) is defined over \(N_{[1]}(\gamma; (m, n)) \otimes \mathcal{G}^2\). Proposing ICM moves for aGP regression proceeds as follows:

(1) Select move \(m \in \{CD, PD, PS\}\) with probability \(w_{CD}, w_{PD},\) and \(w_{PS}\), respectively

(2) (a) If \(m = CD\) : let \(k\) be the index for the randomly selected non-empty active component. Then select \((\gamma', \theta' = \theta) \in A_m(\gamma) = N_m(\gamma; k)\) with probability \(\pi(\gamma'|-)/\sum_{\tilde{\gamma} \in A_m} \pi(\tilde{\gamma}|-)
\)

(b) If \(m \in \{PD, PS\}\) : let \((k, l)\) denote indices of the two randomly selected active components. Then select \((\gamma', \theta') \in A_m(\gamma) = N_m(\gamma; (k, l)) \times \mathcal{G}^2\) with probability \(\pi(\gamma', \theta'|-)/\sum_{(\tilde{\gamma}, \tilde{\theta}) \in A_m} \pi(\tilde{\gamma}, \tilde{\theta}|-)
\)

(3) Accept transition \(\Theta \rightarrow \Theta' = (\gamma', \theta')\) under the defined proposal distribution for
move $m$ with probability

$$
\alpha_m(\Theta, \Theta') = \min \left\{ 1, \frac{\sum_{\Theta \in \mathcal{M}(\Theta)} \pi(\hat{\Theta})}{\sum_{\Theta' \in \mathcal{M}(\Theta')} \pi(\hat{\Theta}')} \right\}.
$$

(4) If $m = \text{CD}$: update scale parameters $\Theta = \{\theta_h = (\rho_h, \lambda_h) : h \in \mathcal{A}\}$ via Gibbs transition kernel $T(\Theta, \Theta'|\gamma, y) = \prod_{h \in \mathcal{A}} \pi(\theta_h|\theta_{s<h}', \theta_{s>h}, \gamma, y)$; here, each component may be drawn successively as in (4.12).

**Remark 4.1.** Transition kernel $T(\Theta, \Theta'|\gamma, y) = \prod_{h \in \mathcal{A}} \pi(\theta_h|\theta_{s<h}', \theta_{s>h}, \gamma, y)$ has stationary distribution $\pi(\Theta|\gamma, y)$. This follows since $T$ is a Gibbs transition kernel.

**Lemma 4.2.** For state $\Theta = \{(\gamma_h, \rho_h, \lambda_h) : h \in \mathcal{A}\}$, ICM transition kernel preserves stationary distribution $\pi(\Theta|y)$.

**Proof:** See Appendix C.

### 4.5.2 Efficient low-rank model fitting

The pMTM neighborhood sampler of Section 3.4.2 is employed in the aGP sampling scheme outlined in Section 4.5. MCMC proceeds by cycling through active components $h \in \mathcal{A}$ and drawing samples from $\rho(\tilde{\gamma}_h|\rho_h, \lambda_h, \Theta_{<h}, y)$, $\tilde{\gamma}_h \sim N(\gamma_h; m), m \in \{A, R, S\}$. Recall $y|\sigma, \Theta \sim N(0, \sigma^2(I_n + \Lambda))^{10}$, $\Lambda = \sum_{h \in \mathcal{A}} \rho_h^2 C_h^{SE}$, and the log-likelihood function is

$$
\ell(\{(\rho_h, \lambda_h, \gamma_h), h \in \mathcal{A}\}) = -\frac{1}{2} \log |I_n + \Lambda| - \frac{n + \nu}{2} \log \left(1 + \frac{y'(I_n + \Lambda)^{-1}y}{\nu s_0^2}\right).
$$

A single matrix inversion involves an $O(n^3)$ computation, required for every $\tilde{\gamma} \in N_m(\gamma) \cup N_m(\gamma')$ and across $k_A$ active components—this can lead to a significant computational bottleneck even for small to moderate sample sizes. Fortunately, one

---

10 The marginal likelihood integrating over all active GP components.
can reduce this cost to $O(nr^2)$ via a low-rank incomplete Cholesky factorization of the marginal covariance. The low-rank approximation is given by
\[ PAP' \approx LL' + D \] (4.16)
where $L$ is $n \times r$ with its first $r \times r$ sub-matrix being lower triangular, $D$ is a diagonal matrix with the first $r$ entries zero and the rest are nonnegative and $P$ is a permutation or pivot matrix (see Harbrecht et al. (2012) for an overview of pivoted matrix inversion). This leads to:
\[ \begin{align*}
I_n + \Lambda &= I_n + \sum_{h \in A} \rho_h^2 C_h^{\text{SE}} \approx A + LL' \\
\end{align*} \] (4.17)
for an $n \times n$ diagonal matrix $A = I_n + D$. By the Sherman-Morrison-Woodbury identity, the inverse is written as
\[ (I_n + \Lambda)^{-1} \approx (A + LL')^{-1} \]
\[ = A^{-1} - A^{-1} L (I_r + L' A^{-1} L)^{-1} L' A^{-1} \]
\[ = A^{-1} - A^{-1} L F^{-1} L' A^{-1}, \] (4.18)
with $F = I_r + L' A^{-1} L$ which can be in terms of its Cholesky factor as $F = GG'$. In addition, $|I_n + \Lambda| \approx |A + LL'| = \det(A) \det(F)$. Then,
\[ \ell(\{\rho_h, \lambda_h, \gamma_h\}) = -\frac{1}{2} \det(A) - \det(G) - \frac{n + \nu}{2} \log \left(1 + \frac{\tilde{y}^T \tilde{y} - z^T z}{\nu s_0^2}\right), \] (4.19)
where $\tilde{y} = A^{-1} y$ and $z = G^{-1} L' \tilde{y}$. As a default, Algorithm 3 for aGP regression sets rank $r = \lfloor \min\{n, 2(\log n)^2\} \rfloor$.

4.5.3 Predictive inference

Fitted values and predictions can be made using the derived Cholesky factors in Section 4.5.2 with an additional incomplete factorization based on the augmented design matrix which includes test-points where predictions are desired. All computations thus scale linearly in the number of data points.
As before, let $\Lambda = \sum_{h \in \mathcal{A}} \rho_h^2 C_h$ denote the covariance matrix for aggregated GP $f(x) = \sum_{h \in \mathcal{A}} f_h(x)$ at the observed locations, and $C_h$ are the component correlation matrices (4.8). In addition, let $\mathbf{f} = (f(x_1), \ldots, f(x_n))'$ denote the $n$-vector of aggregated GP realizations. For a single new observation vector $x^* \in \mathbb{R}^p$, the joint distribution of $f(x^*)$ with aggregated GP realizations over the training data follows from Kolmogorov consistency for Gaussian processes, namely

$$
(f(x^*), f(x_1), \ldots, f(x_n))' | \sigma^2, \Theta \sim N(0, \sigma^2 K),
$$

$$
K = \begin{pmatrix} k^* & J \\ J' & \Lambda \end{pmatrix}
$$

(4.20)

$$
f(x^*) | \mathbf{f}, \sigma^2 \sim N(J \Lambda^{-1} f, \sigma^2 (k^* - J \Lambda^{-1} J'))
$$

with $J = \{ J_{ii} = \text{Cov}(f(x^*), f(x_i)), 1 \leq i \leq n \}$ denoting the $1 \times n$ cross-covariance matrix and $k^* = \sum_{h \in \mathcal{A}} \rho_h^2$. The implied conditional distribution enables sampling from the posterior predictive distribution for $f(x^*)$ compositionally, since

$$
\pi(f(x^*) | y) = \int \pi(f(x^*) | \mathbf{f}, \sigma^2) \pi(\mathbf{f}, \sigma^2 | \Theta, y) \pi(\Theta | y) d(\Theta, \mathbf{f}, \sigma^2).
$$

(4.21)

Posterior draws for aGP parameters $\Theta = \{(\gamma_h, \rho_h, \lambda_h), h \in \mathcal{A}\}$ obtained using Algorithm 3 are used to sample from (4.21) by (1) drawing $\sigma^2 \sim \pi(\cdot | \Theta, y)$ and $\mathbf{f} \sim \pi(\cdot | \sigma, \Theta, y)$ according to (4.13); and (2) drawing $f(x^*) \sim \pi(\cdot | \mathbf{f}, \sigma^2)$ using (4.20).

Using the final $S$ draws from a $T$ iteration Markov chain, the posterior predictive mean is approximated as

$$
\mathbb{E}(y^* | y) \approx \frac{1}{S} \sum_{t=T-S}^T f^{(t)}(x^*).
$$

(4.22)

In addition, quantiles of predictive estimates $(f^{(1)}(x^*), \ldots, f^{(S)}(x^*))'$ provide posterior credible intervals as depicted in Figure 4.4 for every $x^*$ in the test set.
4.6 Simulation and real data analysis

4.6.1 Simulated experiments

The aGP method is compared to state-of-the-art competitors in terms of predictive root-mean-squared error (RMSE) and support recovery across a variety of simulated truths. A 1000 iteration MCMC chain is run for aGP using Algorithm 3 with default initializations. Averaging across independent replicates and test functions, a non-optimized aGP implementation in R completes in $121_{18}$ minutes for $p = 1000$. Results discard the first 200 samples and thin subsequent samples by selecting every fourth draw.

Synthetic data are generated by drawing $x_{ij} \sim \text{U}(0, 1)$ and $y_i \sim \text{N}(f(x_i), 1)$, $1 \leq i \leq n = 100, 1 \leq j \leq p$. When the predictor dimension is small, a single GP prior with an ARD squared exponential kernel can identify important predictors but offers no characterization of lower dimensional interaction. Tree models can accomplish both tasks, but often suffer in terms predictive accuracy for nonlinear functions. We select two popular ensemble methods, BART (Chipman et al., 2010) and Random Forests (RF) (Breiman, 2001). The Lasso (Tibshirani, 1996) is selected as a final competitor. The NULL model reports prediction using a naïve average over training outcomes.

Variable selection and interaction recovery

<table>
<thead>
<tr>
<th>Test function</th>
<th>Expression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friedman</td>
<td>$f = 10 \sin(\pi x_1 x_2) + 10 \cos(\pi (x_3 x_4 + x_5)) + 20(x_6 - 0.5)^2 + 10x_7$</td>
</tr>
<tr>
<td>Confounded</td>
<td>$f = 10 \cos(\pi (x_1 + x_2 + x_3)) + 10 \sin(\pi (x_2 + x_4)) + 10x_5(x_1 + x_2)$</td>
</tr>
<tr>
<td>Linear regression</td>
<td>$f = 5(x_1 + \ldots + x_5) + 2(x_6 + \ldots + x_{10})$</td>
</tr>
<tr>
<td>Single component</td>
<td>$f = 10 \cos(\pi (x_1 + 5x_2))$</td>
</tr>
</tbody>
</table>

11 Runtimes are averaged over 5 independent replicates for each test function considered in Section 4.6.1. Simulations were run on an x86×64 Intel(R) Core(TM) i7-3770.
Simulated experiments in this section concern test functions in Table 4.1. The degree of nonlinearity and additivity varies substantially across these functions, but all are sparse considering the assumed data generating process and predictor dimension. The number of active aGP components is adaptively tuned using Algorithm 2, and Figure 4.6.1 displays a histograms for the number of active (and non-empty) components post burn-in. Figure 4.3 plots the cumulative variance explained as a function of the number of active components (sorted by increasing importance). Marginal variance explained by active components are given by $\rho_h^2/(1 + \sum_{h \in A} \rho_h^2)$, $h \in A$.

An interaction graph summarizes co-appearances of predictor pairs across any of aGP’s active components. Thresholding, marginal inclusion probabilities at the $(1-q/p)$-th quantile, focussing on relationships between predictors deemed to be most important. A measure for the marginal importance of the $j$-th predictor is calculated as: (1) LASSO: $|\hat{\beta}_j|$; (2) BART: $\frac{1}{T} \sum_{t=1}^{T} (n_{jt} - \bar{n}_t)/(\bar{n}_t - \bar{n}_t)$, with $\bar{n}_t = \max_j(n_{jt})$, $\bar{n}_t = \min_j(n_{jt})$, where $n_{jt}$ denotes the number of unique occurrences of the $j$-th predictor across the tree ensemble at MCMC iteration $t$; (3) RF: using ‘importance’ in the randomForest R package; and (4) aGP: $\frac{1}{S} \sum_{t>T-S} \max\{I(j \in \gamma_h(t)), h \in A\}$\textsuperscript{12}. By identifying the most important predictors with the least number of false positives (non-zero coefficients for irrelevant predictors), aGP is the clear winner in terms of support recovery on these test functions. In contrast, inclusion probabilities from BART and RF can be relatively uninformative even when $p$ is small (compare Figures 4.6.1 and C.1). In addition, aGP improves dramatically over all competitors in terms of predictive RMSE (see Section 4.6.2).

\textit{Friedman} Additive effects are well separated for the modified Friedman function (Friedman, 1991). As shown in Figure 4.6.1, the median number of non-empty

\textsuperscript{12} This estimations predictor inclusion as the fraction of post burn-in iterations containing the $j$-th predictor in any component.
Figure 4.2: Simulated test functions 1-4 for Section 4.6.1 with \( p = 1000 \). Left: number of active vs. non-empty (utilized) components, with median sizes appearing as vertical lines; Middle: marginal inclusion probabilities for the aGP model (index on log-scale); Right: an interaction graph among the most important predictors. Vertices (edges between predictor pairs) are drawn proportional to predictor importance (co-appearance across components).
components used by aGP to recover this structure is 5, two of which are used to isolate the two- and three-way interaction effects, while another two capture the univariate effects involving predictors 6 and 7.

Confounded effects Additive effects in the true function share predictor 2, while predictors 1 and 2 also appear as bilinear effects with predictor 5. The latter are weak in comparison to the other additive components. Figure 4.6.1 shows that aGP successfully recovers the (1,5) main effect, in addition to isolating predictor interaction (2,4) from the three-way (1,2,3) predictor interaction. Weaker edge weights between predictor pairs (2,5), (3,5) and (4,5) indicate such configurations were explored during MCMC, but were less persistent. Consistent with this predictor interaction recovery, aGP has 5 or fewer non-empty components 65% of the time. aGP settles on such a configuration in 6 / 10 replicate experiments. The remaining runs discover a different a local configuration mode, wherein a single component models the bilinear (1,5) main effect and the other models interaction (1,2,3,4).

Linear regression The Lasso and aGP correctly identify predictors 1-5 and 9, while aGP also identifies predictor 6 (with inclusion probability 0.33 over a single MCMC chain). aGP has dramatically fewer number of false positives as compared to the Lasso in this example (see Figures 4.6.1 and C.1). In addition, aGP uses a larger number of components to model this fully additive function in comparison to the other simulated test functions. The histogram in Figure 4.6.1 shows that aGP has 6 or more non-empty active components roughly 50% of the time, with components often modeling univariate effects for the 7 (of 10) predictors are identified.

Single component Extreme sparsity, nonlinearity and lack of additive structure make this function fairly difficult to model. However, equipped with good sparsity induc-
ing priors and component-specific scaling (Section 4.4.3) and an efficient sampling procedure (Algorithm 3), the proposed aGP method recovers the true support. As depicted in Figure 4.6.1, aGP contains 3 or fewer non-empty components in 90% of the MCMC iterations. A single component models the (1,2) interaction, while components occasionally explore less persistent joint configurations \( \gamma = \{(1,2), 1, -\} \), and \( \gamma = \{(1,2), 2, -\} \). The competitors fail to recover the support.

![Figure 4.3: A plot of the cumulative variance explained as a function of the median number of active components (sorted by increasing importance). The fraction of the marginal variance explained by an active GP component is \( \rho^2_h / (1 + \sum_{h \in A} \rho^2_h) \), \( h \in A \). Median active sizes appear as vertical lines in histograms shown in Figure 4.6.1.](image)

\[ \text{Active GP components} \]

\[ \text{Cumulative explained variance} \]

\[ \text{Friedman} \]

\[ \text{Confounded} \]

\[ \text{Linear regr.} \]

\[ \text{Single comp.} \]

4.6.2 Predictive performance

RMSE is reported on a set of 200 test observations generated for each function described in Section 4.6.1. Results are averaged over several independent replications and reported along with standard errors. aGP’s improved support and interaction recovery leads to a dramatic improvement in predictive performance and uncertainty quantification over state-of-the-art methods in low and high dimensional settings. This validates the additive-interactive modeling framework for GP regression, and confirms that our adaptive MCMC sampler using paired and inter-component moves
Table 4.2: Predictive RMSE for test functions defined in Section 4.6.1 with \( n = 100 \) and \( p = 1000 \). Standard errors over 10 replicates appear as subscripts on averaged RMSE values.

<table>
<thead>
<tr>
<th>Function</th>
<th>aGP</th>
<th>BART</th>
<th>RF</th>
<th>LASSO</th>
<th>NULL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friedman</td>
<td>1.42</td>
<td>6.81</td>
<td>6.76</td>
<td>6.18</td>
<td>7.82</td>
</tr>
<tr>
<td>Confounded</td>
<td>2.61</td>
<td>9.27</td>
<td>9.00</td>
<td>7.87</td>
<td>9.68</td>
</tr>
<tr>
<td>Linear regr.</td>
<td>1.52</td>
<td>3.38</td>
<td>3.43</td>
<td>1.75</td>
<td>3.70</td>
</tr>
<tr>
<td>Single comp.</td>
<td>1.97</td>
<td>7.07</td>
<td>7.08</td>
<td>7.23</td>
<td>7.07</td>
</tr>
</tbody>
</table>

provides reliable predictor and interaction recovery in high dimensions.

Figure 4.4 plots predicted response values against generated responses for a simulated test dataset using the modified Friedman function. Consistent with aGP’s excellent predictive performance shown in Table 4.2, values appear close to the “\( y = x \)” line. In addition, 95% credible interval bands for aGP are significantly narrower as compared to its competitors.

**Predictive comparisons for lower dimensional data**

A single GP prior with an ARD squared exponential covariance function\(^{13}\) placed on the unknown regression function can identify important predictors and has been the primary workhorse for variable selection in the GP regression setting (Neal, 1997; Williams and Rasmussen, 1996). However, it is well known that MCMC and likelihood based methods for estimation of GP-ARD parameters suffer as the predictor dimension grows. This fact, together with recent theoretical developments for GP regression using an additive-interactive framework, were two compelling drivers for the development of the aGP model in (4.8).

Data are generated with \( n = 100 \) and \( p = 50 \) for each test function, and predictive performance is summarized in Table 4.3. The ARD competitor is fit by iteratively computing MAP\(^{14}\) estimates for discretized covariance parameters until convergence

\(^{13}\) The ARD covariance function is given in (4.5).

\(^{14}\) MAP: Maximum a posteriori probability.
Table 4.3: Predictive RMSE for test functions defined in Section 4.6.1 with \( n = 100 \) and \( p = 50 \). Standard errors over 10 replicates appear as subscripts on averaged RMSE values.

<table>
<thead>
<tr>
<th>Function</th>
<th>aGP ( 0.12 )</th>
<th>ARD-MAP ( 0.27 )</th>
<th>BART ( 0.26 )</th>
<th>RF ( 0.35 )</th>
<th>LASSO ( 0.47 )</th>
<th>NULL ( 0.22 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Friedman Confounded</td>
<td>2.31 ( 0.17 )</td>
<td>4.50 ( 0.24 )</td>
<td>7.61 ( 0.58 )</td>
<td>8.11 ( 0.65 )</td>
<td>6.89 ( 0.43 )</td>
<td>9.53 ( 0.35 )</td>
</tr>
<tr>
<td>Linear regr.</td>
<td>1.36 ( 0.14 )</td>
<td>1.41 ( 0.19 )</td>
<td>1.88 ( 0.17 )</td>
<td>2.85 ( 0.04 )</td>
<td>1.33 ( 0.09 )</td>
<td>3.67 ( 0.07 )</td>
</tr>
<tr>
<td>Single comp.</td>
<td>1.96 ( 0.22 )</td>
<td>2.70 ( 0.23 )</td>
<td>7.28 ( 0.25 )</td>
<td>7.21 ( 0.18 )</td>
<td>7.44 ( 0.20 )</td>
<td>7.31 ( 0.30 )</td>
</tr>
</tbody>
</table>

is reached. aGP’s improvement over ARD-MAP indicates how modeling of additive structure enables superior estimation of the regression function even when \( p \) is small. In addition, BART and RF degrade considerably in terms of predictive performance as predictor dimension \( p \) grows whereas aGP is robust to the 20-fold increase in predictor dimension (comparing to Table 4.2).

![Figure 4.4: Hold-out predictive performance for the modified Friedman function in Section 4.6.2 for experiments with \( p = 50 \). True response values are plotted along the \( x \)-axis and predicted values along the \( y \)-axis, with point-wise 95% credible intervals overlaid. Here, Bayesian Lasso is used in lieu of the Lasso to enable reporting of credible bands.](image-url)
4.6.3 Real data illustrations

We analyze four real datasets which are commonly used in the statistics and machine learning literature, and offer a wide range of predictor effect types, from mostly additive to including high order interaction. In addition, predictor count varies from moderate to very large across these examples. This allows us to validate the aGP model (4.8) on data that may satisfy only some of the additive-interaction assumptions. The number of aGP components (and neighborhood budget for pMTM) is adaptively tuned using Algorithm 2, with BART, RF, and the Lasso taken as competitors.

Competing methods vary dramatically in terms of their relative performance (RMSE) across the four datasets as shown in Table 4.4. For Riboflavin microarray and biscuit-dough spectroscopy (Cookie) data, regularized linear regression (i.e., the Lasso) clearly gives better averaged test prediction than ensemble nonparametric methods, whereas the opposite holds for Crime and Boston housing data (i.e., BART and RF are good competitors). Note that aGP remains competitive (best or second best predictive performance) across the board, owing to its ability to accommodate varying degrees of sparsity, interaction, and additivity. Moreover, such adaptability appears robust across varying predictor dimensions and even accommodating settings with high predictor correlation (e.g., Cookie data). For each real data example, Figure C.2 displays (a) bar-plots for component smoothness parameters $\rho_h, \lambda_h, h \in A$; (b) a trace plot of posterior draws for variance parameter $\sigma^2 \sim \pi(\cdot | \Theta, y)$ in (4.13); and (c) predictor importance scores $v_j, j = 1, \ldots, p$ learned using Algorithm 2.

To understand how learning is distributed across aGP’s active components, Figure 4.5 plots the cumulative variance explained (components sorted by increasing importance) for the median-sized model over all MCMC iterations. These sizes appear as vertical dashed lines in Figure 4.6. For the Riboflavin and Crime data, several active
Table 4.4: Performance of aGP against the same by BART, RF, Lasso, and the mean prediction rule (NULL) for well known real datasets. Left: the fraction of explained variance computed as $R^2 = 1 - (\text{RMSE}/\bar{s}_y)^2$ for aGP; Right: predictive (hold-out) RMSE averaged over random test/train splits with standard errors appearing as subscripts.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>aGP $R^2$</th>
<th>aGP RMSE</th>
<th>BART RMSE</th>
<th>RF RMSE</th>
<th>LASSO RMSE</th>
<th>NULL RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boston</td>
<td>0.890.04</td>
<td>3.000.54</td>
<td>3.480.49</td>
<td>3.470.63</td>
<td>5.020.54</td>
<td>9.150.47</td>
</tr>
<tr>
<td>Crime</td>
<td>0.640.15</td>
<td><strong>421.92</strong> 70.86</td>
<td><strong>415.54</strong> 81.54</td>
<td>430.4078.50</td>
<td>458.7371.66</td>
<td>741.9291.63</td>
</tr>
<tr>
<td>Riboflavin</td>
<td>0.740.13</td>
<td><strong>0.47</strong> 0.12</td>
<td>0.540.08</td>
<td>0.560.07</td>
<td><strong>0.43</strong> 0.13</td>
<td>0.800.10</td>
</tr>
<tr>
<td>Cookie</td>
<td>0.95</td>
<td><strong>0.42</strong></td>
<td>1.55</td>
<td>1.44</td>
<td><strong>0.25</strong></td>
<td>1.98</td>
</tr>
</tbody>
</table>

components contribute to aGP’s learning, whereas learning in the Boston housing and Cookie data is primarily captured in one or two GP components. The right panel in Figure 4.6 plots posterior predictor inclusion probabilities and interactions for the aGP model. As shown, aGP identifies important predictors (only a handful in some, very many in others) and their interactions (strong interaction effects in some; mostly additive, non-interactive effects in others).

Figure 4.5: A plot of the cumulative variance explained as a function of the median number of active components (sorted by increasing importance). The fraction of the marginal variance explained by an active GP component is $\rho_h^2/(1 + \sum_{h \in A} \rho_h^2)$, $h \in A$. Median active sizes appear as vertical lines in histograms shown in Figure 4.6.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Components</th>
<th>Marginal inclusion</th>
<th>Interaction graph</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boston</td>
<td>[Graph]</td>
<td>[Graph]</td>
<td>[Graph]</td>
</tr>
<tr>
<td>Crime</td>
<td>[Graph]</td>
<td>[Graph]</td>
<td>[Graph]</td>
</tr>
<tr>
<td>Riboflavin</td>
<td>[Graph]</td>
<td>[Graph]</td>
<td>[Graph]</td>
</tr>
<tr>
<td>Cookie</td>
<td>[Graph]</td>
<td>[Graph]</td>
<td>[Graph]</td>
</tr>
</tbody>
</table>

Figure 4.6: Comparison across well known datasets. Left: number of active vs. non-empty (utilized) components, with median sizes appearing as vertical lines; Middle: marginal inclusion probabilities for the aGP model (index on log-scale); Right: an interaction graph among the most important predictors. Vertices (edges between predictor pairs) are drawn proportional to predictor importance (co-appearance across components).
4.7 Extensions and future work

Extensions to non-conjugate likelihood settings (e.g., for classification and survival analysis models) are underway. Standard data augmentation techniques enable posterior computation conditional on the imputed latent variables. As discussed in Section 4.5.2, when the GP covariance function $C(x, \tilde{x})$ is smooth (e.g., squared exponential or Matern), the $n \times n$ covariance matrix $C_n$ for the observed response is typically dense and rank deficient. Such matrices often have eigenspectra that decay rapidly, and employing an incomplete rank $r$ Cholesky factorization for $C_n$ typically provides a very good approximation while reducing the computational complexity of likelihood evaluations to $O(nr^2)$. When $n \approx 1000s$, however, most neighborhood search methods fail to scale up. The subsampling approach discussed in Section 3.7 is one promising approach which allows for low-cost and efficient parallelization of likelihood evaluations over neighborhood configurations for the pMTM sampler. Related predictive process based approaches using a set of carefully chosen knots, often employed in computer emulation studies, may also be considered (Tokdar, 2011; Banerjee et al., 2012).

One exciting avenue of future research is to extend subspace learning to the GP regression setting. Indeed, sliced inverse regression (Li, 1991; Wu et al., 2009), sufficient dimensionality reduction and related approaches (Fukumizu et al., 2004, 2009; Cook et al., 2012) are particularly attractive research directions, and developing scalable Bayesian implementations of such techniques in the Gaussian process setting remains at large.
Bayesian Tensor regression

This chapter focuses on Bayesian regression involving a scalar response and a tensor-valued predictor. Such predictors are routinely encountered in applications including imaging and neuroscience, for example. Naively vectorizing the tensor leads to a very high-dimensional vector that fails to exploit the structure of the tensor, often resulting in poor estimation and predictive performance. We develop a novel class of prior distributions on tensor structured predictors which substantially reduces dimensionality relative to vectorizing, providing a multiway analogue of vector shrinkage priors, and enabling high dimensional region selection. Strong theoretical support is established for the proposed class of multiway shrinkage priors by showing posterior distributions over the tensor parameter are consistent under fairly mild conditions. An efficient MCMC algorithm is developed for posterior computation and simulation studies demonstrate that the proposed method delivers substantial improvement over state-of-the-art methods in terms of estimation and inference on voxel-level parameters.
5.1 Introduction

An impressive set of imaging techniques including magnetic resonance imaging (mRI), functional magnetic resonance imaging (fMRI) and diffusion tensor imaging (DTI) have revolutionized numerous branches of medicine over the past two decades, and are routinely used for confirmatory diagnosis for tumors and cancer, for example, as well as in treatment and post-treatment patient care. These scans avail practitioners with vast collections of image data, spurring the need for novel statistical methods that enable the uncovering and efficient learning of rich and complex structures. Indeed, statistical genomics has been played a fundamental role the development of methods that have led to the advances in mapping of diseases and genetic mutations in the ever increasing size of DNA and sequencing data. An important problem that has received increasing attention in the recent years is using such datasets to identify brain regions associated with a clinical outcome or to predict a cognitive outcome (Lindquist et al., 2008; Lazar, 2008; Hinrichs et al., 2009; Ryali et al., 2010).

Existing approaches for quantifying associations between a clinical outcome and an image predictor divide largely into two groups. The first approach proposes performing multiple parametric tests to study the effect of each voxel on the response by rendering a p-value ‘map’ across voxels (Lazar, 2008). This is clearly unsatisfactory as it ignores associations between voxels and their joint influence on the clinical outcome, while additionally relying on a mechanism to control some notion of error (e.g., multiplicity adjusted Type-I error, False discovery rate), potentially causing serious degradation in statistical power. The second casts the problem in a regression framework with known clinical outcome(s) as the response, taking the brain image along with known environmental and genetic factors (confounders) as predictors. The latter allows for a tremendous flexibility in modeling dependence, sparsity, and borrowing information to better estimate parameters and derive meaningful inference.
The standard generalized linear model (Hastie et al., 1986) assumes vector valued predictors, however, image data is most often a tensor structured predictor. Naive vectorization of this multi-dimensional array returns us to the classical modeling framework, albeit with a number of undesirable consequences. First, vectorization fails to preserve spatial structure in the images which carries crucial information that should be incorporated in the statistical method so as to enable efficient learning. For a brain image, for example, regions of the brain associated with explaining variation in the clinical outcome among patients are likely sparse (i.e., few important “active” voxel regions); and second, vectorization leads to the addition of an exponentially large number of model parameters. Typical tensor images have margins of dimension $p_j \in \{64, 128, 256\}$, $1 \leq j \leq D$, leading to $\prod_{j=1}^{D} p_j$ additional parameters (one for each voxel). The ultra-high dimensionality quickly overhelms state-of-the-art penalized regression methods, even for studies with many 100s of subjects. Alternative approaches within the regression framework include

1. functional regression using a chosen set of basis functions: Reiss and Odgen (2010) extend this for a 2D tensor predictor, however, extensions to 3D images prove difficult due to the increase in dimension and multi-collinearity among individual voxels

2. two-stage procedures: such methods first conduct a dimension reduction step (e.g., principal component analysis (PCA)), subsequently fitting a model using the lower dimensional predictors (Caffo, 2010). An obvious disadvantage of this approach is that PCA is an unsupervised dimension reduction technique, hence the resulting principal components are not guaranteed to be informative about the response. Additionally, PCA is only applicable in the 2D setting as there is no analogue for higher dimensional arrays.

Zhou et al. (2013) propose extending the generalized linear regression model to include a tensor structured parameter corresponding to the measured tensor predic-
tor. To circumvent difficulties with extensions to higher order tensor predictors, they impose additional structure on the tensor parameter, assuming it to decompose as a rank-\(R\) parafac sum (see Section 5.2.1). This massively reduces the effective number of parameters to be estimated. They develop a penalized likelihood approach where adaptive lasso penalties may be imposed on individual margins of the parafac decomposition, focusing on good point estimation for the tensor parameter. However, their method relies heavily on cross-validated methods for selecting tuning parameters, with choices for these parameters being sensitive to the tensor dimension, the signal-to-noise ratio (degree of sparsity) and the parafac rank.

Of practical interest to practitioners is a “self calibrating” procedure which adapts the complexity of the model to the data. We propose a principled method to effectively shrink unimportant voxel coefficients to zero while maintaining accuracy in estimating important voxel coefficients. Our framework gives rise to the task of model-based rank selection, with carefully constructed shrinkage priors that naturally induce sparsity within and across ranks for optimal region selection. In addition, the need for valid measures of uncertainty on parameter (predictive) estimates is crucial, especially in settings with low or moderate sample sizes, which naturally motivates our Bayesian approach. Our approach differs from image reconstruction literature as we do not assume a distribution on \(X\) (Qiu, 2007). It also differs significantly from the density estimation and unsupervised learning literature where the response to be modeled is an array/tensor (Dunson and Xing, 2009; Bhattacharya and Dunson, 2012).

### 5.2 Tensor regression

#### 5.2.1 Basic notation

Let \(\beta_1 = (\beta_{11}, \ldots, \beta_{1p_1})'\) and \(\beta_2 = (\beta_{21}, \ldots, \beta_{2p_2})'\) be vectors of length \(p_1\) and \(p_2\), respectively. The vector outer product \(\beta_1 \circ \beta_2\) is a \(p_1 \times p_2\) array with \((i, j)\)-th entry...
A $D$-way outer product between vectors $\beta_j = (\beta_{j1}, \ldots, \beta_{jp})$, $1 \leq j \leq D$, is a $p_1 \times \cdots \times p_D$ multi-dimensional array denoted $B = \beta_1 \circ \beta_2 \circ \cdots \circ \beta_D$ with entries $(B)_{i_1, \ldots, i_D} = \prod_{j=1}^D \beta_{ji}$. Define a vec$(B)$ operator as stacking elements of this $D$-way tensor into a column vector of length $\prod_{j=1}^D p_j$. From the definition of outer products, it is easy to see that vec$(\beta_1 \circ \beta_2 \circ \cdots \circ \beta_D) = \beta_D \otimes \cdots \otimes \beta_1$. A $D$-way tensor $B \in \bigotimes_{j=1}^D \mathbb{R}^{p_j}$ has a Tucker decomposition if it can be expressed as

$$B = \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \cdots \sum_{r_D=1}^{R_D} \lambda_{r_1, \ldots, r_D} \beta_1^{(r_1)} \circ \beta_2^{(r_2)} \circ \cdots \circ \beta_D^{(r_D)}$$

(5.1)

where $\beta_j^{(r_j)}$ is a $p_j$ dimensional vector, $1 \leq j \leq D$, and $\Lambda = (\lambda_{r_1, \ldots, r_D})_{r_1=1}^{R_1}, \ldots, r_D=1$ is referred to as the core tensor. If one considers $\{\beta_j^{(r_j)}; 1 \leq r_j \leq R_j, 1 \leq j \leq D\}$ as “factor loadings” and $\lambda_{r_1, \ldots, r_D}$ to be the corresponding coefficients, then the Tucker decomposition may be thought of as a multiway analogue to factor modeling.

A rank-$R$ parafac decomposition emerges as a special case of Tucker decomposition (5.1) when $R_1 = R_2 = \cdots = R_D = R$ and $\lambda_{r_1, \ldots, r_D} = I(r_1 = r_2 = \cdots = r_D)$. In particular, $B \in \bigotimes_{j=1}^D \mathbb{R}^{p_j}$ assumes a rank-$R$ parafac decomposition if

$$B = \sum_{r=1}^R \beta_1^{(r)} \circ \cdots \circ \beta_D^{(r)}$$

(5.2)

where $\beta_j^{(r)}$ is a $p_j$ dimensional column vector as before, for $1 \leq j \leq D$ and $1 \leq r \leq R$. Terminology refers to these vectors as ‘margins’ of a particular rank. The parafac decomposition is generally preferred in most modeling applications, both in terms of interpretability (i.e., invariance to the order of summation) and from a computational tractability point of view.

### 5.2.2 Model framework

Let $y \in \mathcal{Y}$ denote a subject-specific observe response (e.g., a clinical outcome), with $z \in \mathcal{X} \subset \mathbb{R}^p$ and $X \in \bigotimes_{j=1}^D \mathbb{R}^{p_j}$ as measured scalar and tensor predictors, respectively.
We assume response $y$ follows an exponential family distribution

$$f(y|\theta, \tau) = \exp \left( \frac{y\theta - b(\theta)}{a(\tau)} + c(y, \tau) \right)$$  \hspace{1cm} (5.3)$$

with natural parameter $\theta$, dispersion $\tau > 0$ and known function $a(\tau)$, $b(\theta)$ and $c(y, \tau)$.

In absence of a measured tensor predictor, one models conditional expectation in the standard GLM way, with $g(\mathbb{E}(y|z)) = \alpha + z'\gamma$, for a strictly increasing canonical link function $g(\cdot)$ and model parameters $\alpha \in \mathbb{R}$, $\gamma \in \mathbb{R}^p$. Otherwise, we model

$$g(\mathbb{E}(y|z, X)) = \alpha + z'\gamma + \langle X, B \rangle, \quad \langle X, B \rangle = \text{vec}(X)'\text{vec}(B)$$  \hspace{1cm} (5.4)$$

where $B \in \otimes_{j=1}^{D} \mathbb{R}^{p_j}$ is the tensor parameter corresponding to measured tensor predictor $X$. For concreteness, we focus on the linear regression setting where $g$ is taken as the identity link.

As discussed in Section 5.1, naive vectorization of tensor predictor $X$ requires introducing $\prod_{j=1}^{D} p_j$ additional model parameters (one for each voxel) resulting in an ultra-high dimensional predictor, and fails to incorporate spatial information into the estimation procedure. This necessitates imposing a sufficiently expressive structure on $B$ which simultaneously achieves a large dimensionality reduction. A rank-1 parafac decomposition assumes $B = \beta_1 \otimes \cdots \otimes \beta_D$ and $\text{vec}(B) = \beta_D \otimes \cdots \otimes \beta_1$. This reduces to modeling $g(\mathbb{E}(y|z, X)) = \alpha + z'\gamma + \beta_j^T X \beta_2$ when $D = 2$, corresponding to the bilinear model considered in Li, et al. (2010) and later by Huang and Wang (2011). Since only the single parameter vector $\beta_j$ is able to capture signal along the $j$-th dimension, a rank-1 assumption severely limits the flexibility of estimating any kind of irregular tensor predictor.

Following Zhou et al. (2013), we resort to the more flexible rank-$R$ parafac decomposition for tensor parameter $B = \sum_{r=1}^{R} \beta_1^{(r)} \otimes \cdots \otimes \beta_D^{(r)}$ introduced in (5.2) with
\( \beta_{j}^{(r)} \in \mathbb{R}^{p_j}, \ 1 \leq j \leq D, \text{ and } 1 \leq r \leq R. \) Expression (5.4) then becomes

\[
g(\mathbb{E}(y|z, X)) = \alpha + z'\gamma + \langle X, \sum_{r=1}^{R} \beta_{1}^{(r)} \circ \cdots \circ \beta_{D}^{(r)} \rangle
\]

\[
= \alpha + z'\gamma + \sum_{(i_1, \ldots, i_D)} (X)_{i_1, \ldots, i_D} (B)_{i_1, \ldots, i_D}
\]

where voxel \((X)_{i_1,\ldots,i_D}\) of the tensor predictor has corresponding parameter

\[
(B)_{i_1,\ldots,i_D} = \sum_{r=1}^{R} \prod_{j=1}^{D} \beta_{j,i_j}^{(r)}, \quad (i_1, \ldots, i_D) \in \mathcal{V}_B = \otimes_{j=1}^{D} \{1, \ldots, p_j\}.
\]

The model is therefore nonlinear in the parameters defining \(B\). A hierarchical specification is completed by placing appropriate priors over unknown model parameters. Default priors may be chosen for \(\alpha\) and \(\gamma\), although specification of the prior over the tensor parameter \(B\) is nontrivial and requires special attention; see Sections 5.3.2 and 5.4.

Under the assumed rank-\(R\) decomposition for \(B\), model (5.5) requires estimating \(p + R \sum_{j=1}^{D} p_j\) as opposed to \(p + \prod_{j=1}^{D} p_j\) parameters for the unstructured vectorized (saturated) model. Naturally, one wonders whether such a dramatic dimension reduction retains sufficient flexibility so as to recover accurate estimates of the true tensor parameter. This is understood by verifying if shapes corresponding to regions of interest for tensor parameter in imaging applications are reconstructed well under our proposed model. The answer from extensive simulation studies in Section 5.6 and theoretical analysis outlined in Section 5.7 is in the affirmative.

### 5.2.3 Model identifiability

From model (5.5) it is clear that only voxel-level coefficients are identified and not the individual tensor margins defining their product-sum given in (5.6). In the tensor setting, identifiability restrictions are understood in light of the following indeterminacies:
1. **Scale indeterminacy**: for each \( r = 1, \ldots, R \), define \( \lambda_r = (\lambda_{1r}, \ldots, \lambda_{Dr}) \) such that \( \prod_{j=1}^D \lambda_{jr} = 1 \). Then replacing \( \beta_j^{(r)} \) by \( \lambda_j \beta_j^{(r)} \) leaves the tensor parameter \( B \) unaltered.

2. **Permutation indeterminacy**: \( \sum_{r=1}^R \circ_{j=1}^R \beta_j^{(r)} = \sum_{r=1}^R \circ_{j=1}^R \beta_j^{(P(r))} \) for any permutation \( P(\cdot) \) of \( \{1, 2, \ldots, R\} \). In particular, this implies that \( \circ_{j=1}^R \beta_j^{(r)} \) are not identifiable for \( r = 1, \ldots, R \).

3. **Orthogonal transformation indeterminacy** (\( D = 2 \) only): for any orthonormal matrix \( O \), one has \( (\beta_1^{(r)} O) \circ (\beta_2^{(r)} O) = \beta_1^{(r)} \otimes \beta_2^{(r)} \).

For \( D > 2 \), imposing the following \((D - 1)R\) constraints ensures identifiability of the margin parameters comprising the rank-R parafac decomposition:

\[
\beta_{j,1}^{(r)} = 1, \ 1 \leq j < D, \ 1 \leq r \leq R, \quad \text{and} \quad \beta_{D,1}^{(1)} > \cdots > \beta_{D,1}^{(R)}. \quad (5.7)
\]

In the context of our proposed Bayesian method, we seek consistent estimation of \( B \) along with state-of-the-art predictive performance. Neither of these rely on identifiability of the tensor margins, \( \beta_j^{(r)} \), and hence priors placed over these parameters are not subject to such identifiability considerations. Nevertheless, model-fitting can be sensitive to the chosen parameterization and as such, due consideration is taken in proposing priors over the tensor parameter in Section 5.4, as well as in devising an efficient MCMC algorithm for posterior computation.

### 5.3 Bayesian Tensor regression

#### 5.3.1 Shrinkage priors

Although the spike-slab prior for selective predictor inclusion (3.1) possesses attractive theoretical properties and is conceptually straightforward, intractability of exploring an exponentially large space of predictor inclusion along with the belief that many regression coefficients may be small rather than exactly zero has led to
considerable growth in the appeal for continuous shrinkage priors. An impressive variety of shrinkage priors have emerged in recent years (Park and Casella, 2008; Carvalho et al., 2010b; Armagan et al., 2013; Bhattacharya et al., 2014) and references therein, and it has become an somewhat of an art to design priors that sharply shrink small ‘nuisance signals’ near zero, while having sufficiently heavy tails so as to prevent undesirable attenuation (bias) in the estimation of nonzero coefficients. In fact, Polson and Scott (2012) showed that many of the recently devised continuous shrinkage priors can be subsumed under the class of ‘global-local’ (GL) shrinkage priors which in standard form, can be expressed as a scale-mixture of Gaussians. In particular, a shrinkage prior on parameter vector \( \theta = (\theta_1, \ldots, \theta_p) \) is of the form

\[
\theta_j \sim N(0, \psi_j \tau), \quad \psi_j \sim g, \quad \tau \sim h.
\]

The global-local family of prior distributions (5.8) is widely used and extremely versatile, in part owing to the computational tractability afforded by their representation as a scale-mixture of Gaussians. However, Bhattacharya et al. (2014), note that suboptimal shrinkage for several of the recently developed GL priors can occur, owing to insufficient concentration of the induced global scale-mixture distribution \( \theta_j | \tau \sim \mathcal{K}(\cdot, \tau) \) in a neighborhood around zero. Whereas a majority of these priors focus on inducing overall marginal behavior similar to the point-mass mixture prior (3.1), they propose a new class of priors which aims to resemble the joint distribution under a two-component mixture prior by considering

\[
\theta_j \sim \mathcal{K}(\cdot, \phi_j \tau), \quad \phi_j \sim \text{Dirichlet}({\{a_j\}}), \quad \tau \sim h.
\]

The proposed Dirichlet-Laplace prior results by choosing \( \mathcal{K} = \text{DE}(\cdot | \phi_j \tau), \ 1 \leq j \leq p. \) Of course one requires that the induced marginal prior \( \theta_j | \tau \) is exchangeable, which

\footnote{Some recent developments offer various adaptive MCMC schemes that appear to offer promise in terms of alleviating some of the intractability posed by the curse of dimensionality with a growing predictor dimension; see Chapter 3.}
by symmetry, is satisfied having chosen a Dirichlet distribution over the collection of local scale parameters \( \{ \phi_j \} \).

### 5.3.2 Criteria for a multiway prior

We propose a new class of multiway shrinkage priors in the generalized linear model setting with tensor valued predictors. Assuming the tensor parameter \( B \) admits a rank-\( R \) parafac decomposition, model (5.5) results in voxel-level coefficients that are a nonlinear function of the corresponding tensor margin parameters. Moreover, this implies simultaneous shrinkage on each of the \( \prod_{j=1}^{D} p_j \) voxel coefficients as imposed by the prior over \( R \sum_{j=1}^{D} p_j \) parameters. This necessitates careful prior specification on the tensor margins \( \{ \beta^{(r)}_j; 1 \leq j \leq D, 1 \leq r \leq R \} \) such that the induced voxel-level prior has adequate tails so as to prevent over shrinkage.

Motivated by the flexibility of GL priors for ordinary high dimensional regression and recent developments within this setting, we propose a new class of multiway shrinkage priors for state-of-the-art shrinkage in setting where model parameters to be estimated arise from a rank-\( R \) parafac decomposition, namely, \( B_{i_1,\ldots,i_D} = \sum_{r=1}^{R} \prod_{j=1}^{D} \beta^{(r)}_{j,i_j} \), with shrinkage induced through coefficients \( \beta^{(r)}_{j,i_j} \) in this expression, \( 1 \leq i_j \leq p_j \). Given the assumed structure of voxel-level coefficients, a valid prior specification must ensure that all such parameters have the same distribution a priori.

In particular, priors over vector margin parameters \( \beta^{(r)}_{j} \) will

1. ensure that for each \( r = 1, \ldots, R: \) \( (\beta^{(r)}_{1,i_1}, \ldots, \beta^{(r)}_{D,i_D}) \) and \( (\beta^{(r)}_{1,k_1}, \ldots, \beta^{(r)}_{D,k_D}) \) are equal in distribution, for any \( (i_1, \ldots, i_D), (k_1, \ldots, k_D) \in \mathcal{V}_B \times \mathcal{V}_B \) and \( (i_1, \ldots, i_D) \neq (k_1, \ldots, k_D) \)

2. encode shrinkage across ranks to encourage a parsimonious decomposition, where unnecessary ranks may be effectively “turned off” as the model adapts to the complexity and signal in the data. Additionally, within-rank structure shall
be incorporated to allow margin-specific shrinkage to enhance the method’s ability to reconstruct irregular regions (shapes) in the tensor parameter.

Finally, it is crucial that the proposed multiway way prior lead to efficient and reliable model fitting (e.g., posterior computation). We propose two different multiway shrinkage prior in the following section. To the best of our knowledge, this is the first work to propose and incorporate a richly structured shrinkage prior within a regression setting with tensor valued predictors.

5.4 Multiway shrinkage priors

There are potentially many ways of specifying priors over tensor margins \( \beta^{(r)} \), which satisfy the criteria outlined in Section 5.3.2. The proposed M-MGP (multiway multiplicative gamma process) and M-DGDP (multiway Dirichlet generalized double pareto) priors are two illustrations of the multiway shrinkage prior, but achieve shrinkage within (and across) ranks in very different ways. More specifically, the M-DGDP prior induces shrinkage across ranks in an exchangeable way, whereas the M-MGP prior (5.9) has an exchangeable within-rank prior for voxel-level parameters (per the multiway criteria) but implements shrinkage across ranks by inducing a stochastic ordering of the rank-specific scaling.

5.4.1 The multiway MGP prior

We consider the simplest global-local prior independently on each of the \( D \) margins for a given rank \( r \), namely

\[
\beta^{(r)}_j \sim N(0, (\omega_{jr}, \tau_r)^{-1}I_{p_j})
\]

\[
\omega_{jr} \sim \text{Ga}(a_\omega, b_\omega)
\]

with margin elements \( \beta^{(r)}_{j,k} \), \( 1 \leq k \leq p_j \), conditionally independently given shared margin-specific scale \( \omega_{jr} \), \( 1 \leq j \leq D \), which is given a conjugate gamma prior.
These act as local adjustments to the rank-specific (global) scaling, allowing for a
greater degree of heterogeneity within margins comprising each rank-1 outer product.

Rewriting voxel-level coefficient in (5.6) as \( (B)_{i_1,\ldots,i_D} = \sum_{r=1}^{R} \tau_r^{-D} \prod_{j=1}^{D} \tilde{\beta}_{j,i_j}^{(r)} \), \( \tilde{\beta}_{j,i_j}^{(r)} \sim N(0, \omega_{j,r}^{-1}) \), it is natural to consider a sparsity inducing prior across terms of this sum so as to favor a sparse representation. One way to do so is to parametrize rank weights for constituent terms of this sum in terms of a truncated Dirichlet process. For \( r = 1, \ldots, R \), this draws \( S_r \sim \text{Beta}(1, \alpha_S) \) and sets

\[
\tau_r^{-D} = \begin{cases} 
S_1 & \text{if } r = 1 \\
S_r \prod_{l=1}^{r-1} (1 - S_l) & \text{if } r \leq R,
\end{cases}
\]

with \( S_R = 1 \) ensuring that \( \sum_{r=1}^{R} S_r = 1 \), with smaller values for concentration parameter \( \alpha_S \) favoring an increasingly sparse vector of scales \( (\tau_1^{-D/2}, \ldots, \tau_R^{-D/2}) \). However, conjugate (or closed-form) updating for the stick breaking weights \( S_r, 1 \leq r < R \), is not available under model (5.5). Though posterior computation for the Bayesian model can proceed via Metropolis Hastings updates for these scale parameters, this may slow down mixing efficiency considerably in the tensor regression setting.

Instead, we consider a multiplicative gamma prior (Bhattacharya and Dunson, 2011) for the rank scaling parameter \( \tau_r, 1 \leq r \leq R \), as

\[
\tau_r = \prod_{l=1}^{r} \delta_l, \quad \delta_l \sim \text{Ga}(a_r, 1). 
\]

For \( r > 1 \), choosing \( a_r > 1 \) induces a stochastic ordering on the precision for \( \beta_{j,i}^{(r)} \). In particular, \( \tau_1 \geq \cdots \geq \tau_R \), with \( B_r \) increasingly shrunk toward zero a priori, allowing the proposed prior to favor a sparse parafac decomposition of the tensor. This is in contrast to the stick-breaking construction where one has \( \tau_1 \geq \cdots \geq \tau_R \) with probability 1. Careful choices for prior hyper-parameters \( a_r, a_\omega, b_\omega \) are necessary to ensure sufficient regularization so as to achieve good performance; see Section 5.4.3.
5.4.2 The multiway Dirichlet GDP prior

We propose the multiway Dirichlet generalized double pareto (M-DGDP) prior as a second illustration of a multiway shrinkage prior. Keeping with the notation in Section 5.4.1, set $\tau_r = \phi_r \tau$ as the “global scaling” specific to rank $r = 1, \ldots, R$. Here, $\tau \sim \text{Ga}(a_r, b_r)$ and $\Phi = (\phi_1, \ldots, \phi_R) \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_R)$. In addition, define $W_{jr} = \text{diag}(w_{jr,1}, \ldots, w_{jr,p_j})$, $1 \leq j \leq D$ and $1 \leq r \leq R$ as the collection of local scale parameters. The hierarchical margin-level prior is given by

$$
\beta_j^{(r)} \sim \mathcal{N}(0, (\phi_r \tau)W_{jr})
$$

$$
w_{jr,k} \sim \text{Exp}(\lambda_{jr}^2/2), \quad \lambda_{jr} \sim \text{Ga}(a_{\lambda}, b_{\lambda}).
$$

By parameterizing the rank-specific scales in terms of a vector of weights drawn from a Dirichlet distribution, prior (5.10) induces shrinkage across these scales in an exchangeable way. Figure 5.1 plots realizations of random draws from the Dirichlet distribution when $R = 3$ for different values of the concentration parameter$^2 \alpha$. As $\alpha \to 0$, points increasingly tend to concentrate around vertices of the $\mathcal{S}^{R-1}$ probability simplex, thus leading to increasingly sparse realizations$^3$. In terms of prior (5.10), this has the effect of shrinking an increasing number of rank-specific scales toward zero and encouraging a more parsimonious decomposition for the tensor parameter. As discussed, careful calibration and learning of this key hyper-parameter plays a crucial role in M-DGDP’s ability to learn and favor sparse representations of tensor parameter $B$. Further details are given in Sections 5.4.3 and 5.5.

Additional flexibility in estimating $B_r = \{\beta_j^{(r)}; 1 \leq j \leq D\}$ is accommodated by modeling heterogeneity within margins via element-specific scaling $\omega_{jr,k}$. A common rate parameter $\lambda_{jr}$ encourages sharing of information between the margin elements so as to avoid overparametrization. Collapsing over the element-specific scales,

---

$^2$ Here, we assume for simplicity $\alpha_1 = \cdots = \alpha_R = \alpha$.

$^3$ This observation generalizes for any $R > 1$. 

121
\( \beta_{j,k}^{(c)} | \lambda_{jr}, \phi_r, \tau \sim \text{DE}(\lambda_{jr}/\sqrt{\phi_r \tau}), 1 \leq k \leq p_j, \) otherwise known as the Laplace distribution\(^4\). In particular, prior (5.10) leads to a GDP prior (Armagan et al., 2013) on the individual margin coefficients.

### 5.4.3 Prior hyper-parameter elicitation

When the conditional prior distributions on margins of the parafac decomposition are normal (as in the two proposed multiway priors), the distribution of the voxel-level coefficients (5.6) is not available in closed form. When \( D = 2 \), voxel \((B)_{i_1,i_2} = \sum_{r=1}^{R} (B_r)_{i_1,i_2}, (B_r)_{i_1,i_2} = \beta_{1,i_1}^{(r)} \beta_{2,i_2}^{(r)} \) and

\[
\beta_{1,i_1}^{(r)} \beta_{2,i_2}^{(r)} \sim f(\cdot|\cdot) = \frac{K_0(|\cdot|)}{\pi \sigma_{1,i_1}^{(r)} \sigma_{2,i_2}^{(r)}}
\]

where \( K_n(\cdot) \) is the modified Bessel function of the second kind and \( \sigma_{j,i_j}^{(r)} \) are variances for margin-element \((j, i_j)\). For \( D = 3 \), an analogous expression for the distribution of \( \prod_{j=1}^{D} \beta_{j,i_j}^{(r)} \) may be given in terms of the so called Meijer G-functions.

Hyper-parameters in each of the proposed multiway priors play an important role in determining the amount of sparsity induced within the margins of each rank as well as governing the effective number of ranks utilized to estimate true tensor

---

\(^4\) The double exponential (Laplace) density is \( f(x; \xi) = \xi/2 \exp (-|x|/\xi) \).
parameter $B_0$. Lower and upper bounds on variance for the voxel-level coefficients $B_{i_1,\ldots,i_D}, (i_1,\ldots,i_D) \in \mathcal{V}_B$, are derived and turn out to be informative on default choices for these parameters.

\textbf{M-MGP}

The voxel-level variance under M-MGP prior (5.9) is given by

$$\text{var}(B_{i_1,\ldots,i_D}) = \mathbb{E}\left(\text{var}\left\{\sum_{r=1}^{R} \prod_{j=1}^{D} \beta_{j,r}^{(r)}|W, \delta_1,\ldots,\delta_R\right\}\right)$$

$$= \mathbb{E}\left(\sum_{r=1}^{R} \prod_{j=1}^{D} (\omega_{j,r}^{-1})^{-1}\right) = \sum_{r=1}^{R} \mathbb{E}(\delta_1^{-D}\cdots\delta_r^{-D}) \prod_{j=1}^{D} \mathbb{E}(\omega_{j,r}^{-1})$$

$$= \sum_{r=1}^{R} \left(\frac{b_{\omega}}{a_{\omega} - 1}\right)^D \mathbb{E}(\delta_1^{-D}\cdots\delta_r^{-D}).$$

\textbf{Lemma 5.1.} Under M-MGP shrinkage prior (5.9) with constants $C_\omega = b_{\omega}/(a_{\omega} - 1)$, $B_r(r) = \prod_{l=1}^{r} (\Gamma(a_l - rD)/\Gamma(a_l))^{1/r}$, $A_r(r) = \prod_{l=1}^{r} (a_l - 1)$ and $a_\omega, a_l > 1$, then the voxel-level variance is bounded below by $\sum_{r=1}^{R} (C_\omega/A_r(r))^D$ and above by $\sum_{r=1}^{R} C_\omega^D/B_r(r)$.

\textbf{Proof:} See Appendix D.2.

To obtain a valid upper bound in Lemma 5.1, we choose $a_l > rD$. In particular, since $\tau_R = \delta_1\cdots\delta_R$, we set $a_l = RD + 1$, $1 \leq l \leq R$, and $a_\omega = 2$. The lower bound on the voxel-level variance is controlled to $O(1)$ independently of the parafac rank $R$ by setting $b_{\omega}(r) = (RD)^r/\sqrt{R}$ (n.b. the hyper-parameter value varies as a function of rank index $r$).
The voxel-level variance under the M-DGDP prior (5.10) is given by

\[ \text{var}(B_{i_1,\ldots,i_D}) = \mathbb{E} \left( \text{var} \left\{ \sum_{r=1}^{R} \prod_{j=1}^{D} \beta_{j,i_d}^{(r)} \mid W, \Lambda, \Phi, \tau \right\} \right) \]

\[ = \mathbb{E}_\Phi \left( \sum_{r=1}^{R} \phi_r^D \mathbb{E}_r \left\{ \tau^D \right\} \mathbb{E}_{\Lambda,r} \left\{ \mathbb{E}_{W_r|\Lambda_r} \left( \prod_{j=1}^{D} \omega_{j,r,i_j} \right) \right\} \right) \]

\[ = \frac{\Gamma(\alpha_0 + D)}{\Gamma(\alpha_0) b_r^D} (2C_\lambda)^D \mathbb{E}_\Phi \left( \sum_{r=1}^{R} \phi_r^D \right) \]

where the last step follows from the MGF of a Gamma distributed random variable.

**Lemma 5.2.** Under M-DGDP shrinkage prior (5.10) and for \( D > 1 \), if \( \alpha_1 = \cdots = \alpha_R = c/R, c \in \mathbb{N}_+ \), and with constants \( C_\lambda = b_2^3/((a_\lambda - 1)(a_\lambda - 2)) \), \( a_\lambda > 2 \), \( A_r = \exp((D^2 - 3D)/2) \), then the voxel-level variance is bounded below by \( R\alpha_1^D(2C_\lambda/b_r)^D \) and above by \( A_r(2C_\lambda/b_r)^D \exp(\alpha_1 RD) \).

**Proof:** See Appendix D.2.

The upper bound in Lemma 5.2 grows exponentially in the parafac rank \( R \). If we choose \( \alpha = 1/R \), the Dirichlet vector will tend to concentrate around sparse vectors in \( S^{R-1} \) as discussed in Section 5.4.2. Such a choice also tightens the variance upper bound by removing the exponential dependence on \( R \). In addition, setting \( b_r = \alpha(R/v)^{1/D}, v > 0 \), allows one to control the variance lower bound to a constant independent of \( R \). This is important for allowing the induced voxel-level prior to have sufficiently large tails. For \( C_\lambda = C(a_\lambda, b_\lambda) \) with \( a_\lambda = 3 \) and \( D \geq 2 \), the prior voxel variance then satisfies

\[ v b_\lambda^{2D} \leq \text{var}(B_{i_1,\ldots,i_D}) \leq v b_\lambda^{2D} R^{D-1} \exp((D^2 - D)/2). \quad (5.11) \]
5.4.4 Prior calibration and initialization

For computational considerations outlined in Section 5.5, we adopt M-DGDP as the preferred multiway prior for the remainder of this chapter. Figures and Tables below provide views of the induced voxel-level prior for default hyper-parameter choices elicited in Section 5.4.3. Armagan and Dunson (2011) study various choices of \((a_\lambda, \zeta = b_\lambda/a_\lambda)\) that lead to desirable shrinkage properties such as Cauchy-like tail behavior for \(\beta_{j,k}^{(c)}\) while retaining Laplace-like shrinkage near zero. From (5.11), setting \(b_\lambda = 2\sqrt{a_\lambda}\) appears to avoid overly narrow prior sparsity as measured by quantiles and tail probabilities of the induced prior on \(B_{i_1,...,i_D}\).

Whereas empirical results from simulation studies across a variety of data settings in Section 5.6 reveal no strong sensitivity to choices for hyper-parameters \(a_\lambda, b_\lambda\), careful adaptation of the Dirichlet concentration parameter can lead to substantial improvements in estimation and MCMC mixing efficiency. We propose learning this parameter via griddy-Gibbs (Ritter and Tanner, 1992) by defining a grid \(A\) of values over which \(\alpha\) varies. Additional details for efficient model fitting are outlined in Section 5.5.

Figure 5.2: Induced voxel level prior distribution for default specification as a function of \(a_\lambda\), with \(b_\lambda = 2\sqrt{a_\lambda}\), \(b_\tau = \alpha R^{1/D}\) and \(\alpha = 1/R\).
Figure 5.3: Induced voxel level prior distribution for default specification for \( a_\lambda \in \{2, 3, 5\}, b_\lambda \in \{1, 3\} \), with \( b_r = \alpha R^{1/D} \) and \( \alpha = 1/R \). Here, \( R = 10 \) and \( D = 2 \); note that \( z\sqrt{a_\lambda} \in (1.18, 1.50) \) for the range of \( a_\lambda \) considered.

Table 5.1: Percentiles for \(|B_{i_1, \ldots, i_D}|\) under the M-DGDP prior with default \( a_\lambda = 3 \), \( b_\lambda = z\sqrt{a_\lambda} \), \( b_r = \alpha R^{1/D} \) and \( \alpha = 1/R \). Statistics are displayed as the dimension \( D \) of the tensor and its parafac rank decomposition \( R \) vary.

<table>
<thead>
<tr>
<th>( R )</th>
<th>5%</th>
<th>25%</th>
<th>50%</th>
<th>75%</th>
<th>95%</th>
</tr>
</thead>
<tbody>
<tr>
<td>( D = 2 )</td>
<td>1</td>
<td>0.001</td>
<td>0.011</td>
<td>0.057</td>
<td>0.254</td>
</tr>
<tr>
<td>5</td>
<td>0.004</td>
<td>0.040</td>
<td>0.164</td>
<td>0.595</td>
<td>3.332</td>
</tr>
<tr>
<td>10</td>
<td>0.005</td>
<td>0.058</td>
<td>0.237</td>
<td>0.852</td>
<td>4.635</td>
</tr>
<tr>
<td>( D = 3 )</td>
<td>1</td>
<td>0.000</td>
<td>0.001</td>
<td>0.010</td>
<td>0.072</td>
</tr>
<tr>
<td>5</td>
<td>0.000</td>
<td>0.009</td>
<td>0.061</td>
<td>0.341</td>
<td>3.382</td>
</tr>
<tr>
<td>10</td>
<td>0.001</td>
<td>0.017</td>
<td>0.111</td>
<td>0.608</td>
<td>5.996</td>
</tr>
</tbody>
</table>

5.5 Posterior computation and model fitting

In the regression setting, a measured response \( y \in \mathbb{R} \) is modeled under (5.5) with the identity link. For \( z \in \mathbb{R}^p \) and \( X \in \otimes_{j=1}^D \mathbb{R}^{p_j} \) denoting observed scalar and tensor predictors, respectively, a complete hierarchical model is given as

\[
y|X, \gamma, B, \sigma \sim N(z'\gamma + \langle X, B \rangle, \sigma^2)\]

\[
B = \sum_{r=1}^R B_r, \quad B_r = \beta_D^{(r)} \otimes \cdots \otimes \beta_1^{(r)}
\]

\[
\sigma^2 \sim \pi_\sigma, \quad \gamma \sim \pi_\gamma, \quad \beta_j^{(r)} \sim \pi_\beta
\]

with appropriate priors chosen for the unknown model parameters. The noise variance is modeled through a conjugate inverse-gamma prior, namely \( \sigma^2 \sim IG(v/2, vs_0^2/2) \)
with $s_0^2$ chosen (depending on $v$) so that $\Pr(\sigma^2 \leq 1) = 0.95$. This is done assuming the response is centered and scaled, which also removes the need to explicitly model an intercept term in (5.12); as a default choice, we set $v = 2$. Fixed effects are given conjugate normal prior $\gamma \sim \mathcal{N}(0, \sigma^2 \Sigma_0\gamma)$, with rescaled prior covariance for efficiency purposes. Finally, voxel data for the tensor predictor is standardized to have mean zero and variance 1, allowing one to assume default values for hyper-parameters of the proposed multiway priors.

5.5.1 Model fitting preliminaries

The proposed class of multiway priors (see Section 5.4) leads to straightforward posterior computation using a back-fitting Gibbs sampler for tensor regression model (5.12). Section 5.5.2 specifies an efficient sampling procedure under M-DGDP prior (5.10), with a similar algorithm (not shown) derived under M-MGP prior (5.9). Although parameter updates for the latter are available in closed-form, estimation of voxel coefficients (both zero and non-zero) does not fare as well and MCMC mixing is markedly worse. Reasons for this are two-fold: (1) updates to global shrinkage parameters $\delta_1, \ldots, \delta_R$ that define rank-specific scales $\tau_r$, $1 \leq r \leq R$, do not admit a closed-form block-update; and (2) M-DGDP uses a more flexible shrinkage prior for the local margin-specific scale parameters, whereas M-MGP uses a single margin-specific inverse-gamma prior which concentrates less near zero, thus amounting to less satisfactory shrinkage.

5.5.2 M-DGDP Gibbs sampler

For $\{(\beta_{jr}^{(r)}, \omega_{jr}; j \leq D, r \leq R), (\Phi, \tau), (\gamma, \sigma)\}$ under model (5.12) and prior (5.10), we rely on marginalization and blocking in our MCMC sampling to reduce autocorrelation, drawing from (i) $[\alpha, \Phi, \tau | \beta, W]$; (ii) $[\beta, W | \Phi, \tau, \gamma, \sigma, y]$; and (iii) $[\gamma, \sigma | \beta, y]$. Step (i) is non-trivial and we propose an efficient way to sample this block of pa-
rameters compositionally. This is *essential* for good learning and mixing under the M-DGDP prior. We first sample from the marginal distribution of Dirichlet concentration parameter, leading to an efficient update via griddy-Gibbs. Next, the rank-specific scales are updated as shown below (also see Appendix D.1). Step (ii) is sampled using a sequence of draws from full conditional distributions using a back-fitting procedure to iterate through margin-level parameters and across the ranks.

(1) Sample $[\alpha, \Phi, \tau | B, W] = [\alpha | B, W][\Phi, \tau | \alpha, B, W]$:

(a) draw $[\alpha | B, W]$: form a reference set by drawing $M$ samples from $[\Phi, \tau | \alpha_j, B, W]$ for each $\alpha_j \in A$. Set $w_{j,l} = \pi(B | \alpha_j, \Phi_l, \tau_l, W) \pi(\Phi_l, \tau_l | \alpha_j), 1 \leq l \leq M | A,$ and $p(\alpha_j | B, W) = \pi(\alpha_j) \sum_{l=1}^{M | A} w_{j,l}/M$. The approximation is made arbitrarily small by choosing $M$ large enough. Set $\tilde{\pi}_j = p(\alpha_j | B, W)/\sum_{\alpha \in A} p(\alpha | B, W)$ and draw $\alpha^* \sim \sum_{\alpha \in A} \tilde{\pi}_j \delta(\alpha - \alpha_j)$

(b) draw $[\Phi, \tau | \alpha^*, B, W] = [\tau | \Phi, B, W][\Phi | B, W]$: define $p_0 = \sum_{j=1}^D p_j$, and recall $a_\tau = \sum_{r=1}^R \alpha_\tau = R\alpha$ and $b_\tau = \alpha(R/v)^{1/D}$ (see Section 5.4.2), then

- draw $\psi_r \sim \text{giG}(\alpha - p_0/2, 2b_\tau, 2C_\tau), C_\tau = \sum_{j=1}^D \beta_{j,r}^T W_{j,r}^{-1} \beta_{j,r}$, and set $\phi_r = \psi_r/\sum_{l=1}^R \psi_l$ in parallel for $1 \leq r \leq R$
- draw $\tau \sim \text{giG}(a_\tau - Rp_0/2, 2b_\tau, 2\sum_{r=1}^R D_\tau), D_\tau = C_\tau/\phi_r$

(2) Sample $\{(\beta_j^{(r)}, \omega_{jr}, \lambda_{jr}); 1 \leq j \leq D, 1 \leq r \leq R\}|\Phi, \tau, \gamma, \sigma, y$. For $r = 1, \ldots, R$ and $j = 1, \ldots, D$, cycle over $[(\beta_j^{(r)}, \omega_{jr}, \lambda_{jr}) | \beta_j^{(r)}, B_{-r}, \Phi, \tau, \gamma, \sigma, y]$, where $\beta_j^{(r)} = \{\beta_{l,r}^{(r)}, l \neq j\}$ and $B_{-r} = B \setminus B_r$;

(a) draw $[\omega_{jr}, \lambda_{jr} | \beta_j^{(r)}, \phi_r, \tau] = [\omega_{jr} | \lambda_{jr}, \beta_j^{(r)}, \phi_r, \tau][\lambda_{jr} | \beta_j^{(r)}, \phi_r, \tau]$: sample (i) $\lambda_{jr} \sim \text{Ga}(a_\lambda + p_j, b_\lambda + ||\beta_j^{(r)}||_1/\sqrt{\phi_r \tau});$ and (ii) $\omega_{jr,k} \sim \text{giG}(\frac{1}{2}, \lambda_{jr}^2, \beta_{j,k}^{2(r)}/(\phi_r \tau))$ independently for $1 \leq k \leq p_j$
Lemma 5.3. The sampling procedure in Section 5.5.2 is a valid MCMC scheme having the correct target posterior as its stationary distribution.

Proof. This follows straightforwardly as steps (i) and (iii) are block-updates drawn compositionally from a sequence of closed-form conditional distributions. Step (ii) is a nested Gibbs sampler across margins and ranks, with repeated draws from the set of back-fitting full conditionals converging to the desired joint conditional distribution.

Hence, the overall sampling scheme has samples converging to the joint posterior distribution $\pi$ over the model parameters.

5.6 Simulation studies

To illustrate finite-sample performance of the proposed multiway priors, we show results from a simulation study with various dimensionality $(p, R)$ and define $\bar{b} = \max |B_{i_1, \ldots, i_D}|$ as a measure of signal-to-noise ratio. Throughout, set $p_j = p$, $\sigma_0^2 = 1$ and $\bar{b} = 1$ for convenience. In addition, we set $\gamma_0 = (0, \ldots, 0)$ and focus exclusively on inference for tensor parameter $B$. The following simulated setups are considered:

1. “Generated” tensor: We construct tensor parameters having rank $R_0 = \{3, 5\}$ with $p = \{64, 100\}$ and $D = 2$
2. “Ready made” tensor: We use three tensor (2D) images without generating them from a parafac decomposition with known rank.

Given the tensor parameter, 5 replicated datasets with \( n = 1000 \) are generated according to model (5.12) with voxel data \( x_{i_1,...,i_D} \sim N(0,1), 1 \leq i_j \leq p_j, 1 \leq j \leq D \). The tensor parameters considered are shown in Figure 5.4, where the magnitude of the non-zero voxels is \( \bar{b} = 1 \). Examples are chosen to demonstrate recovery of voxel-level coefficients across varying degrees of complexity (dimension, parafac rank) and sparsity (% of non-zero voxels; see Figure 5.4). The performance of our method with M-DGDP prior (5.10) is compared with the following three competitors: (i) a state-of-the-art optimization based tensor regression method (Zhou et al., 2013); (ii) the Lasso (on the vectorized tensor predictor); and (iii) a Bayesian model competitor using M-MGP prior (5.9) (in forthcoming manuscript). In each simulation, comparisons are made on the basis of (a) voxel mean squared estimation error (true non-zero, true zero, and overall); and (b) frequentist coverage (and length) of 95% credible intervals. MCMC for our method was run for 1000 iterations, with a 200 iteration burn-in and remaining samples thinned by 5. “Zhou” uses \( R = 5 \), selecting the tuning parameter over a grid of values that minimizes overall RMSE on the tensor parameter. In addition, warm starts for estimation the tensor parameter was employed using an initial run with a coarsened (16 x 16) image\(^5\).

Voxel-level RMSE reported in Table 5.2 demonstrates the clear versatility of our method (M-DGDP), out performing the state-of-the-art tensor regression method (“Zhou”) consistently across each of the six different examples. For cases where the tensor parameter corresponds to a low-rank parafac decomposition (‘R3-ex’ and ‘R5-ex’), our proposed method and “Zhou” perform best. In addition, our method has lower RMSE on both true zero and non-zero voxels. This validates empirically prior

\(^5\) On several of the simulated examples, this seems to help optimization, both in terms of runtime as well as in terms of RMSE at convergence.
(5.10) along with our suggested default hyper-parameter choices (see Section 5.4); in particular, M-DGDP effectively adapts to varying degrees of sparsity, shrinking many coefficients close to zero while accommodating superior estimation of nonzero voxels. Meanwhile, “Zhou” faces the following drawbacks: (i) the grid of penalty parameters is sensitive to tensor dimension \( (p, D) \) and the assumed parafac rank \( R \); and (ii) in some cases (e.g., for simpler tensor parameters), overall RMSE improved slightly for \( R = 3 \), though “non-zero” performance worsened dramatically (on a relative % basis). Thus the procedure for selecting tuning parameters via cross validation on a chosen diagnostic (e.g., AIC, BIC, RMSE) is ad-hoc and requires manual intervention. In contrast, our Bayesian method effectively utilizes the structured sparsity of the proposed multiway prior to trade off between estimation accuracy and model complexity in a principled manner.

Results in Table 5.3 demonstrate that M-DGDP yields credible intervals with good frequentist coverage across each of the simulated settings, both overall as well as on the true non-zero coefficients. We believe our method is (one of) the first method(s) to offer uncertainty quantification for tensor valued predictors, and that this will be of practical importance to scientists and practitioners. Finally, Table 5.4 provides evidence supporting the robustness of our method robustness to an increasing predictor dimension using two of the simulated examples. In both cases, RMSE for “Zhou” worsens considerably on the true zero coefficients\(^6\), whereas M-DGDP does not suffer at all. For the true nonzero voxels, RMSE increases for both methods as the margin dimension increases; on a relative % basis, however, “Zhou” worsens considerably more, while on an absolute scale, M-DGDP remains the clear winner.

\(^6\) This demonstrates how the penalty grid is sensitive to the tensor dimension for the ‘Zhou’ method.
Figure 5.4: Simulated data with $64 \times 64$ 2D tensor images ($p = 64$, $D = 2$). Row 1: The first two images (from left) have a rank-3 and rank-5 parafac decomposition; the third image is “regular”, although does not have a low-rank parafac decomposition. Row 2: All three images are irregular, and do not have a low-rank parafac decomposition. Sparisty (% non-zero voxels) are displayed in sub-captions.

Table 5.2: Comparison of voxel estimation as measured by root mean squared error (RMSE) for the six 2D tensor images portrayed in Figure 5.4. Results from “Zhou” (Zhou et al., 2013) use $R = 5$. By default, $R = 10$ is used in all M-DGDP runs.

<table>
<thead>
<tr>
<th></th>
<th>R3-ex</th>
<th>R5-ex</th>
<th>Shapes</th>
<th>Eagle</th>
<th>Palmtree</th>
<th>Horse</th>
</tr>
</thead>
<tbody>
<tr>
<td>$</td>
<td>\text{vox}_0</td>
<td>&gt; 0$</td>
<td>M-DGDP</td>
<td>Zhou</td>
<td>Lasso</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>\text{vox}_0</td>
<td>&gt; 0$</td>
<td>0.023$_{0.00}$</td>
<td>0.021$_{0.00}$</td>
<td>0.243$_{0.01}$</td>
<td>0.226$_{0.02}$</td>
</tr>
<tr>
<td>$</td>
<td>\text{vox}_0</td>
<td>= 0$</td>
<td>M-DGDP</td>
<td>Zhou</td>
<td>Lasso</td>
<td></td>
</tr>
<tr>
<td>$</td>
<td>\text{vox}_0</td>
<td>= 0$</td>
<td>0.011$_{0.00}$</td>
<td>0.014$_{0.00}$</td>
<td>0.071$_{0.00}$</td>
<td>0.085$_{0.00}$</td>
</tr>
<tr>
<td>Overall</td>
<td>M-DGDP</td>
<td>Zhou</td>
<td>Lasso</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Overall</td>
<td>0.013</td>
<td>0.015</td>
<td>0.093</td>
<td>0.102</td>
<td>0.131</td>
<td>0.172</td>
</tr>
</tbody>
</table>

132
Figure 5.5: Recovered images for the $64 \times 64$ 2D tensor images in Figure 5.4. By default, $R = 10$ is used in all M-DGDP runs.

Table 5.3: M-DGDP coverage statistics on generated and ready-made 2D tensor images with simulated tensor predictor data.

| $|\text{vox}_0| > 0$ coverage | R3-ex | R5-ex | Shapes | Eagle | Palmtree | Horse |
|----------------------------|-------|-------|--------|-------|----------|-------|
| 0.986_{0.02}               | 0.946_{0.02} | 0.747_{0.01} | 0.731_{0.04} | 0.677_{0.04} | 0.795_{0.02} |
| Overall coverage length    | 0.995_{0.01} | 0.970_{0.01} | 0.965_{0.00} | 0.940_{0.02} | 0.948_{0.02} | 0.927_{0.01} |
| 0.066_{0.01}               | 0.061_{0.01} | 0.290_{0.00} | 0.301_{0.03} | 0.410_{0.03} | 0.566_{0.02} |

Table 5.4: Sensitivity analysis of voxel estimation error (RMSE) as the tensor dimension increases; here $p_j = p \in \{64,100\}$ for the 2D tensor images ‘R5-ex’ and ‘Shapes’.

| $|\text{vox}_0|$ | R5-ex | Shapes |
|-----------------|-------|--------|
|                | 64    | 100    | 64    | 100    |
| M-DGDP coverage | > 0   | 0.946_{0.02} | 0.991_{0.01} | 0.747_{0.01} | 0.590_{0.06} |
| length          | > 0   | 0.061_{0.01} | 0.069_{0.01} | 0.290_{0.00} | 0.247_{0.01} |
| rmse            | > 0   | 0.021_{0.00} | 0.032_{0.01} | 0.243_{0.01} | 0.320_{0.03} |
| rmse            | = 0   | 0.014_{0.00} | 0.014_{0.00} | 0.071_{0.00} | 0.063_{0.00} |
| Zhou            | > 0   | 0.030_{0.00} | 0.369_{0.06} | 0.415_{0.03} | 0.586_{0.14} |
|                | = 0   | 0.020_{0.00} | 0.111_{0.02} | 0.127_{0.02} | 0.135_{0.02} |
5.7 Posterior consistency for Tensor regression

We establish convergence results for tensor regression model (5.12) under the following simplifying assumptions\(^7\): (i) as before, the intercept is omitted by centering the response; (ii) the error variance is assumed known, i.e., \(\sigma_0^2 = 1\); and (iii) fixed effects are known or equivalently \(\gamma = (0, \ldots, 0)\).

In studying the theoretical behavior of our method, we assume an asymptotic setting where margins \(p_{j,n}\) are growing in \(n\). Although the tensor dimension remains fixed in an practical application (e.g., neuroimaging), an increasing \(p_{j,n}\) paradigm attempts to capture the fact that tensor dimension \(\prod_j p_{j,n}\) is almost always substantially larger than the sample size. However, a growing dimension of the true tensor parameter results in a changing neighborhood around it, posing an additional theoretical challenge over the usual literature on posterior consistency\(^8\). Some recent work on posterior consistency has been developed for the increasing \(p_n\) paradigm; e.g., Armagan et al. (2013) for high dimensional linear regression, and Zhou et al. (2014) for multiway contingency tables, though neither of these extend to a tensor regression setting under the proposed class of multiway shrinkage priors.

We suppose the data generating model is the support of the model class (5.12), i.e., having true tensor parameter \(B_0^n \in \otimes_{j=1}^D \mathbb{R}^{p_{j,n}}\), error variance \(\sigma_0^2 = 1\) and that \(B_0^n\) admits a rank \(R\) decomposition, namely

\[
B_0^n = \sum_{r=1}^R \beta_{1,n}^{0(r)} \circ \cdots \circ \beta_{D,n}^{0(r)}, \quad \beta_{j,n}^{0(r)} = (\beta_{j,n,1}^{0(r)}, \ldots, \beta_{j,n,p_{j,n}}^{0(r)})' \in \mathbb{R}^{p_{j,n}}.
\]

In addition, define \(F_n, F_0^n \in \mathbb{R}^{\sum_{j=1}^D p_{j,n}}\) as the vectorized parameters generating

\(^7\) Simplifying assumptions are merely to ease notation and calculations; all results generalize in a straightforward manner.

\(^8\) In particular, one must show that the posterior assigns all such neighborhoods’ probabilities converging to one.
model and true tensor parameter $B_n$ and $B_0^n$, respectively;

$$F_n = \text{vec}(\beta^{(1)}_{1,n}, \ldots, \beta^{(R)}_{1,n}, \ldots, \beta^{(1)}_{D,n}, \ldots, \beta^{(R)}_{D,n})$$

$$F_0^n = \text{vec}(\beta^{(0)(1)}_{1,n}, \ldots, \beta^{(0)(R)}_{1,n}, \ldots, \beta^{(0)(1)}_{D,n}, \ldots, \beta^{(0)(R)}_{D,n})$$

5.7.1 Notation and framework

Define a Kulback-Leibler (KL) neighborhood around the true tensor $B_0^n$ as

$$B_n = \left\{ B_n : \frac{1}{n} \sum_{i=1}^{n} \text{KL}(f(y_i|B_0^n), f(y_i|B_n)) < \epsilon \right\}.$$

For simplicity, we will denote $\text{KL}(f(y_i|B_0^n), f(y_i|B_n))$ as $\text{KL}i$ from here onwards.

Since $\text{KL}_i = \frac{1}{2} (\langle X_i, B_0^n \rangle - \langle X_i, B_n \rangle)^2$, the KL neighborhood of radius $\epsilon$ around $B_0^n$ can be rewritten as $B_n = \left\{ B_n : \frac{1}{n} \sum_{i=1}^{n} (\langle X_i, B_0^n \rangle - \langle X_i, B_n \rangle)^2 < \epsilon \right\}$. Further let $\pi_n$ and $\Pi_n$ denote prior and posterior densities with $n$ observations, respectively, and

$$\Pi_n(B_n^c) = \frac{\int_{B_n^c} f(y_n|B_n)\pi_n(F_n)}{\int f(y_n|B_n)\pi_n(F_n)}$$

with $y_n = (y_1, \ldots, y_n)'$ and $f(y_n|B_n)$ is the density of $y_n$ under model (5.12). Posterior consistency is established by showing that

$$\Pi_n(B_n^c) \to 0 \quad \text{under } B_0^n \quad \text{a.s. as } n \to \infty. \quad (5.13)$$

5.7.2 Main result

The proof of (5.13) relies on defining an exponentially consistent sequence of tests. Proofs of Lemma 5.4, Lemma 5.5 and Theorem 5.6 are provided in a forthcoming manuscript.

**Definition** An exponentially consistent sequence of test functions $\Phi_n = I(y_n \in C_n)$ for testing $H_0 : B_n = B_0^n$ vs. $H_1 : B_n \neq B_0^n$ satisfies

$$\mathbb{E}_{B_0^n}(\Phi_n) \leq c_1 \exp(-b_1 n), \quad \sup_{B_n \in B_0^n} \mathbb{E}_{B_n}(1 - \Phi_n) \leq c_2 \exp(-b_2 n)$$

for some $c_1, c_2, b_1, b_2 > 0$. 135
**Assumption (A1)** Suppose there exist an exponentially consistent sequence of tests $\Phi_n$ for testing $H_0: B_n = B_0^n$ vs. $H_1: B_n \neq B_0^n$. Under (A1) one has

$$\Pi_n(B_n^c) = \frac{\int_{B_n^c} f(y_n | B_n) \pi_n(F_n)}{\int f(y_n | B_n) \pi_n(F_n)} = \frac{\int_{B_n^c} \frac{f(y_n | B_n)}{f(y_n | B_0^n)} \pi_n(F_n)}{\int \frac{f(y_n | B_n)}{f(y_n | B_0^n)} \pi_n(F_n)} = \frac{N}{D} \leq \Phi_n + (1 - \Phi_n) \frac{N}{D}.$$ 

The proof proceeds by showing

$$\Phi_n \to 0 \quad a.s. \quad (5.14)$$

$$\exp(bn)(1 - \Phi_n)N \to 0 \quad a.s. \quad (5.15)$$

$$\exp(\tilde{b}n)D \to \infty \quad a.s., \quad b > \tilde{b}. \quad (5.16)$$

**Lemma 5.4.** Under Assumption (A1), (5.14) and (5.15) are satisfied.

**Lemma 5.5.** Suppose $R \sum_{j=1}^D p_{j,n} = o(n)$, then (A1) is satisfied.

With the previous two Lemma (5.14) and (5.15) are verified. Hence, it remains only to show (5.16) which is done by proving Theorem 5.6.

**Theorem 5.6.** Let $\zeta_n = n^{1+\theta_3} (\rho_3 > 0), \ M_n = \frac{1}{n} \sqrt{\sum_{i=1}^n \|X_i\|^2}$. Given Lemma 5.5, for any $\epsilon > 0, \Pi_n(B_n : \frac{1}{n} \sum_{i=1}^n KL_i > \epsilon) \to 0$ a.s. under $B_0^n$, for the prior $\pi_n(B_n)$ that satisfies

$$\pi_n \left( B_n : \|B_n - B_0^n\| \leq \frac{2\eta}{3M_n \zeta_n} \right) \to \exp(-dn), \quad \text{for all large } n \quad (5.17)$$

for any $d > 0$ and $\eta < \frac{\epsilon}{32} - d$.

Theorem 5.6 provides a simple sufficient condition on the concentration of the prior around the truth $B_0^n$. In particular, Theorem 5.6 is used to verify that the
proposed multiway shrinkage priors in Section 5.4 satisfy (5.17) and are posterior consistent.

**Theorem 5.7.** M-MGP prior (5.9) yields posterior consistency under the following assumptions

(a) $H_1 n^{p_1} < M_n < H_2 n^{p_2}$

(b) $\sup_{l=1,\ldots,p_j,n} |\hat{\beta}_{j,n,l}^{(0)}| < M_1 < \infty$, for all $j = 1,\ldots,D$; $r = 1,\ldots,R$

(c) $\sum_{j=1}^D p_{j,n} = o(n)$.

**Theorem 5.8.** M-DGDP prior (5.10) yields posterior consistency under the following assumptions

(a) $H_1 n^{p_1} < M_n < H_2 n^{p_2}$

(b) $\sup_{l=1,\ldots,p_j,n} |\hat{\beta}_{j,n,l}^{(0)}| < M_1 < \infty$, for all $j = 1,\ldots,D$; $r = 1,\ldots,R$

(c) $\sum_{j=1}^D p_{j,n} \log(p_{j,n}) = o(n)$.

**Remark 5.9.** Theorem 5.7 and Theorem 5.8 require that $\sum_{j=1}^D p_{j,n}$ grows sub-linearly with sample size $n$. This justifies the use of our proposed tensor regression model for massive $\prod_{j=1}^D p_{j,n}$ even for moderately large $n$.

### 5.8 Extensions and future work

This work develops a novel class of prior distributions on tensor valued predictors which substantially reduces dimensionality relative to vectorizing, providing a multiway analogue of vector shrinkage priors, and enabling high dimensional region selection. Theoretical support is established for the proposed class of multiway shrinkage priors and an efficient MCMC algorithm was developed in the regression setting. We plan to extend methods developed here to settings where the measured response
for each subject is binary (e.g., indicator of a health outcome) or multivariate, i.e., $y = (y_1, \ldots, y_d)$.

In clinical trials and various longitudinal studies, monitoring the evolution in the predictor-response relationship (i.e., changes to the scalar and tensor parameters) is of fundamental interest. One application involves subjects receiving various treatments (e.g., chemotherapy), with tensor valued predictors corresponding to mRI (fMRI) scans obtained at regular intervals over a period of time. In such settings it is of crucial importance to monitor the progression of the disease in response to the treatment being administered. We plan to extend our method to the anova setting where $T_s$ tensor predictors are measured on for each subject $s = 1, \ldots, S$, and inferences on the probability of survival after a certain time or on the size of a tumor, for example, is of interest. Here, borrowing of information within subjects and between time points, and between subjects at a specific time (with necessary ‘alignment’ considerations for subject-specific images) can directly be incorporated into the hierarchical model and estimation procedure.
Appendix A

Bayesian Conditional density filtering (C-DF)

A.1 Additional tables and figures

<table>
<thead>
<tr>
<th>Linear regression</th>
<th>One-way Anova</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \beta_1 )</td>
<td>( \zeta_1 )</td>
</tr>
<tr>
<td>( \beta_2 )</td>
<td>( \zeta_2 )</td>
</tr>
<tr>
<td>( \beta_4 )</td>
<td>( \zeta_5 )</td>
</tr>
<tr>
<td>( \sigma^2 )</td>
<td>( \zeta_{10} )</td>
</tr>
<tr>
<td>( \mu )</td>
<td>( \tau^2 )</td>
</tr>
<tr>
<td>( \tau^2 )</td>
<td>( \sigma^2 )</td>
</tr>
</tbody>
</table>

Figure A.1: Parameter accuracy plotted over time as defined in (2.2) for the motivating examples of Section 2.4. Image 1: accuracy for representative regression coefficients \( \beta_j \) and \( \sigma^2 \) in the linear regression example. Images 2 and 3: accuracy for representative group means \( \zeta_j \), along with hierarchical parameters \( \mu \), \( \tau^2 \) and \( \sigma^2 \) for the one-way Anova model.
Table A.1: Predictive coverage for each simulation experiment in Table 2.7. Empirical 95% confidence intervals of the coverage probabilities over independent replications are also reported.

<table>
<thead>
<tr>
<th></th>
<th>Case 1*</th>
<th>Case 2*</th>
<th>Case 3*</th>
<th>Case 4*</th>
<th>Case 5</th>
<th>Case 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>VB</td>
<td>0.82</td>
<td>0.83</td>
<td>0.82</td>
<td>0.83</td>
<td>0.81</td>
<td>0.83</td>
</tr>
<tr>
<td></td>
<td>(0.81,0.84)</td>
<td>(0.76,0.87)</td>
<td>(0.81,0.85)</td>
<td>(0.75,0.87)</td>
<td>(0.76,0.86)</td>
<td>(0.80,0.85)</td>
</tr>
<tr>
<td>C-DF</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.98</td>
<td>0.97</td>
<td>0.99</td>
</tr>
<tr>
<td></td>
<td>(0.98,0.99)</td>
<td>(0.97,0.99)</td>
<td>(0.98,0.99)</td>
<td>(0.97,0.98)</td>
<td>(0.97,1)</td>
<td>(0.99,1)</td>
</tr>
<tr>
<td>SMCMC</td>
<td>0.96</td>
<td>0.93</td>
<td>0.95</td>
<td>0.93</td>
<td>0.96</td>
<td>0.95</td>
</tr>
<tr>
<td></td>
<td>(0.95,0.96)</td>
<td>(0.91,0.96)</td>
<td>(0.95,0.96)</td>
<td>(0.90,0.96)</td>
<td>(0.96,0.96)</td>
<td>(0.93,0.97)</td>
</tr>
</tbody>
</table>

Table A.2: Effective sample sizes (number of MCMC samples) required until MSPE $\leq 5$ are shown above for each simulation experiment, with results averaged over 10 independent replications.

<table>
<thead>
<tr>
<th></th>
<th>Case 1*</th>
<th>Case 2*</th>
<th>Case 3*</th>
<th>Case 4*</th>
<th>Case 5</th>
<th>Case 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>C-DF</td>
<td>2040</td>
<td>2610</td>
<td>1710</td>
<td>3630</td>
<td>5250</td>
<td>210</td>
</tr>
<tr>
<td>SMCMC</td>
<td>2940</td>
<td>3990</td>
<td>2820</td>
<td>4410</td>
<td>4440</td>
<td>1050</td>
</tr>
</tbody>
</table>

A.2  Convergence proofs

proof of Lemma 2.3

Proof. The proof follows by induction. First we prove an identity that is used in the proof that follows. Letting $A \in B(R^d)$,

$$K^{(r)}(\Theta, A) - T^{(r)}(\Theta, A) = \int \left[ K^{(r-1)}(\Theta', A) - T^{(r-1)}(\Theta', A) \right] T(\Theta, d\Theta') + \int \left[ K(\Theta, d\Theta') - T(\Theta, d\Theta') \right] K^{(r-1)}(\Theta', A).$$  \hspace{1cm} (A.1)

(A.1) relates the differences between the kernels at $r$-th iteration of the Markov chain.

Using $||\nu_1 - \nu_2||_{TV} = \sup_{g: R^d \to [0,1]} |\int g d\nu_1 - \int g d\nu_2|$ and the fact that r.h.s of (A.1) is free of $A$ yields

$$||K^{(r)}(\Theta, \cdot) - T^{(r)}(\Theta, \cdot)||_{TV} \leq \int ||K^{(r-1)}(\Theta', \cdot) - T^{(r-1)}(\Theta', \cdot)||_{TV} T(\Theta, d\Theta') + ||K(\Theta, \cdot) - T(\Theta, \cdot)||_{TV}. \hspace{1cm} (A.2)$$
Suppose (2.8) holds for \(r = 1\). Using (A.2) we find
\[
\sup_{\Theta} ||K^{(r)}(\Theta, \cdot) - T^{(r)}(\Theta, \cdot)||_{TV}
\leq \sup_{\Theta} ||K(\Theta, \cdot) - T(\Theta, \cdot)||_{TV} + \sup_{\Theta} \int ||K^{(r-1)}(\Theta', \cdot) - T^{(r-1)}(\Theta', \cdot)||_{TV} T(\Theta, d\Theta')
\leq \sup_{\Theta} ||K(\Theta, \cdot) - T(\Theta, \cdot)||_{TV} + (r - 1) \rho < \rho + (r - 1) \rho = \rho r.
\]
\[(A.3)\]

Also note that there exists \(r_0\) s.t. for all \(r > r_0\), we have \(||T^{(r)} - \mu_1||_{TV} < \frac{1}{2}||\mu_1 - \mu_2||_{TV}\) and \(||K^{(r)} - \mu_2||_{TV} < \frac{1}{2}||\mu_1 - \mu_2||_{TV}\). By the triangle inequality, for all \(r > r_0\)
\[
||K^{(r)} - T^{(r)}||_{TV} \leq ||K^{(r)} - \mu_2||_{TV} + ||\mu_1 - \mu_2||_{TV} + ||T^{(r)} - \mu_1||_{TV} < 2||\mu_1 - \mu_2||_{TV}.
\]
\[(A.4)\]

Comparing (A.3) and (A.4) the result follows.

\[\Box\]

**proof of Lemma 2.4**

*Proof.* We show that \(\int T_t(\Theta, \Theta') f_t(\Theta) d\Theta = f_t(\Theta')\) for case (1) where closed-form sampling from the two sets of full conditionals is assumed (i.e., when the C-DF kernel has the form of a Gibbs kernel, using appropriate approximating substitutions for the full conditionals in terms of SCSS). The proof for case (2) follows in an identical manner taking into account that the MH kernel \(Q(\Theta_2, \Theta_2 | \Theta_{1,t-1})\) has \(\pi_t(\Theta_2 | \Theta_{1,t-1})\)
as its stationary distribution.

\[ \int T_t(\Theta, \Theta') f_t(\Theta) d\Theta \]

\[ = \int \left[ \prod_{i=1}^{p_1} \pi_t(\Theta'_{1i} | \hat{\Theta}_{2,t-1}, \Theta'_{1l}, l < i, \Theta_{1l}, l > i) \right] \left[ \prod_{i=1}^{p_2} \pi_t(\Theta'_{2i} | \hat{\Theta}_{1,t-1}, \Theta'_{2l}, l < i, \Theta_{2l}, l > i) \right] \pi_t(\Theta_1 | \hat{\Theta}_{2,t-1}) \pi_t(\Theta_2 | \hat{\Theta}_{1,t-1}) d\Theta_1 d\Theta_2 \]

\[ = \int \left[ \prod_{i=1}^{p_1} \pi_t(\Theta'_{1i} | \hat{\Theta}_{2,t-1}, \Theta'_{1l}, l < i, \Theta_{1l}, l > i) \right] \pi_t(\Theta_1 | \hat{\Theta}_{1,t-1}) d\Theta_1 \]

\[ \times \int \left[ \prod_{i=1}^{p_2} \pi_t(\Theta'_{2i} | \hat{\Theta}_{1,t-1}, \Theta'_{2l}, l < i, \Theta_{2l}, l > i) \right] \pi_t(\Theta_2 | \hat{\Theta}_{1,t-1}) d\Theta_2 \]

\[ = \pi_t(\Theta'_{1} | \hat{\Theta}_{2,t-1}) \pi_t(\Theta'_{2} | \hat{\Theta}_{1,t-1}) . \]

The last step follows from the fact that

\[ \left[ \prod_{i=1}^{p_1} \pi_t(\Theta'_{1i} | \hat{\Theta}_{1,t-1}, \Theta'_{1l}, l < i, \Theta_{1l}, l > i) \right] \text{ and } \left[ \prod_{i=1}^{p_2} \pi_t(\Theta'_{2i} | \hat{\Theta}_{1,t-1}, \Theta'_{2l}, l < i, \Theta_{2l}, l > i) \right] \]

are the Gibbs kernels with the stationary distribution \( \pi_t(\Theta_2 | \hat{\Theta}_{1,t-1}) \) and \( \pi_t(\Theta_1 | \hat{\Theta}_{2,t-1}) \), respectively.

\[ \square \]

Proof of Lemma 2.6 The following proof builds on Theorem 3.6 from Yang and Dunson (2013). Although most of the proof coincides with their result, we present the entire proof for completeness.

**Proof.** Fix \( \epsilon \in (0, 1) \). Choose \( n_t, t \geq 1 \) s.t. \( \rho^{n_t} < \epsilon \). Using the fact that universal ergodicity condition implies uniform ergodicity, one obtains

\[ d_{TV}(T_t^{n_t}, \pi_t) \leq \alpha_t^{n_t} < \epsilon. \]

Let \( h = T_{t-1}^{n_{t-1}} \cdots T_1^{n_1} \pi_0 \), then

\[ d_{TV}(T_t^{n_t} \cdots T_1^{n_1} \pi_0, f_t) = d_{TV}(T_t^{n_t} h, f_t) \leq d_{TV}(T_t^{n_t}, f_t) d_{TV}(h, f_t) \]

\[ \leq \alpha_t^{n_t} d_{TV}(h, f_t) \leq \epsilon (d_{TV}(h, f_{t-1}) + d_{TV}(f_t, f_{t-1})). \]

(A.5)
using the result repeatedly, one obtains

$$d_{TV}(T^n_t \cdots T^n_1 \pi_0, f_t) \leq \sum_{i=1}^t \epsilon^{t+1-i}d_{TV}(f_i, f_{i-1}).$$

R.H.S clearly converges to 0 applying condition (ii). The proof is completed by using condition (iii) and the fact that

$$d_{TV}(T^n_t \cdots T^n_1 \pi_0, f_t) \leq d_{TV}(T^n_t \cdots T^n_1 \pi_0, f_t) + d_{TV}(f_t, \pi_t).$$

\[ \square \]

**proof of lemma 2.10**

*Proof.* Stationary distribution $f_t$ of the C-DF transition kernel $T_t$ is the approximate posterior distribution to $\pi_t$ obtained at time $t$, and by Lemma 2.4 is given by

$$f_t(\Theta_1, \Theta_2) = \pi_t(\Theta_1 | \hat{\Theta}_{2,t}) \pi_t(\Theta_2 | \hat{\Theta}_{1,t})$$

$$= \frac{\prod_{l=1}^t p_{\Theta_1, \hat{\Theta}_{2,t}}(D_l) \prod_{l=1}^t p_{\Theta_1, \hat{\Theta}_{2,t}}(D_l) \pi_0(\hat{\Theta}_{1,t}, \Theta_2) \pi_0(\Theta_1, \hat{\Theta}_{2,t})}{\int \prod_{l=1}^t p_{\Theta_1, \hat{\Theta}_{2,t}}(D_l) \prod_{l=1}^t p_{\Theta_1, \hat{\Theta}_{2,t}}(D_l) \pi_0(\hat{\Theta}_{1,t}, \Theta_2) \pi_0(\Theta_1, \hat{\Theta}_{2,t})}.$$

By assumption, $\hat{\Theta}_{1,t} \to \Theta_1^0$, $\hat{\Theta}_{2,t} \to \Theta_2^0$ a.s. under $\Theta^0$, there exists $\Omega_0$ which has probability 1 under the data generating law s.t. for all $\omega \in \Omega_0$, $\hat{\Theta}_{1,t}(\omega)$ and $\hat{\Theta}_{2,t}(\omega)$ are in an arbitrarily small neighborhood of $\Theta_1^0$ and $\Theta_2^0$, respectively. Also by assumption, prior $\pi_0$ is continuous at $\Theta^0$. That is, given $\epsilon > 0$ and $\eta > 0$, there exists a neighborhood $N_{\epsilon, \eta}$ s.t. for all $\Theta \in N_{\epsilon, \eta}$ one has

$$|\pi_0(\Theta_1, \Theta_2) - \pi_0(\Theta_1^0, \Theta_2^0)| < \epsilon.$$ (A.6)

Using (A.6) and the consistency of $\hat{\Theta}_{1,t}$ and $\hat{\Theta}_{2,t}$ as above, one obtains for all $t > t_0$ and $\omega \in \Omega_0$

$$|\pi_0(\Theta_1, \hat{\Theta}_{2,t}) - \pi_0(\Theta^0)| < \epsilon, \quad |\pi_0(\hat{\Theta}_{1,t}, \Theta_2) - \pi_0(\Theta^0)| < \epsilon.$$ (A.7)

Similarly, continuity of $p_{\Theta}(\cdot)$ at $\Theta^0$ leads to the condition that for all $t > t_0$,

$$|p_{\Theta_1, \Theta_2}(D_l) - p_{\Theta_1^0, \Theta_2^0}(D_l)| < \epsilon.$$ (A.8)

143
Further, consistency assumptions on \( f_t \) and \( \pi_t \) yield that for all \( t > t_1 \) and \( \omega \in \Omega_1 \)
\[
f_t(N_{e,n}|D^{(t)}(\omega)) > 1 - \eta, \quad \pi_t(N_{e,n}|D^{(t)}(\omega)) > 1 - \eta,
\]
where \( \Omega_1 \) has probability 1 under the data generating law. Considering \( \Omega = \Omega_0 \cap \Omega_1 \) and \( t_2 = \max\{t_1, t_0\} \) it is evident that \( \Omega \) has also probability 1 under the true data generating law and all of the above conditions hold for \( t > t_2 \) and \( \omega \in \Omega \). Simple algebraic manipulations yield
\[
\frac{f_t(\Theta|D^{(t)}(\omega))}{\pi_t(\Theta|D^{(t)}(\omega))} = \frac{f_t(N_{e,n}|D^{(t)}(\omega)) \prod_{l=1}^{t} p_{\theta_l}(D_l) \pi_0(\Theta)}{\pi_t(N_{e,n}|D^{(t)}(\omega)) \prod_{l=1}^{t} p_{\theta_l}(D_l) \pi_0(\Theta)}
\]
(\text{A.9})
\[
\times \left[ \prod_{l=1}^{t} p_{\theta_1,\theta_2,l}(D_l) \prod_{l=1}^{t} p_{\theta_1,\theta_2,l}(D_l) \right] \pi_0(\hat{\Theta}_{1,t},\Theta_2) \pi_0(\Theta_1,\hat{\Theta}_{2,t})
\]
\[
\leq \left( \pi_0(\Theta^0) + \epsilon \right)^2 \int_{N_{e,n}} \prod_{l=1}^{t} p_{\theta_1,\theta_2,l}(D_l) \prod_{l=1}^{t} p_{\theta_1,\theta_2,l}(D_l) \pi_0(\hat{\Theta}_{1,t},\Theta_2) \pi_0(\Theta_1,\hat{\Theta}_{2,t})
\]
\[
\leq \left( \pi_0(\Theta^0) + \epsilon \right)^2 \int_{N_{e,n}} \prod_{l=1}^{t} p_{\theta_1,\theta_2,l}(D_l) \prod_{l=1}^{t} p_{\theta_1,\theta_2,l}(D_l).
\]
Similarly,
\[
(\pi_0(\Theta^0) - \epsilon) \int_{N_{e,n}} \prod_{l=1}^{t} p_{\theta}(D_l) \leq \int_{N_{e,n}} \left[ \prod_{l=1}^{t} p_{\theta}(D_l) \right] \pi_0(\Theta) \leq (\pi_0(\Theta^0) + \epsilon) \int_{N_{e,n}} \prod_{l=1}^{t} p_{\theta}(D_l).
\]
Therefore,
\[
\frac{f_t(\Theta|D^{(t)}(\omega))}{\pi_t(\Theta|D^{(t)}(\omega))} \leq (1 - \eta)^{-1} \frac{\prod_{l=1}^{t} p_{\theta_1,\theta_2,l}(D_l) p_{\theta_1,t,\theta_2}(D_l) \prod_{l=1}^{t} p_{\theta}(D_l)}{\prod_{l=1}^{t} p_{\theta_l}(D_l) (\pi_0(\Theta^0) + \epsilon)^2}.
\]
144
Using similar calculations we have
\[
\frac{f_t(\Theta|D^{(l)}(\omega))}{\pi_t(\Theta|D^{(l)}(\omega))} \geq (1 - \eta) \frac{\prod_{l=1}^{t} p_{\Theta_1,\Theta_2,t}(D_l) \prod_{l=1}^{t} \hat{p}_{\Theta_1,t,\Theta_2}(D_l) \left( \pi_0(\Theta) - \epsilon \right)^3}{\prod_{l=1}^{t} p_{\Theta}(D_l) \prod_{l=1}^{t} \hat{p}_{\Theta}(D_l) \left( \pi_0(\Theta) + \epsilon \right)^3}.
\]
Condition (A.8) now gives us
\[
\frac{\prod_{l=1}^{t} (p_{\Theta_0}(D_l) - \epsilon)^3}{\prod_{l=1}^{t} (p_{\Theta_0}(D_l) + \epsilon)^3} \leq \frac{\prod_{l=1}^{t} p_{\Theta_1,\Theta_2,t}(D_l) \prod_{l=1}^{t} \hat{p}_{\Theta_1,t,\Theta_2}(D_l) \left( \pi_0(\Theta) - \epsilon \right)^3}{\prod_{l=1}^{t} p_{\Theta}(D_l) \prod_{l=1}^{t} \hat{p}_{\Theta}(D_l) \left( \pi_0(\Theta) + \epsilon \right)^3}.
\]
Using the condition that \( \lim_{t \to \infty} \sqrt{t} p_{\Theta_0}(D^{(l)}) \) is bounded away from 0 and \( \infty \) and choosing \( \epsilon, \eta \) sufficiently small, we have
\[
\left| \frac{f_t(\Theta|D^{(l)}(\omega))}{\pi_t(\Theta|D^{(l)}(\omega))} - 1 \right| < \kappa \quad \text{for all} \ t > t_2 \text{ and } \omega \in \Omega.
\]
Finally,
\[
\int |\pi_t(\Theta) - f_t(\Theta)| \leq \int_{N_{\epsilon,\eta}} |\pi_t(\Theta) - f_t(\Theta)| + \int_{N_{\epsilon,\eta}}^c |\pi_t(\Theta) - f_t(\Theta)| \leq \int_{N_{\epsilon,\eta}} |\pi_t(\Theta) - f_t(\Theta)| + 2\eta \\
\leq \pi_t(N_{\epsilon,\eta}) \kappa + 2\eta < \kappa + 2\eta.
\]
Appendix B

Stochastic neighborhood search for Bayesian variable selection

Stationary distribution $\pi(\gamma|\cdot)$ is abbreviated below as $\pi(\gamma)$. In addition, let $T(\gamma, \gamma') = \Pr(\gamma'|\gamma) = q(\gamma \to \gamma') \alpha(\gamma \to \gamma')$ denote the transition probability from inclusion state $\gamma$ to $\gamma'$ under a valid proposal pmf $q$ with corresponding MH acceptance ratio $\alpha$.

Proof of lemma 3.1 (pRNS)

Proof. Consider paired-move neighborhood sampler having proposal distribution (3.8) and acceptance probability (3.9). Detailed balance is verified for a paired “Add-remove” (“Remove-add” follows from symmetry, i.e., satisfied if reversibility holds) and “Swap-swap” proposals. Given component inclusion vector $\gamma$, one first chooses between Add, Remove and Swap with probabilities $w_A(|\gamma|)$, $w_R(|\gamma|)$ and $w_S(|\gamma|)$, respectively.

If $m = A$: construct one-add neighborhood $N_A(\gamma) = \{\tilde{\gamma} = \gamma + 1_j : j \notin \gamma\}$. Select $\gamma' = \gamma + 1_k$ with probability $\pi(\gamma')/\sum_{\tilde{\gamma} \in N_A(\gamma)} \pi(\tilde{\gamma})$, namely the RNS proposal distribution (3.8) restricted this set. The paired reverse one-remove neighborhood $N_R(\gamma')$ always
contains \( \gamma \). Accepting proposals with probability \( \alpha_m(\gamma, \gamma') \) given by (3.9), one has
\[
\pi(\gamma) T(\gamma, \gamma') = \pi(\gamma) w_A(|\gamma|) \frac{\pi(\gamma')}{\sum_{\hat{\gamma} \in N_A(\gamma)} \pi(\hat{\gamma})} \min \left\{ 1, \frac{w_R(|\gamma'|)}{w_A(|\gamma|)} \right\}
= \pi(\gamma) \pi(\gamma') \min \left\{ \frac{w_A(|\gamma|)}{\sum_{\hat{\gamma} \in N_A(\gamma)} \pi(\hat{\gamma})}, \frac{w_R(|\gamma'|)}{\sum_{\hat{\gamma} \in N_R(\gamma)} \pi(\hat{\gamma})} \right\}.
\]

If \( m = S \): construct one-swap neighborhood \( N_S(\gamma) = \{ \hat{\gamma} = \gamma - 1_m + 1_n : (m, n) \in \gamma \times [\gamma] \} \). Select \( \gamma' = \gamma - 1_j + 1_k \) with probability \( \pi(\gamma') / \sum_{\hat{\gamma} \in N_S(\gamma)} \pi(\hat{\gamma}) \), namely the RNS proposal distribution (3.8) restricted this set. The paired reverse swap neighborhood \( N_S(\gamma') \) always contains \( \gamma \). Accept with probability \( \alpha_m(\gamma, \gamma') \) given by (3.9). Then,
\[
\pi(\gamma) T(\gamma, \gamma') = \pi(\gamma) w_S(|\gamma|) \frac{\pi(\gamma')}{\sum_{\hat{\gamma} \in N_S(\gamma)} \pi(\hat{\gamma})} \min \left\{ 1, \frac{\sum_{\hat{\gamma} \in N_S(\gamma)} \pi(\hat{\gamma})}{\sum_{\hat{\gamma} \in N_S(\gamma')} \pi(\hat{\gamma})} \right\}
= \pi(\gamma) \pi(\gamma') \min \left\{ \frac{w_S(|\gamma|)}{\sum_{\hat{\gamma} \in N_S(\gamma)} \pi(\hat{\gamma})}, \frac{w_S(|\gamma'|)}{\sum_{\hat{\gamma} \in N_S(\gamma')} \pi(\hat{\gamma})} \right\},
\]
\( w_S(|\gamma|) = w_S(|\gamma'|) \).

In each case, \( \pi(\gamma) T(\gamma, \gamma') \) is symmetric in \( \gamma \) and \( \gamma' \) and the result follows.

**Proof of Lemma 3.5 (pMTM)**

*Proof.* The proof makes use of weight function \( \omega(\gamma_j, v_j, m) \) defined in (3.11), and acceptance probability \( \alpha_m(\gamma, \gamma') \) given by (3.12). Given an inclusion vector \( \gamma \), choose between Add, Remove and Swap with probabilities \( w_A(|\gamma|), w_R(|\gamma|) \) and \( w_S(|\gamma|) \), respectively. Let \( m \in \{ A, R, S \} \) denote this choice, and \( m' \in \{ R, A, S \} \) is the paired reverse neighborhood. In each case, the transition kernel is described in Section 3.4.5.

If \( m \in \{ A, R \} \): See steps (2a) and (4a/b) in Section 3.4.5 for construction of transition kernel \( T(\gamma, \gamma') \). Let \( \omega_j = \omega(\gamma_j, v_j, m) \) denote the forward Bernoulli inclusion
probability for predictor $j$, and $\bar{\omega}_j = \omega(\gamma'_j, v_j, m')$ denotes its inclusion probability in the reverse paired neighborhood. Then for $\gamma' = \text{tog}(k, \gamma)$,

$$
\pi(\gamma) T(\gamma, \gamma') = \pi(\gamma) w_m(\gamma) \sum_{\eta, \eta' \in \{0, 1\}^p \atop \eta_k = \eta'_k = 1} \left[ \omega_k \left\{ \prod_{j \neq k} \omega_j^{n_j} (1 - \omega_j)^{1-n_j} \bar{\omega}_j^{n_j} (1 - \bar{\omega}_j)^{1-n_j} \right\} \right.
\times \frac{\pi(\gamma')}{{\pi(\text{tog}(j, \gamma))}} \min \left\{ 1, \frac{w_m(\gamma')}{w_m(\gamma)} \frac{\omega_k \sum_{j, n_j = 1} \pi(\text{tog}(j, \gamma))}{\omega_k \sum_{j, n'_j = 1} \pi(\text{tog}(j, \gamma'))} \right\] 

$$

$$
= \pi(\gamma) \pi(\gamma') \sum_{\eta, \eta' \in \{0, 1\}^p \atop \eta_k = \eta'_k = 1} \left[ \left\{ \prod_{j \neq k} \omega_j^{n_j} (1 - \omega_j)^{1-n_j} \bar{\omega}_j^{n_j} (1 - \bar{\omega}_j)^{1-n_j} \right\} \right.
\times \min \left\{ \frac{w_m(\gamma)}{\sum_{j, n_j = 1} \pi(\text{tog}(j, \gamma))}, \frac{w_m(\gamma')}{{\sum_{j, n'_j = 1} \pi(\text{tog}(j, \gamma'))} } \right\].

If $m = S$: See steps (2b) and (4c) in Section 3.4.5 for construction of transition kernel $T(\gamma, \gamma')$. Denote forward bernoulli inclusion probabilities for a pair of predictors $(r, a) \in \gamma \times \gamma^c$ as $\omega_r = \omega(\gamma_r, v_r, m = R)$, $\omega_a = \omega(\gamma_a, v_a, m = A)$. Likewise for reverse inclusion probabilities, let $\omega'_r = \omega(\gamma'_r, v_r, m = R)$, $\omega'_a = \omega(\gamma'_a, v_a, m = A)$ and $(r, a) \in \gamma' \times \gamma^c$. For $\gamma' = \text{tog}((r^*, a^*), \gamma)$, the following facts are noted: (i) $\omega_r = \omega'_r$, $r \in \gamma \setminus \{r^*\}$; (ii) $\omega_a = \omega'_a$, $a \in \gamma \setminus \{a^*\}$; (iii) $\omega_{r=a^*} = \omega'_{r=a^*} = 1$; (iv) $\omega_{a=r^*} = f(v_{a^*})$ and $\omega'_{a=r^*} = f(v_{a^*})$, see (3.13); and (v) $w_S(\gamma) = w_S(\gamma')$. Then,
\[ \pi(\gamma) T(\gamma, \gamma') \]
\[ = \pi(\gamma) w_S(\lfloor \gamma \rfloor) \]
\[ \times \sum_{(r,a) \in \gamma \times [n]} \omega_{r*} \omega_{a*} \left\{ \prod_{(r',a') \neq (r*,a*)} \omega_{r'}^{\eta_r + \eta'_r (1 - \omega_r)^2 - \eta_r} \omega_{a'}^{\eta_a + \eta'_a (1 - \omega_a)^2 - \eta_a} \right\} \]
\[ \times \frac{\pi(\gamma')}{\sum_{(r,a): \eta_r = \eta_a = 1} \pi(\log((r, a), \gamma))} \]
\[ \times \min \left\{ 1, \frac{w_S(\lfloor \gamma' \rfloor)}{w_S(\lfloor \gamma \rfloor)} \frac{\omega_{r*}^{\eta_r} \omega_{a*}^{\eta_a}}{\sum_{(r,a): \eta_r = \eta_a = 1} \pi(\log((r, a), \gamma))} \right\} \]
\[ = \pi(\gamma) \pi(\gamma') \sum_{(r,a) \neq (r*,a*)} \left\{ \prod_{(r',a') \neq (r*,a*)} \omega_{r'}^{\eta_r + \eta'_r (1 - \omega_r)^2 - \eta_r} \omega_{a'}^{\eta_a + \eta'_a (1 - \omega_a)^2 - \eta_a} \right\} \]
\[ \times \min \left\{ \frac{w_S(\lfloor \gamma \rfloor)}{\sum_{(r,a): \eta_r = \eta_a = 1} \pi(\log((r, a), \gamma))}, \frac{w_S(\lfloor \gamma' \rfloor)}{\sum_{(r,a): \eta'_r = \eta'_a = 1} \pi(\log((r, a), \gamma'))} \right\}. \]

In each case, i.e. for paired-moves \( m \in \{ A, R, S \} \), \( \pi(\gamma) T(\gamma, \gamma') \) is symmetric in \( \gamma \) and \( \gamma' \) and the result follows. \( \square \)
Appendix C

Additive GP (aGP)

C.1 Theory guarantee for structured nonparametric regression

Raskutti et al. (2012) consider sparse additive models with univariate components, where \( f(x) = \sum_{h=1}^{k} f_h(x_{\gamma_h}) \), the sum of \( k \) univariate functions each having unknown smoothness \( \alpha > 0 \) and \( \gamma_h \in \{1, \ldots, p\}, \gamma_h \cap \gamma_p = \emptyset \). This corresponds to the generalized additive model setting (Hastie et al., 1986) with a sparsity assumption on the number of important predictors, \( k \ll p \). Under mild regularity conditions, they prove that

\[
\mathbb{E}^2_n \geq k n^{-2\alpha/(2\alpha + 1)} + k \frac{\log p}{n}, \quad p, n \to \infty. \tag{C.1}
\]

The first term in (C.1) is the sum of minimax risks of estimating each component and the second term is the penalty for variable selection uncertainty. Here, the additional assumption of univariate additive structure in the unknown regression function improves vastly over minimax rate \( n^{-2\alpha/(2\alpha + d)} + d \log(p/d)/n \) when \( f \) is allowed to be fully nonparametric, which is small only when \( d = o(\log n) = o(\log \log p) \), i.e., \( f \) is extremely sparse in the observed predictors.
Full additivity of the GLM is overly restrictive and does now allow inference on meaningful higher-order interaction effects between predictors. The proposed additive-interactive function class (4.6) relaxes this requirement, providing a new avenue for nonparametric regression in the \( p \gg n \) setting. Yang and Tokdar (2014) provide a minimax error rate for estimating \( f \) under an \( L_2 \) loss which depends on \( n, p, \) and component sizes \( d_1, \ldots, d_k, \) and show existence of statistical methods that offer adaptive and near optimal performance under this assumption. Their result subsumes the earlier result of Raskutti et al. (2010) as a special case when each component function is univariate with common smoothness \( \alpha. \)

**Theorem 2 (Yang and Tokdar, 2014)** Under fairly mild assumption on a measure \( Q \) over the predictor design, the minimax risk for the additive interactive framework is

\[
r_n^2(M(\alpha, d, L, k, p, \tilde{d})) = c(d_{\text{max}}, \tilde{d}) \sum_{h=1}^{k} \left[ \left( \frac{n}{\sigma^2} \right)^{-\frac{2\alpha_h}{2\alpha_h + |\gamma_h|}} + \sigma^2 \frac{|\gamma_h| \log(p/|\gamma_h|)}{n} \right]
\]

(C.2)

where \( c(d_{\text{max}}, \tilde{d}) \) is a number between \( 1/B \) and \( \sqrt{B} \) with \( B = 1 + d_{\text{max}}(\tilde{d} - 1). \)

The overall rate in (C.2) is dictated by the component function having the largest number of predictors. This improves massively over the minimax rate of estimation in the traditional sparse nonparametric regression setting in which \( f \) is assumed to depend on \( d \) of the original \( p \) predictors; in particular, when \( d \leftarrow \sum_{h=1}^{k} d_h \) as above, the latter has \( r_n^2 \approx n^{-2\alpha/(2\alpha+d)} \) which can be significantly larger than (C.2) when \( \max_h |\gamma_h| \leq \tilde{d} \ll d. \)

**C.2 An MCMC algorithm**

The paired-move neighborhood sampler (Sections 3.4.5) together with inter-component moves (Section 4.5.1) are used to efficiently explore the space of one-away inclusion vectors for aGP regression. Algorithm 3 details the sampling scheme.
Algorithm 3 aGP Markov chain sampling

Input: $k_{\text{min}} = \lfloor \log(p) \rfloor$, $k_{\text{max}} = \lfloor p^{1/2} \rfloor$, $b_0 = \max\{100, \lfloor T/10 \rfloor\}$, $\Delta = 0.10$.
Output: Approximate posterior draws for aGP parameters $\{\rho, \lambda, \gamma\}$

1: function aGP.SAMPLE \(\triangleright\) Sampling for aGP model (4.8)
2: Call UPDATE.ACTIVESET($\Theta, \mathcal{A}, B$) \(\triangleright\) see Algorithm 2
3: Sample $\tau \sim \text{Beta}(\mu', (d^* + \sum_{h \in \mathcal{A}} |\gamma_h|)/\nu', \nu' = (1 + k_A)p)$
4: if Unif$(0, 1) < \Delta$ then \(\triangleright\) Draw an ICM proposal
5: Select ICM move $m \in \{\text{CD}, \text{PD}, \text{PS}\}$
6: Propose $\Theta' = (\gamma', \theta') \sim p(\cdot|A_m(\gamma), \theta, y)$
7: if Unif$(0, 1) < \alpha_m(\Theta, \Theta')$ then \(\triangleright\) see Section 4.5.1 (4.14)
8: Set $\gamma, \rho, \theta \leftarrow \gamma', \rho', \theta'$
9: if $m = \text{CD}$ then
10: go to line 13
11: else
12: for $h \in \mathcal{A}$ do
13: Sample $\rho_h, \lambda_h \sim \pi(\cdot|\gamma_h, \Theta_{-h}, y)$ \(\triangleright\) see Section 4.5 (4.12)
14: end for
15: end if
16: else \(\triangleright\) Draw a pMTM proposal
17: for $h \in \mathcal{A}$ do
18: Select move $m \in \{A, R, S\}$
19: Propose $\gamma'_h \sim p(\cdot|N_m(\gamma_h), \Theta_{-h}, y)$ \(\triangleright\) see Section 3.4.5 (3.12)
20: if Unif$(0, 1) < \alpha_m(\gamma_h, \gamma'_h)$ then
21: Set $\gamma_h \leftarrow \gamma'_h$
22: end if
23: Sample $\rho_h, \lambda_h \sim \pi(\cdot|\gamma_h, \Theta_{-h}, y)$ \(\triangleright\) see Section 4.5 (4.12)
24: end if
25: if $\rho_h > 0, |\gamma_h| > 0$ then
26: Update $v_j, j \in \gamma_h$ \(\triangleright\) see Section 3.4.6 (3.14)
27: end if
28: end for
29: end if
30: end function
31: end function

C.2.1 MCMC stationary proof(s)

proof of Lemma 4.2

Proof. Given state vector $\xi = (\gamma, \theta)$, $\xi_l = (\gamma_l, \rho_l, \lambda_l)$ and $l \in \mathcal{A}$, an ICM move proceeds by choosing between “cross donate”, “paired donate”, or “paired swap” moves with probabilities $w_{\text{CD}}, w_{\text{PD}}$ and $w_{\text{PS}}$, respectively. For move $m \in \{\text{CD}, \text{PD}, \text{PS}\}$, the paired reverse neighborhood $A_m(\gamma)$ is constructed (see Section 4.5.1). Select $\xi'$ with probability $\pi(\xi'|\cdot)/\sum_{\xi \in A_m(\gamma)} \pi(\xi|\cdot)$ and accept it with probability $\alpha_m(\xi, \xi')$ given by (4.14). Below, $k_{\text{ne}} \leq k_A$ denotes the number of non-empty active components.
If $m = \text{CD}$: For $\gamma' = \{\ldots, \gamma_u + 1, \gamma_v - 1, \ldots\}$, where component $\gamma_v$ donates predictor $k$ to component $\gamma_u$, and
\[
\pi(\gamma|\theta, -) T(\gamma, \gamma')
\]
\[
= \pi(\gamma|\theta, -) \frac{w_{\text{CD}}}{k_{\text{ne}}} \frac{\pi(\gamma'|\theta, -)}{\sum_{\gamma \in N_{\text{CD}}(\gamma; \psi)} \pi(\gamma|\theta, -)} \min \left\{ 1, \frac{\sum_{\gamma \in N_{\text{CD}}(\gamma; \psi)} \pi(\gamma|\theta, -)}{\sum_{\gamma \in N_{\text{CD}}(\gamma'; \psi)} \pi(\gamma|\theta, -)} \right\}
\]
\[
= \pi(\gamma|\theta, -) \pi(\gamma'|\theta, -) \min \left\{ \frac{w_{\text{CD}}/k_{\text{ne}}}{\sum_{\gamma \in N_{\text{CD}}(\gamma; \psi)} \pi(\gamma|\theta, -)}, w_{\text{CD}}/k_{\text{ne}} \right\} \cdot \sum_{\gamma \in N_{\text{CD}}(\gamma'; \psi)} \pi(\gamma|\theta, -)
\]

If $m \in \{\text{PD, PS}\}$: Choose component indices $(u, v) \in A$ with $d_u, d_v > 0$. If $m = \text{PD}$, then $\gamma' = \{\ldots, \gamma_u + 1, \gamma_v - 1, \ldots\}$, where component $\gamma_v$ donates predictor $k$ to component $\gamma_u$; otherwise $m = \text{PS}$, and for $j \in \gamma_u(\neq \gamma_v)$ and $k \in \gamma_v(\neq \gamma_u)$, $\gamma' = \{\ldots, \gamma_u - 1, 1, \gamma_v - 1, k + 1, \ldots\}$. In addition, here $A_m(\gamma; (u, v)) = N_m(\gamma; (u, v)) \times G^2$ and
\[
\pi(\xi) T(\xi, \xi')
\]
\[
= \pi(\xi) \frac{w_m}{k_{\text{ne}}/2} \pi(\xi') \min \left\{ 1, \frac{\sum_{\xi \in A_m(\gamma; (u, v))} \pi(\xi)}{\sum_{\xi \in A_m(\gamma'; (u, v))} \pi(\xi)} \right\}
\]
\[
= \pi(\xi) \pi(\xi') \frac{w_m}{k_{\text{ne}}/2} \min \left\{ \frac{w_m/k_{\text{ne}}}{\sum_{\xi \in A_m(\gamma; (u, v))} \pi(\xi)}, \frac{w_m/k_{\text{ne}}}{\sum_{\xi \in A_m(\gamma'; (u, v))} \pi(\xi)} \right\}.
\]

In all cases, i.e. $m \in \{\text{CD, PD, PS}\}$, the expression $\pi(\xi) T(\xi, \xi')$ is symmetric in its arguments. Hence the mixture ICM proposal distribution comprising “cross donate”, “paired donate”, and “paired swap” moves preserves stationarity of pMTM.

C.3 Additional figures
**Figure C.1:** Marginal predictor importance (inclusion probability) for BART, RF, and Lasso on each simulated test function considered in Section 4.6.1. Competitors use the same data set \( (n = 100, p = 1000) \) that generated Figure 4.6.1.
Importance scores $\{v_j, j = 1, \ldots, p\}$

GP parameters $\{\rho_l, \lambda_l, l \in A\}$

Trace plot ($\sigma^2$)

Boston

Crime

Riboflavin

Cookie

Figure C.2: Posterior plots and comparisons for the real datasets considered in Section 4.6.3. Left: plot of predictor importance scores, $v_j$, $j = 1, \ldots, p$. Middle: bar-plots of GP scale parameters $\rho, \lambda$ for contributing components (i.e., having 75% quantile over posterior draws for $\rho_l > 0$). Right: trace plot of posterior draws for variance parameter, $\sigma^2$. 

155
Appendix D

Tensor regression

D.1 MCMC algorithm

The following derivations concern the M-DGDP prior (5.10) and the sampling algorithm outlined in Section 5.5.2.

For step (1b) Recall from Section 5.4.2 that $\tau \sim \text{Ga}(a_r, b_r)$ and $\Phi \sim \text{Dirichlet}(\alpha_1, \ldots, \alpha_R)$ and denote $p_0 = \sum_{j=1}^{D} p_j$. Then,

$$
\pi(\Phi|B, \omega) \propto \pi(\Phi) \int_{0}^{\infty} \pi(B|\omega, \Phi, \tau) \pi(\tau) d\tau
$$

$$
\propto \left[ \prod_{r=1}^{R} \phi_r^{a_r-1} \right] \int_{0}^{\infty} \prod_{r=1}^{R} \left( \tau \phi_r \right)^{-p_0/2} \exp \left( -\frac{1}{\tau \phi_r} \sum_{j=1}^{d} \| \beta_{jr} \|^2 / (2\omega_{jr}) \right) \tau^{a_r-1} \exp(-b_r \tau) d\tau
$$

$$
\propto \left[ \prod_{r=1}^{R} \phi_r^{a_r-R/2-1} \right] \int_{0}^{\infty} \tau^{-R/2-1} \prod_{r=1}^{R} \exp \left( -\frac{C_r}{\tau \phi_r} - b_r (\tau \phi_r) \right) d\tau
$$

with $C_r = \sum_{j=1}^{d} \| \beta_{jr} \|^2 / (2\omega_{jr})$. When $a_r = \sum_{r=1}^{R} \alpha_r$, this is the kernel of a generalized inverse Gaussian (gIG) distribution on variable $(\tau \phi_r)$. Recall: $X \sim f_X(x) = \text{giG}(p, a, b) \propto x^{p-1} \exp(-ax + b/x)/2$. 

156
In addition, following the result from normalized random measures (see Lemma D.2),

for independent random variable $T_r \sim f_r$ on $(0, \infty)$, the joint density of $\{\phi_r = T_r / \sum_{r \neq r} T_r : r = 1, \ldots, R\}$ has support on $S^{R-1}$. In particular,

$$f(\phi_1, \ldots, \phi_{R-1}) = \int_0^\infty t^{R-1} \prod_{r=1}^R f_r(\phi_r t) \, dt, \quad \phi_R = 1 - \sum_{r<R} \phi_r.$$  

Substituting $f_r(x) \propto x^{-\delta_r} \exp(-C_r/x) \exp(-b_r x)$ in the above expression yields

$$f(\phi_1, \ldots, \phi_{R-1}) \propto \int_0^\infty \tau^{R-1} \prod_{r=1}^R (\phi_r \tau)^{-\delta_r} \exp \left( -\frac{C_r}{(\phi_r \tau)} - b_r (\phi_r \tau) \right) d\tau$$

$$= \left[ \prod_{r=1}^R \phi_r^{-\delta_r} \right] \int_0^\infty \tau^{R-\sum_r \delta_r-1} \prod_{r=1}^R \exp \left( -\frac{C_r}{(\phi_r \tau)} - b_r (\phi_r \tau) \right) d\tau.$$

Matching exponents between this expression and the preceding one implies (1) $a_r - R(p_0/2) - 1 = R - \sum_r \delta_r - 1$, and (2) $\delta_r = 1 + p_0/2 - \alpha_r$. Then,

$$a_r = R(1 + p_0/2) - \sum_r \delta_r$$

$$a_r = R(1 + p_0/2) - (R + Rp_0/2 - \sum_r \alpha_r) = \sum_r \alpha_r$$

as previously noted. Hence, draws from $[\Phi|\alpha, B, W]$ are obtained by sampling $T_r \sim f_r = g_\beta(\alpha_r - p_0/2, 2b_r, 2C_r)$ independently for $r = 1, \ldots, R$, and subsequently renomalizing as described above.

D.2 Prior moments

proof of lemma 5.1

Proof. Appealing to lemma D.1 and convexity of $h(x) = x^{-D}, D > 0$, the lower bound is straightforward, giving

$$\text{var}(B_{i_1, \ldots, i_D}) \geq \sum_{r=1}^R \left( \frac{b_\omega}{a_\omega - 1} \right)^D \prod_{l=1}^r (a_l - 1)^{-D}.$$
Using the generalized Hölder inequality, for \( q \in (0, \infty) \) and \( p_1, \ldots, p_r \in (0, \infty] \) such that \( \sum_{i=1}^r p_i^{-1} = 1/q \), one has \( (\prod_{i=1}^r |X_i|^q)^p \leq \left( \prod_{i=1}^r |X_i|^{p_i} \right)^q \). From (5.9), let \( X_i = \delta_i^{-1} \) then \( X_i \sim \text{IG}(a_i, 1) \). Setting \( q = D \) and \( p_l = rD \), we have

\[
\mathbb{E}|X_l|^{p_l} = \int x^{rD} x^{-a_l-1} \exp(-1/x) \, dx = \Gamma(a_l - rD)/\Gamma(a_l).
\]

Hence we choose \( a_l > rD \) and since \( \tau_R = \delta_1 \cdots \delta_R \), we set \( a_l = RD \) in particular for \( 1 \leq l \leq R \). Hence,

\[
\text{var}(B_{i_1, \ldots, i_D}) \leq \sum_{r=1}^R \left( \frac{b_\omega}{a_\omega - 1} \right)^D \prod_{l=1}^r \left( \frac{\Gamma(a_l - rD)}{\Gamma(a_l)} \right)^{D/p_l}.
\]

\[ \square \]

**Proof of Lemma 5.2**

*Proof.* Using priors defined in (5.10), one has \( C_\lambda = \mathbb{E}_\lambda(1/\lambda^2) = \frac{b_\omega^2}{(a_\lambda - 1)(a_\lambda - 2)} \) for any \( a_\lambda > 2 \). In addition, the following inequalities are useful to bound the latter quantity:

- If \( \alpha_1 = c/R, \ c \in \mathbb{N}_+ \), \( \Gamma(\alpha_0 + D)/\Gamma(\alpha_0) = \alpha_0(\alpha_0 + 1) \cdots (\alpha_0 + D - 1) \). Using the fact that \( \log(x + 1) \leq x, \ x \geq 0 \), one has \( \log(\alpha_0) + \cdots + \log(\alpha_0 + D - 1) \leq \alpha_0D - 1 + \sum_{k=1}^{D-2} k \). Then \( \alpha_0^D \leq \Gamma(\alpha_0 + D)/\Gamma(\alpha_0) \leq A_r \exp(\alpha_0D) \) where \( A_r = \exp(-1 + \sum_{k=1}^{D-2} k) = \exp((D^2 - 3D)/2), \ D \geq 2 \).

- Trivially, \( ||\Phi||_D^D \leq 1 \); in addition, by Hölder’s inequality, for any \( x \in \mathbb{R}^k \) and \( 0 < r < p \), one has \( ||x||_p \geq k^{-(\frac{1}{r} - \frac{1}{p})} ||x||_r \). In our setting, \( D \geq 2 \). Taking \( r = 1 \) in the latter yields \( ||\Phi||_D^D \geq R^{-(D-1)} \).

Recall \( \alpha_0 = \sum_{r=1}^R \alpha_r = \alpha_1 R \) (assuming exchangeability between the ranks a priori under the M-DGDP prior). This leads to the lower and upper bounds expressions.
for the prior voxel-level variance:
\[
\text{var}(B_{i_1,\ldots,i_D}) \geq (2C_\lambda)^D \left( \alpha_1 R \right)^D R^{-(D-1)} / b^D_r = (2C_\lambda)^D \alpha_1^D R / b^D_r,
\]
\[
\text{var}(B_{i_1,\ldots,i_D}) \leq A_r (2C_\lambda)^D \exp(\alpha_1 RD) / b^D_r.
\]

D.3 Miscellaneous lemma

**Lemma D.1.** If \( f_1, \ldots, f_d \) are convex functions such that \( f_i > 0 \) and \( f'_i < 0 \) for all \( i = 1, \ldots, d \), then \( \prod_{i=1}^d f_i \) is convex.

**proof of lemma D.1**

Proof. First we prove the result for \( d = 2 \). Note that \( (f_1 f_2)'' = f''_1 f_2 + f''_2 f_1 + 2 f'_1 f'_2 > 0 \). So the result holds for \( d = 2 \). Also \( f_1 f_2 > 0 \) and \( (f_1 f_2)' = f'_1 f_2 + f'_2 f_1 < 0 \).

Assume the result to hold for \( d - 1 \), i.e. \( \prod_{i=1}^{d-1} f_i \) is convex and \( (\prod_{i=1}^{d-1} f_i)' < 0 \). Then \( (\prod_{i=1}^d f_i)'' = f''_d \prod_{i=1}^{d-1} f_i + f_d (\prod_{i=1}^{d-1} f_i)' + 2 f'_d (\prod_{i=1}^{d-1} f_i)' > 0 \). Hence \( \prod_{i=1}^d f_i \) is convex.

**Lemma D.2.** Suppose \( T_1, \ldots, T_m \) are independent random variables with \( T_j \) having density \( f_j \) supported in \((0, \infty)\). Let \( \phi_j = \frac{T_j}{\sum_{i=1}^m T_i} \). Then the joint density of \( (\phi_1, \ldots, \phi_{m-1}) \) has a joint density supported on the simplex \( S^{m-1} \) and is given by
\[
f(\phi_1, \ldots, \phi_{m-1}) = \int_0^{\infty} t^{m-1} \prod_{i=1}^m f_j(\phi_j t) dt,
\]
where \( \phi_m = 1 - \sum_{i=1}^{m-1} \phi_i \).

**proof of lemma D.2**

Proof. This result is well known in the theory of normalized random measures (Kruijer et al., 2010; Zhou and Carin, 2013).
Bibliography


Hastie, T., Tibshirani, R., and Friedman, J. (2009), The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Springer.


Manolopoulou, I., Chan, C., and West, M. (2010), “Selection sampling from large data sets for targeted inference in mixture modeling,” *Bayesian analysis (Online)*, 5, 1.


167


Biography

Shaan Qamar was born on January 3, 1987 in Karachi, Pakistan. He received his Bachelors degree in Applied Physics from Cornell University in 2008 and went on to obtain a Masters in Operations Research from Cornell University in 2009. Shaan moved to Durham, NC in August 2011 to pursue doctoral studies at the Department of Statistical Science at Duke University. In 2014, he earned a Masters degree in Statistics en route to his Ph.D.