UNIVERSITY OF CALIFORNIA
SANTA CRUZ

# BAYESIAN NONSTATIONARY GAUSSIAN PROCESS MODELS VIA TREED PROCESS CONVOLUTIONS

A dissertation submitted in partial satisfaction of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in

STATISTICS AND STOCHASTIC MODELING

by

**Waley W. J. Liang**

June 2012

The Dissertation of Waley W. J. Liang
is approved:

Professor Herbert K. H. Lee, Chair

Professor Bruno Sansó

Professor Athanasios Kottas

Professor Andrew Fisher

Tyrus Miller
Vice Provost and Dean of Graduate Studies

# Table of Contents

**List of Figures** **vii**

**Abstract** **xii**

**Dedication** **xiii**

**Acknowledgments** **xiv**

**1 Introduction** **1**

1.1 Literature Review . . . . . 2

1.2 Motivation . . . . . 5

1.3 Bayesian Modeling and Inference . . . . . 7

1.3.1 Markov chain Monte Carlo . . . . . 8

1.3.2 Prediction . . . . . 10

1.4 Random Fields . . . . . 11

1.4.1 Stationarity . . . . . 12

1.4.2 Smoothness . . . . . 13

1.5 Stationary Gaussian Process Models . . . . . 14

1.5.1 Gaussian Processes . . . . . 14
1.5.2 Modeling and Bayesian Estimation . . . . . 16
1.6 Process Convolution Gaussian Process Models . . . . . 18
1.6.1 Process Convolutions . . . . . 18
1.6.2 Modeling and Bayesian Estimation . . . . . 21
1.7 Treed Models . . . . . 22

**2 Simulation Studies on Process Convolution GP Models 26**

2.1 Introduction . . . . . 26
2.2 Simulation setup . . . . . 27
2.3 Simulation results . . . . . 35
2.3.1 Gaussian kernel . . . . . 35
2.3.2 Bézier kernel . . . . . 37
2.3.3 Exponential kernel . . . . . 38
2.3.4 A rule of thumb for the number of bases . . . . . 39
2.4 Conclusion . . . . . 40

**3 Treed Process Convolution GP model 56**

3.1 Introduction . . . . . 56
3.2 Model setup . . . . . 57
3.3 Bayesian Estimation . . . . . 60
3.4 Treed Model Proposals . . . . . 65
3.5 Illustration . . . . . 71

3.5.1 1-d Synthetic Sinusoidal Data . . . 71
3.5.2 2-d Real Precipitation Data . . . 74
3.6 Conclusion . . . 82

**4 Variable Kernels Across Partitions 89**
4.1 Introduction . . . 89
4.2 Illustration . . . 91
4.2.1 1-d Synthetic Sinusoidal Data . . . 91
4.2.2 2-d Synthetic Exponential Data . . . 93
4.2.3 2-d Real Precipitation Data . . . 96
4.3 Conclusion . . . 102

**5 Sequential Process Convolution GP Models 119**
5.1 Introduction . . . 119
5.2 Sequential Monte Carlo and Particle Learning . . . 121
5.3 Particle Learning for Process Convolution GP . . . 124
5.4 Illustration . . . 127
5.4.1 1-d Synthetic Sinusoidal Data . . . 127
5.4.2 The Pump-and-Treat Problem . . . 132
5.5 Conclusion . . . 137

**6 Conclusion and Future Work 142**

**A Derivation 145**

A.1 Joint posterior distribution . . . . . 146

A.2 Conditional Posterior Distribution for $\mathbf{x}$ . . . . . 147

A.3 Conditional Posterior Distribution for $\lambda_\nu$ . . . . . 148

A.4 Conditional Posterior Distribution for $\phi_\nu$ . . . . . 148

A.5 Conditional Posterior Distribution for $\boldsymbol{\beta}_\nu$ . . . . . 151

A.6 Conditional Posterior Distribution for $\mathbf{Q}_\nu$ . . . . . 152

A.7 Conditional Posterior Distribution for $\boldsymbol{\beta}_0$ . . . . . 153

A.8 Conditional Posterior Distribution for $\mathbf{C}$ . . . . . 154

A.9 Conditional Posterior Distribution for $b_x$ and $b_y$ . . . . . 154

A.10 Conditional posterior distribution for $(\mathcal{T}, \boldsymbol{\rho})$ . . . . . 155

**Bibliography** **157**

# List of Figures

1.1 An example of a 1-d process convolution GP . . . . . 20

1.2 An exmaple of binary treed partitioning over a 2-d domain . . . . . 24

2.1 Correlation functions . . . . . 32

2.2 1-d Gaussian data generated based on various correlation functions . . . 33

2.3 Interpolated 2-d Gaussian realizations based on various correlation functions . . . . . 34

2.4 Process convolution GP model performance based on a Gaussian kernel for data generated via a Gaussian correlation function . . . . . 42

2.5 Process convolution GP model performance based on a Gaussian kernel for data generated via a Matérn correlation function . . . . . 43

2.6 Process convolution GP model performance based on a Gaussian kernel for data generated via a Spherical correlation function . . . . . 44

2.7 Process convolution GP model performance based on a Gaussian kernel for data generated via an Exponential correlation function . . . . . 45

2.8 Process convolution GP model performance based on a Bézier kernel for data generated via a correlation function induced by the Bézier kernel . 46
2.9 Process convolution GP model performance based on a Bézier kernel for data generated via a Matérn correlation function . . . . . 47
2.10 Process convolution GP model performance based on a Bézier kernel for data generated via a Spherical correlation function . . . . . 48
2.11 Process convolution GP model performance based on a Bézier kernel for data generated via an Exponential correlation function . . . . . 49
2.12 Process convolution GP model performance based on an Exponential kernel for data generated via a correlation function induced by the Exponential kernel . . . . . 50
2.13 Process convolution GP model performance based on an Exponential kernel for data generated via a Matérn correlation function . . . . . 51
2.14 Process convolution GP model performance based on an Exponential kernel for data generated via an Spherical correlation function . . . . . 52
2.15 Process convolution GP model performance based on an Exponential kernel for data generated via an Exponential correlation function . . . . 53
2.16 Basis spacing v.s. practical range of correlation function for 1-d data . . 54
2.17 Basis spacing v.s. practical range of correlation function for 2-d data . . 55

3.1 An example of the *Grow* and *Prune* operations . . . . . 69
3.2 An example of the *Change* operation . . . . . 69

3.3 An example of the left *Swap* operation . . . . . 69
3.4 An example of the right *Swap* operation . . . . . 70
3.5 An example of the left & right *Swap* operation . . . . . 70
3.6 An example of tree *rotations* . . . . . 70
3.7 1-d sinusoidal data . . . . . 71
3.8 Posterior predictive summary from modeling of 1-d sinusoidal data . . . 75
3.9 DIC v.s. number of basis from modeling of 1-d sinusoidal data . . . . . 76
3.10 Sensitivity analysis of prior parameters of TPCGP with a fixed kernel on 1-d sinusoidal data . . . . . 77
3.11 Precipitation and elevation data over the contiguous U.S . . . . . 78
3.12 Posterior predictive summary of TPCGP with a fixed kernel on precipitation data . . . . . 83
3.13 Posterior summary of background points from TPCGP with a fixed kernel on precipitation data . . . . . 84
3.14 Posterior summary of the linear component and observation error standard deviation from TPCGP with a fixed kernel on precipitation data . 85
3.15 Log conditional of treed models, number of partitions visited, and the accepted tree operations from TPCGP with a fixed kernel . . . . . 86
3.16 Fitted residuals from TPCGP with a fixed kernel . . . . . 87
3.17 Prediction residuals from TPCGP with a fixed kernel . . . . . 88

4.1 1-d sinusoidal data . . . . . 92

4.2 Posterior predictive summary of TPCGP with variable kernels on 1-d sinusoidal data . . . . . 94
4.3 2-d exponential response and model results from TPCGP with variable kernels . . . . . 97
4.4 Log conditional of treed models, number of partitions visited, and fitted residuals from TPCGP with variable kernels on 2-d exponential data . . 98
4.5 Posterior predictive summary of TPCGP with variable kernels on precipitation data . . . . . 104
4.6 Posterior summary of background points from TPCGP with variable kernels on precipitation data . . . . . 105
4.7 Posterior summary of the linear component and observation error standard deviation from TPCGP with variable kernels on precipitation data 106
4.8 Log conditional of treed models, number of partitions visited, and the accepted tree operations from TPCGP with variable kernels . . . . . 107
4.9 Fitted residuals from TPCGP with variable kernels . . . . . 108
4.10 Prediction residuals from TPCGP with variable kernels . . . . . 109
4.11 Fitted surface from Covariance Tapering . . . . . 110
4.12 Fitted residuals from Covariance Tapering . . . . . 111
4.13 Prediction residuals from Covariance Tapering . . . . . 112
4.14 Fitted surface from Multivariate Adaptive Regression Splines . . . . . 113
4.15 Fitted residuals from Multivariate Adaptive Regression Splines . . . . . 114
4.16 Prediction residuals from Multivariate Adaptive Regression Splines . . . 115

4.17 Fitted surface from the Predictive Process model . . . . . 116

4.18 Fitted residuals from the Predictive Process model . . . . . 117

4.19 Prediction residuals from the Predictive Process model . . . . . 118

5.1 1-d sinusoidal response and data . . . . . 127

5.2 Posterior predictive summary from SPCGP for $t = \{5, 10, 20, 30, 40, 50\}$ 129

5.3 Posterior predictive summary from SPCGP for $t = 100$ with $500, 100$ and 20 particles . . . . . 130

5.4 Model comparisons between different GP models based on PL and MCMC131

5.5 Lockwood Solvent Groundwater Plume Site . . . . . 133

5.6 Posterior predictive mean from SPCGP for $t = \{6, 9, 15, 30, 40, 50\}$ for the Pump-and-Treat problem . . . . . 139

5.7 Posterior predictive summary from SPCGP, PCGP, and PLGP for the Pump-and-Treat problem . . . . . 140

5.8 Posterior predictive mean from SPCGP based on 500, 100 and 20 particles for $g = 1$ and $g = 2$ for the Pump-and-Treat problem . . . . . 141

## Abstract

Bayesian Nonstationary Gaussian Process Models via Treed Process Convolutions

by

Waley W. J. Liang

Spatial modeling with stationary Gaussian processes (GPs) has been widely used, but the assumption that the correlation structure is independent of spatial location is invalid in many applications. Various nonstationary GP models have been developed to solve this problem, however, many of them become impractical when the sample size is large. To tackle this problem, a more computationally efficient GP model is developed by convolving a smoothing kernel with a latent process. Nonstationarity in the GP is obtained by partitioning the spatial domain and allowing a separate latent process and kernel for each partition. Partitioning is achieved using a binary tree generating process. A Bayesian approach is used to simultaneously guide partitioning and estimate the parameters of the treed model. Results based on a large real dataset show that this model is fairly computational efficient and has better prediction performance than other competitive models in the literature. In addition to the treed model, a sequential design for the standard process convolution GP model is also developed based on a method called *Particle Learning*, which makes on-line inference more efficient than running a batch inference procedure.

To my parents and friends

## Acknowledgments

I want to thank all my committee members for their insightful comments to earlier versions of this thesis. I would also like to thank the Gates Millennium Scholars Program for financially supporting my entire graduate study at UC Santa Cruz.

# Chapter 1

# Introduction

Gaussian spatial models are widely adopted in many applications such as geology, climatology, hydrology, and computer simulation experiments. In these models, a Gaussian process (GP) is specified to describe the unobserved phenomona of interest. They are convenient methods for modeling point-referenced datasets (e.g., observed precipitation counts referenced by locations over a bounded domain) and can capture much of the spatial behavior based on the specification of the correlation structure. In many cases, the correlation structure is assumed to be homogeneous across locations, meaning that the correlation between any two points in the GP does not depend on their locations. Furthermore, the underlying process is assumed to have a mean value that does not depend on locations. The mathematical term for these behaviors is called *stationarity*. The stationary assumption may be valid in some applications, but proves to be untenable in many cases. In fact, most spatial phenomona exhibit a correlation structure that depends on locations to some degree. Another issue in spatial statistics

is that recent advances in geographical information systems and the global positioning systems enable the formation of large spatial datasets. Developing efficient nonstationary spatial models for large datasets is becoming one of the core interests of spatial statisticians.

## 1.1 Literature Review

There are several categories of spatial models which have adopted different approaches for modeling spatial datasets. First, there is the standard Bayesian GP approach where a GP under a certain correlation function is specified as the prior for the underlying process of interest. Nonstationarity is either due to the pre-specified correlation function (Paciorek and Schervish, 2006) or can be induced by partitioning of the spatial domain such that each partition is modeled with a separate GP (Kim et al., 2005; Gramacy and Lee, 2008). Inference of the model parameters is carried out using Markov chain Monte Carlo (MCMC). The draw back of this approach is that each MCMC iteration requires a matrix decomposition whose complexity increases at a rate of cube of the sample size, which renders this modeling approach impractical for large datasets.

The second category approximates the underlying process of interest using process convolutions, low rank splines or basis functions (Higdon, 2002; Wikle and Cressie, 1999; Lin et al., 2000; Hoef et al., 2004; Xia and Gelfand, 2006; Kammann and Wand, 2003; Paciorek, 2007). In the process convolution approach, the process of interest is

modeled by convolving a smoothing kernel with a latent process. Nonstationarity can be induced by either varying the kernel spatially while fixing the latent process (Higdon et al., 1999; Lemos and Sansó, 2009), or by varying the dependence structure of the latent process while fixing the kernel (Fuentes and Smith, 2001; Lee et al., 2005). The drawback of varying the kernels over space is that some parameters of the kernel re-parametrization have to be fixed and overfitting and mixing problems are found when these parameters are allowed to vary (Swall, 1999). The approach by Fuentes and Smith (2001) requires defining a set of local regions wherein a stationary process is assumed for each. This set of stationary processes are treated as the latent process in the convolution. Prior knowledge of the local regions and the modeling of multiple levels of processes renders this approach not very convenient in practice. The model by Lee et al. (2005) requires the user to have substantial knowledge of the spatial field being modeled in order to tune the variogram of the latent process. One recently developed model that is very similar to the process convolution approach is the predictive process model by Banerjee et al. (2008). It defines a discrete process by fixing a grid of knots over the spatial domain and assuming the response (dependent variable) at those knots to follow a Multivariate Gaussian distribution with a specific correlation function. Then, the process of interest is modeled with a GP whose covariance is a transformation of the covariance of the discrete process. Although this approach is nonstationary by construction, it requires defining a separate set of parameters for each region when the process of interest exhibits region-specific anisotropy. However, this requires knowing the specific regions in advance and such information may not be easily available in many

applications. In general, all these methodologies reduce the dimension of the problem and is suitable for large datasets because the computational complexity depends on the number of bases instead of the sample size, where the former is often significantly smaller than the latter.

The third category is nonparametric methods. This includes the well known Multivariate Adaptive Regression Splines (MARS) (Friedman, 1991), which fits the data using the sum of a set of basis functions (hinge functions) with different weights. Despite its fast computational speed, fitting MARS to nonstationary data usually results in large residual errors. Another notable model is the deformation approach of Sampson and Guttorp (1992). The observations in the original domain are viewed as a nonlinear transformation of points in virtual domain wherein stationarirty can be asssumed. Bayesian versions have been pursued by (Damian et al., 2001) where the transformation is implemented using thin plate splines, and by (Schmidt and O'Hagan, 2003) where the transformation is explained using a bivariate GP. Nonstationarity is introduced through a nonparametric specification of the covariance function. Other other hand, Gelfand et al. (2005) have proposed a spatial Dirichlet process (SDP) mixture model to produce a random spatial process that is neither Gaussian nor stationary. Essentially, it is a Dirichlet process defined on a space of surfaces and adopts the distribution of a stochastic process as its base measure. The realizations are discrete probability measures having countable support with probability one. Although in principle it can capture virtually any distribution for the observables, the way inference has to be done is not satisfactory because it insists that given the countable collection of surfaces, only

one realization is taken as the model surface. Improvement has been introduced by Duan et al. (2007), where a random distribution is placed on the spatial effects that allows different surface selection at diffeent sites. However, this improvement renders the model computationally more demanding and may not be suitable for large datasets.

## 1.2 Motivation

Among the models mentioned above, the kernel convolutions/basis functions approach is generally more efficient than the other two categories in terms of computational speed. Although the partitioning GP models can alleviate this disadvantage to a certain degree (since the inversion of the sub-covariance matrix in each partition is faster than the inversion of the joint covariance matrix), the computational drawback is inevitable as the sample size becomes huge. On the other hand, the computational complexity of process convolutions is tied to the dimension of the spatial domain. Since many environmental applications have large datasets collected over a geographical domain of dimension two, the standard GP approach is simply impractical, and process convolutions is a more suitable choice. In this dissertation, a nonstationary GP model is developed based on the process convolution approach. The spatial domain is partitioned using a binary tree generating procedure similar to that of Classification and Regression Trees (CART) (Breiman et al., 1984). Nonstationarity in the GP is induced by allowing a separate latent process and kernel for each partition. Under this setup, a Bayesian approach is used to explore the treed model space and estimate the parameters

simultaneously.

In addition to the treed model, this dissertation also presents a sequential design for the standard process convolution GP model. The motivation is that for sequential problems such as computer simulation experiements, the model has to be updated as new data points become available. Bayesian inference using Markov chain Monte Carlo (MCMC) would be inefficient for this problem because MCMC has to be repeated for every new data arrival. A better alternative is to apply a sequential inference procedure so that the model can easily be updated with new information without having to re-learn from old data. The approach taken in this dissertation is based on a Sequential Monte Carlo method called *Particle Learning* (Carvalho et al., 2010).

This dissertation is organized as follows. The next few sections provide a brief review of Bayesian modeling, random fields, standard GP models, process convolution GP models, and treed models. Chapter 2 summarizes results from a simulation study on process convolution GP models, with a particular interest in finding the sufficient number of bases. Chapter 3 provides a detailed formulation of the treed process convolution GP model based on a fixed kernel. Chapter 4 extends the basic treed model from Chapter 3 to allow variable kernels across partitions. Chapter 5 presents a sequential design for the standard process convolution GP model via Particle Learning. Discussion and conclusion is given in Chapter 6.

## 1.3 Bayesian Modeling and Inference

The models developed in this thesis are based on a Bayesian approach. In general, given a model $\mathcal{M}$, the data $\mathbf{y}$ is assumed to follow a parametric distribution $P(\mathbf{y}|\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is a parameter vector that describes the distribution. Inference of the parameters proceeds by imposing a prior distribution $P(\boldsymbol{\theta})$, on the parameter vector $\boldsymbol{\theta}$, which yields a posterior distribution $P(\boldsymbol{\theta}|\mathbf{y})$ such that

$$P(\boldsymbol{\theta}|\mathbf{y}) = \frac{P(\mathbf{y}|\boldsymbol{\theta})P(\boldsymbol{\theta})}{P(\mathbf{y})}. \tag{1.1}$$

Prior distributions provide knowledge (or ignorance) about the model parameters. Specification is based on past research results, or can be defined hierarchically in terms of other parameters, which have their own hyperprior distributions. When the resulting posterior distributions are in the same family of the priors distributions, the priors are called *conjugate*. Conjugate priors are convenient because they produce closed-form posterior distributions, which are analytically tractable. If a conjugate prior for the full set of parameters $\boldsymbol{\theta}$ is not avaiable (and hence a closed-form posterior for $\boldsymbol{\theta}$ is unknown), it may still be possible to define a *conditionally conjugate* prior distribution for a subset of the parameters conditioned on the others. For example, let $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$, it may be possible to define $P(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)$ such that $P(\boldsymbol{\theta}_1|\mathbf{y}, \boldsymbol{\theta}_2)$ can be obtained in closed-form and is in the same family as $P(\boldsymbol{\theta}_1|\boldsymbol{\theta}_2)$. The main advantage of Bayesian statistical modeling over the frequentist approach is the full account of uncertainty. The posterior distribution of a model contains a full summary of the parameters rather than just point estimates.

The Bayesian approach not only can be applied to the estimation of parameters of a fixed model, but can also be used to explore the the joint distribution of the model and its parameters $\{\boldsymbol{\theta}, \mathcal{M}\}$, when the model is assumed to be random. In this case, a prior distribution $P(\boldsymbol{\theta}, \mathcal{M})$ is specified in order to obtain the posterior distribution for $\{\boldsymbol{\theta}, \mathcal{M}\}$ in a similar fashion as (1.1):

$$P(\boldsymbol{\theta}, \mathcal{M}|\mathbf{y}) = \frac{P(\mathbf{y}|\boldsymbol{\theta}, \mathcal{M})P(\boldsymbol{\theta}, \mathcal{M})}{P(\mathbf{y})}. \tag{1.2}$$

In many problems, it is more convenient to assume a conditional relationship between the model and its parameters: $P(\boldsymbol{\theta}, \mathcal{M}) = P(\boldsymbol{\theta}|\mathcal{M})P(\mathcal{M})$, so that the $P(\boldsymbol{\theta}, \mathcal{M})$ can be identified by specifying $P(\boldsymbol{\theta}|\mathcal{M})$ and $P(\mathcal{M})$ separately. Model selection can be done by picking the model with the maximum posterior probability, so called the *maximum a' posteriori* (MAP) model, or via the use of *Bayes factors* (Kass and Raftery, 1995). Alternatively, model averaging can be performed by integrating over the model space (Hoeting et al., 1999).

### 1.3.1 Markov chain Monte Carlo

When the posterior distributions of parameters cannot be obtained in closed form, inference is usually carried out by a simulation procedure called Markov chain Monte Carlo (MCMC). The is the main tool of inference used in this dissertation. The key of MCMC is to establish a markov chain whose stationary distribution is the posterior distribution of interest. The states of the chain after convergence are treated as samples from the posterior distribution. The transition from state $\boldsymbol{\theta}_t$ to $\boldsymbol{\theta}_{t+1}$ can

be setup using either the Metropolis-Hastings (MH) algorithm (Metropolis et al., 1953; Hastings, 1970) or Gibbs sampling (Geman and Geman, 1984).

The MH algorithm proposes a new sample (or state) $\boldsymbol{\theta}^*$ from a proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_t)$ based on the sample at time $t$. The next sample $\boldsymbol{\theta}_{t+1}$ is set equal to $\boldsymbol{\theta}^*$ with probability:

$$\alpha = min\left\{1, \frac{P(\mathbf{y}|\boldsymbol{\theta}^*)P(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}_t|\boldsymbol{\theta}^*)}{P(\mathbf{y}|\boldsymbol{\theta}_t)P(\boldsymbol{\theta}_t)q(\boldsymbol{\theta}^*|\boldsymbol{\theta}_t)}\right\}, \tag{1.3}$$

or remains as $\boldsymbol{\theta}_t$ with probability $1 - \alpha$. Equation (1.3) is also referred as the MH acceptance ratio since it is a ratio of posteriors that governs the acceptance rate. Since the ratio of posterior is of interest, calculation of the scaling factor $P(\mathbf{y})$ is not necessary. Hence, the MH algorithm is useful when the posterior distributions are not available in closed form. The Gibbs algorithm is a special case of MH where the acceptance probability $\alpha$ is always 1. Parameters with conjugate priors can usually be sampled using the Gibbs algorithm. In problems where only a subset of the parameters have conjugate priors (whose posteriors can be obtained in closed form), combination of the MH and Gibbs algorithm can be utilized to obtain samples from the joint posterior distribution.

For model selection problems, the dimension of the parameter space can vary when moving from $\{\boldsymbol{\theta}_t, \mathcal{M}_t\}$ to $\{\boldsymbol{\theta}^*, \mathcal{M}^*\}$. To accommodate this change, inference is carried out using the Reversible jump Markov chain Monte Carlo (RJ-MCMC) (Green, 1995). It is an adjusted version of the regular MCMC such that a Jacobian term is added to Equation (1.3) to order to account for a stretching or shrinking of the parameter space.

It can be shown that if the proposal distribution of the augmenting parameter is equal to its prior distribution, the Jacobian is reduced to 1, and the MH acceptance ratio is

$$\alpha = min\left\{1, \frac{P(\mathbf{y}|\boldsymbol{\theta}^*, \mathcal{M}^*)P(\boldsymbol{\theta}^*|\mathcal{M}^*)P(\mathcal{M}^*)q(\boldsymbol{\theta}_t, \mathcal{M}_t|\boldsymbol{\theta}^*, \mathcal{M}^*)}{P(\mathbf{y}|\boldsymbol{\theta}_t, \mathcal{M}_t)P(\boldsymbol{\theta}_t|\mathcal{M}_t)P(\mathcal{M}_t)q(\boldsymbol{\theta}^*, \mathcal{M}^*|\boldsymbol{\theta}_t, \mathcal{M}_t)}\right\}. \tag{1.4}$$

The proposed sample $\{\boldsymbol{\theta}^*, \mathcal{M}^*\}$ is accepted as $\{\boldsymbol{\theta}_{t+1}, \mathcal{M}_{t+1}\}$ with prabability $\alpha$.

### 1.3.2 Prediction

Once a set of samples from the posterior distribution $P(\boldsymbol{\theta}|\mathbf{y})$ is obtained, the posterior predictive distribution can be computed as

$$\begin{aligned} P(\tilde{\mathbf{y}}|\mathbf{y}) &= \int P(\tilde{\mathbf{y}}|\boldsymbol{\theta}, \mathbf{y})P(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta} \\ &= \int P(\tilde{\mathbf{y}}|\boldsymbol{\theta})P(\boldsymbol{\theta}|\mathbf{y})d\boldsymbol{\theta}. \end{aligned} \tag{1.5}$$

This is a distribution of unobserved samples $\tilde{\mathbf{y}}$ (prediction) conditional on the observed data $\mathbf{y}$. In practice, computing the above integral is usually not necessary. Samples from $P(\tilde{\mathbf{y}}|\mathbf{y})$ can be obtained by drawing from $P(\tilde{\mathbf{y}}|\boldsymbol{\theta})$ based on posterior samples of $\boldsymbol{\theta}$. In the case of model selection problems, the posterior predictive distribution can be computed in a similar fashion:

$$\begin{aligned} P(\tilde{\mathbf{y}}|\mathbf{y}) &= \int P(\tilde{\mathbf{y}}|\boldsymbol{\theta}, \mathcal{M}, \mathbf{y})dP(\boldsymbol{\theta}, \mathcal{M}|\mathbf{y}) \\ &= \int P(\tilde{\mathbf{y}}|\boldsymbol{\theta}, \mathcal{M})dP(\boldsymbol{\theta}, \mathcal{M}|\mathbf{y}). \end{aligned} \tag{1.6}$$

The sampling procedure is analogous. The posterior predictive distribution can be used as a model checking tool. In general, a good model is obtained when the observed data is well fitted with the posterior predictive distribution.

## 1.4 Random Fields

Random fields are the foundation of spatial models. A random field is the generalization of a stochatic process such that the indexing variable can either be a scalar or a multidimensional vector. In mathematical terms, a random field $z$ is a collection of random variables, $z(\mathbf{s}, \omega)$, on some probability space $(\Omega, F, P)$ indexed by a variable $\mathbf{s} \in \mathcal{S}$. Values in of the random field can be real, integer, or even complex numbers. In this dissertation, only real-valued random fields are considered, i.e., $z(\mathbf{s}, \omega) \in \mathbb{R}$, and the domain $\mathcal{S}$ represents either the covariate or geographical space, namely, $\mathcal{S} \subseteq \mathbb{R}^d$. A random field is characterized by its finite-dimensional cumulative distribution $F$, such that for any finite set of locations $\{s_1, \cdots, s_n\} \in \mathcal{S}$, and any positive integer $n$,

$$F_{\mathbf{s}_1,\cdots,\mathbf{s}_n}(x_1, \cdots, x_n) = P(z(\mathbf{s}_1, \omega) \leq x_1, \cdots, z(\mathbf{s}_n, \omega) \leq x_n), \tag{1.7}$$

where $P$ denotes probability. The *mean function* of a random field is defined as

$$\mu(\mathbf{s}) = E[z(\mathbf{s}, \omega)] = \int_{\Omega} z(\mathbf{s}, \omega) dP(\omega), \tag{1.8}$$

and the *covariance function* $C(\mathbf{s}_i, \mathbf{s}_{i'})$ is given by

$$\begin{aligned} C(\mathbf{s}_i, \mathbf{s}_{i'}) &= Cov(z(\mathbf{s}_i, \omega), z(\mathbf{s}_{i'}, \omega)) \\ &= E[(z(\mathbf{s}_i, \omega) - \mu(\mathbf{s}_i))(z(\mathbf{s}_{i'}, \omega) - \mu(\mathbf{s}_{i'}))] \\ &= \int_{\Omega} (z(\mathbf{s}_i, \omega) - \mu(\mathbf{s}_i))(z(\mathbf{s}_{i'}, \omega) - \mu(\mathbf{s}_{i'})) dP(\omega). \end{aligned} \tag{1.9}$$

The covariance function specifies the second-order dependence of the random field, i.e., the strength of dependence among values in the random field as a function of indexing

locations. In order for $C(\cdot,\cdot)$ to be a valid covariance function, it must be positive definite, namely,

$$\sum_{i}\sum_{i'} a_i a_{i'} C(\mathbf{s}_i, \mathbf{s}_{i'}) > 0, \tag{1.10}$$

for any nonzero $a_i$ and $a_{i'}$. For the rest of this dissertation, the notation $z(\cdot,\omega)$ is replaced with $z(\cdot)$ for convenience.

An alternative characterization of the second-order dependence of a random field is the *variogram*, $2\gamma(\cdot)$, which is the variance of the difference between any two points in the random field,

$$\begin{aligned} 2\gamma(z(\mathbf{s}_i), z(\mathbf{s}_{i'})) &= Var[z(\mathbf{s}_i) - z(\mathbf{s}_{i'})] \\ &= E[((z(\mathbf{s}_i) - z(\mathbf{s}_{i'})) - (\mu(\mathbf{s}_i) - \mu(\mathbf{s}_{i'})))^2]. \end{aligned} \tag{1.11}$$

### 1.4.1 Stationarity

*Stationarity* is a concept describing the dependence structure of a random field across the domain. A random field $z$ is said to be *strictly stationary* if for any vector $\mathbf{h}$ the finite dimensional distributions of $\{z(\mathbf{s}_1), \cdots, z(\mathbf{s}_n)\}$ and $\{z(\mathbf{s}_1 + \mathbf{h}), \cdots, z(\mathbf{s}_n + \mathbf{h})\}$ are identical for an arbitrary $n$. That is, the random field is invariant under translation. This condition imposes a very strong requirement that few random fields can meet, making it of little use in real-life applications. In fact, in most environmental and geostatistical applications, a weaker version of statonarity is sufficient to provide a foundation for modeling and analysis. This condition is termed *second-order stationarity*, which requires that the expectation exists and is not a function of the location, and

the covariance exists and depends only on the vector $\mathbf{h}$ separating the two locations. That is,

$$\mu(\mathbf{s}) = E[z(\mathbf{s})] = \mu, \tag{1.12}$$

$$C(\mathbf{s}+\mathbf{h}, \mathbf{s}) = C(\mathbf{s}+\mathbf{h}-\mathbf{s}) = C(\mathbf{h}). \tag{1.13}$$

For $\mathbf{h} = 0$, the covariance becomes the variance,

$$Var[z(\mathbf{s})] = C(\mathbf{s}, \mathbf{s}) = C(0), \tag{1.14}$$

which implies that the variance does not depend on location. Based on second-order stationarity, the variogram can be wrriten as a function of the covariance,

$$Var[z(\mathbf{s}_i) - z(\mathbf{s}_{i'})] = 2[C(0) - C(h)]. \tag{1.15}$$

When the covariance function depends only the magnitude of the distance vector, $||\mathbf{h}||$, the random field is said to be *isotropic*. In most cases, Euclidean distance is used as the distance metric.

### 1.4.2 Smoothness

Another important property of a random field is the smoothness of the field surface. Mathematically, the smoothness of a random field is governed by its continuity and differentialbility. There are two common types of continuity and differentialbility:

1. A random field $z$ is *mean square* continuous if $E[(z(\mathbf{s}+\mathbf{h}) - z(\mathbf{s}))^2] \to 0$ as $\mathbf{h} \to 0$. The random field $z$ is *mean-square differentiable*, with mean-square derivative

$z'(\mathbf{s})$, if

$$E\left[\left\{\frac{z(\mathbf{s}+\mathbf{h})-z(\mathbf{s})}{\mathbf{h}}-z'(\mathbf{s})\right\}^2\right]\to 0, \quad \text{as } \mathbf{h}\to 0. \tag{1.16}$$

2. A random field $z$ is *path-continous*, or more generally $k$ *times path-differentiable* if its realizations are continuous or $k$ times differentiable functions, respectively.

In general, the mean-square property is a more convenient measure of the smoothness of a random field, and will be adopted in the rest of this dissertation whenever continuity and differentiability is mentioned. For more details regarding the continuity and differentiability of random fields, see Section 2.5 of Paciorek (2003).

## 1.5 Stationary Gaussian Process Models

### 1.5.1 Gaussian Processes

A common random field used in spatial modeling is the Gaussian random field. This type of random field is usually referred to as Gaussian process (GP) in the literature. A Gaussian process is a collection of random variables $\{z(\mathbf{s}) : \mathbf{s}\in\mathbb{R}^d\}$ such that for any finite set of locations $\{\mathbf{s}_1,\cdots,\mathbf{s}_n : \mathbf{s}_i\in\mathbb{R}^d\}$, the joint distribution of $\mathbf{z}=\{z(\mathbf{s}_1),\cdots,z(\mathbf{s}_n)\}$ follows a multivariate Gaussian distribution. A GP can be completely specified by its *mean function* $\mu(\mathbf{s})=E[z(\mathbf{s})]$, and its *covariance function*, $C(\mathbf{s}_i,\mathbf{s}_{i'})=Cov(z(\mathbf{s}_i),z(\mathbf{s}_{i'}))$. The distribution of $\mathbf{z}$ is given by

$$\mathrm{p}(\mathbf{z})\sim(2\pi)^n|\boldsymbol{\Sigma}|^{-1/2}\exp\Big\{-\frac{1}{2}(\mathbf{z}-\boldsymbol{\mu})^\top\boldsymbol{\Sigma}^{-1}(\mathbf{z}-\boldsymbol{\mu})\Big\},$$

where $\boldsymbol{\mu} = (\mu(\mathbf{s}_1), \cdots, \mu(\mathbf{s}_n))^\top$ is the mean vector, and $\boldsymbol{\Sigma}$ is the covariance matrix with elements $\boldsymbol{\Sigma}_{i,i'} = C(\mathbf{s}_i, \mathbf{s}_{i'})$. In order for $\boldsymbol{\Sigma}$ to be a valid covariance matrix, it must be positive definite, and this can be guaranteed by specifying a positive definite covariance function. Generally, the dependence structure and smoothness in $z$ are governed by the choice of the covariance function. With the assumption of second-order stationarity and isotropy, the mean and covariance functions can be simplified as

$$\boldsymbol{\mu}(\mathbf{s}) = \mu, \quad C(\mathbf{s}_i, \mathbf{s}_{i'}) = \frac{1}{\lambda_z}\rho(||\mathbf{h}||),$$

where $\lambda_z$ denotes the marginal precision, $\mathbf{h} = \mathbf{s}_i - \mathbf{s}_{i'}$, and $\rho(\cdot)$ is a correlation function that computes the correlation between $z(\mathbf{s}_i)$ and $z(\mathbf{s}_{i'})$, which depends only on their spatial distance $||\mathbf{h}||$. Common correlation functions include the following classes:

**Power exponential**

$$\rho(u) = \exp(-(u/\phi)^\kappa) \tag{1.17}$$

A isotropic GP based on the power exponential correlation function is mean-square continuous and not mean-square differentiable for all $0 < \kappa < 2$, but infinitely mean-square differentiable when $\kappa = 2$. When $\kappa = 1$, it is a so-called *exponential correlation function*. When $\kappa = 2$, it is called the *Gaussian correlation function*, which is infinitely differentiable.

**Spherical**

$$\rho(u) = \begin{cases} 1 - \frac{3}{2}(u/\phi) + \frac{1}{2}(u/\phi)^3 & \text{if } \phi \leq u \leq \phi \\ 0 & u > \phi \end{cases} \tag{1.18}$$

The spherical correlation function has a finite range, and once differentiable at $u = \phi$.

**Matérn**

$$\rho(u) = \{2^{\kappa-1}\Gamma(\kappa)\}^{-1}(u/\phi)K_\kappa(u/\phi) \tag{1.19}$$

The Matérn correlation function is $\lceil \kappa - 1$ times mean-square differentiable, where $\lceil \kappa - 1$ denotes the smallest integer greater than or equal to $\kappa$.

In general, a stationary stochastic process with correlation function $\rho(u)$ is $\kappa$ times mean-square differentiable if and only if $\rho(u)$ is $2\kappa$ times differentiable at $u = 0$ (Bartlett (1955)). More details on Gaussian processes and modeling of their correlation functions are available by Cressie (1991), Stein (1999), and Banerjee et al. (2003).

### 1.5.2 Modeling and Bayesian Estimation

Suppose that a set of observations $\{y(\mathbf{s}_1), \cdots, y(\mathbf{s}_n)\}$ is collected over a spatial domain $\mathcal{S} \subset \mathbb{R}^d$. In almost any spatial application, each observation $y(\mathbf{s}_i)$ at a location $\mathbf{s}_i \in \mathcal{S}$ is assumed to be a combination of the true value $z(\mathbf{s}_i)$ and some Gaussian noise $\epsilon(\mathbf{s}_i)$, i.e.,

$$y(\mathbf{s}_i) = z(\mathbf{s}_i) + \epsilon(\mathbf{s}_i), \quad i = 1, \cdots, n, \tag{1.20}$$

$$\epsilon(\mathbf{s}_i) \overset{\text{iid}}{\sim} N(0, \phi^{-1}), \quad i = 1, \cdots, n. \tag{1.21}$$

The underlying process $z$ is modeled with a stationary GP, and is achieved by imposing a GP prior on $z$ under the Bayesian setting (details are given in the next section). The likelihood is given by

$$L(\mathbf{z}, \phi|\mathbf{y}) \propto |\mathbf{\Sigma}_y|^{-1/2} \exp\Big\{ -\frac{1}{2}(\mathbf{y}-\mathbf{z})^\top \mathbf{\Sigma}_y^{-1}(\mathbf{y}-\mathbf{z})\Big\}, \tag{1.22}$$

where $\mathbf{y} = (y(\mathbf{s}_1), \cdots, y(\mathbf{s}_n))^\top$, $\mathbf{z} = (z(\mathbf{s}_1), \cdots, z(\mathbf{s}_n))^\top$, $\mathbf{\Sigma}_y = \phi^{-1}\mathbf{I}_n$, and $\mathbf{I}_n$ denotes the $n \times n$ identity matrix. The goal is to estimate the underlying process $z$ and the error precision $\phi$. Some common methods include Maximum Likehood Estimation, Restricted Maximum Likehood, and Bayesian inference. In the Bayesian setting (which is also the main tool of inference in this dissertation), the underlying process $z$ is given a zero-mean Gaussian process prior, and the error precision is given a Gamma distributed prior. That is,

$$\begin{aligned} P(\mathbf{z}) &= (2\pi)^{-n/2}|\mathbf{\Sigma}_z|^{-1/2} \exp\Big\{ -\frac{1}{2}\mathbf{z}^\top \mathbf{\Sigma}_z^{-1}\mathbf{z}\Big\}, & (1.23)\\ P(\phi) &= \frac{{b_y}^{a_y}}{\Gamma(a_y)}\phi^{a_y-1}\exp\{-b_y\phi\}, & (1.24) \end{aligned}$$

where $\mathbf{\Sigma}_z$ is the prior covariance matrix and is specified based on a correlation function such as those in Section 1.4. The resulting conditional posterior distributions are given by

$$\begin{aligned} P(\mathbf{z}|\mathbf{y}, \phi) &\propto L(\mathbf{z}, \phi|\mathbf{y})P(\mathbf{z}) \\ &\propto \exp\Big\{ -\frac{1}{2}\mathbf{z}^\top(\mathbf{\Sigma}_y^{-1} + \mathbf{\Sigma}_z^{-1})\mathbf{z} + \mathbf{z}^\top\mathbf{\Sigma}_y^{-1}\mathbf{y}\Big\}, & (1.25)\\ P(\phi|\mathbf{y}, \mathbf{z}) &\propto L(\mathbf{z}, \phi|\mathbf{y})P(\phi) \\ &\propto \phi^{n/2+a_y-1}\exp\Big\{ -\phi\Big(\frac{1}{2}(\mathbf{y}-\mathbf{z})^\top(\mathbf{y}-\mathbf{z}) + b_y\Big)\Big\}. & (1.26) \end{aligned}$$

Note that the priors are conjugate and the resulting posterior distributions are in the same family of the corresponding priors:

$$\mathbf{z}|\mathbf{y},\phi \sim N((\mathbf{\Sigma}_y^{-1}+\mathbf{\Sigma}_z^{-1})^{-1}\mathbf{\Sigma}_y^{-1}\mathbf{y}\ ,\ (\mathbf{\Sigma}_y^{-1}+\mathbf{\Sigma}_z^{-1})^{-1}), \tag{1.27}$$

$$\phi|\mathbf{y},\mathbf{z} \sim G(n/2+a_y\ ,\ 0.5((\mathbf{y}-\mathbf{z})^\top(\mathbf{y}-\mathbf{z}))+b_y), \tag{1.28}$$

where $G$ denotes the Gamma distribution. Note that the posterior mean of $\mathbf{z}$ is a weighted average of the prior mean (0 in this case) and the data. The Gibbs sampler can be used to obtain posterior samples of $\mathbf{z}$ and $\phi$.

## 1.6 Process Convolution Gaussian Process Models

### 1.6.1 Process Convolutions

An alternative specification of a stationary (second-order) Gaussian process can be done via process convolutions (Higdon, 1998, 2002, 2005; Kern, 2000) as follows,

$$z(\mathbf{s}) = \int_{\mathcal{S}} k(\mathbf{u}-\mathbf{s})x(\mathbf{u})d\mathbf{u}, \quad \mathbf{s},\mathbf{u}\in\mathcal{S}\subseteq\mathbb{R}^d, \tag{1.29}$$

where $x(\cdot)$ is a White noise process with mean zero and precision $\lambda$, and $k(\cdot)$ is a symmetric kernel (e.g, Gassusian) defined over $\mathcal{S}\subseteq\mathbb{R}^d$. The expectation and covariance of $z$ under this construction are given by

$$E[z(\mathbf{s})] = \int_{\mathcal{S}} k(\mathbf{u}-\mathbf{s})E[x(\mathbf{u})]d\mathbf{u}, \tag{1.30}$$

$$
\begin{aligned}
& Cov(z(\mathbf{s}_i), z(\mathbf{s}_{i'})) \\
= \;& E[(z(\mathbf{s}_i) - E[z(\mathbf{s}_i)])(z(\mathbf{s}_{i'}) - E[z(\mathbf{s}_{i'})])] \\
= \;& E[z(\mathbf{s}_i)z(\mathbf{s}_{i'})] - E[z(\mathbf{s}_i)E[z(\mathbf{s}_{i'})]] - E[z(\mathbf{s}_{i'})E[z(\mathbf{s}_i)]] + E[E[z(\mathbf{s}_i)]E[z(\mathbf{s}_{i'})]] \\
= \;& E[z(\mathbf{s}_i)z(\mathbf{s}_{i'})] - E[z(\mathbf{s}_i)]E[z(\mathbf{s}_{i'})] \\
= \;& E\left[\int_{\mathcal{S}} k(\mathbf{u}_j - \mathbf{s}_i)x(\mathbf{u}_j)d\mathbf{u}_j \int_{\mathcal{S}} k(\mathbf{u}_{j'} - \mathbf{s}_{i'})x(\mathbf{u}_{j'})d\mathbf{u}_{j'}\right] - \\
& E\left[\int_{\mathcal{S}} k(\mathbf{u}_j - \mathbf{s}_i)x(\mathbf{u}_j)d\mathbf{u}_j\right] E\left[\int_{\mathcal{S}} k(\mathbf{u}_{j'} - \mathbf{s}_{i'})x(\mathbf{u}_{j'})d\mathbf{u}_{j'}\right] \\
= \;& \int_{\mathcal{S}}\int_{\mathcal{S}} k(\mathbf{s}_i - \mathbf{u}_j)k(\mathbf{s}_{i'} - \mathbf{u}_{j'})Cov(x(\mathbf{u}_j), x(\mathbf{u}_{j'}))d\mathbf{u}_j d\mathbf{u}_{j'}. \qquad (1.31)
\end{aligned}
$$

When $x$ is a White noise process with precison $\lambda$, the expectation is 0 and the covariance can be written as

$$
\begin{aligned}
Cov(\mathbf{h}) &= Cov(z(\mathbf{s}_i), z(\mathbf{s}_{i'})) \\
&= \lambda^{-1}\int_{\mathcal{S}} k(\mathbf{u} - \mathbf{s}_i)k(\mathbf{u} - \mathbf{s}_{i'})d\mathbf{u} \\
&= \lambda^{-1}\int_{\mathcal{S}} k(\mathbf{u} - \mathbf{h})k(\mathbf{u})d\mathbf{u}, \quad \mathbf{h} = \mathbf{s}_i - \mathbf{s}_{i'}. \qquad (1.32)
\end{aligned}
$$

That is, the covariance depends only on the vector difference $\mathbf{h} = \mathbf{s}_i - \mathbf{s}_{i'}$, and is actually the result of convolving the kernel with itself. As stated in Kern (2000), if $\mathcal{S} = \mathbb{R}^d$ and $k(\mathbf{s})$ is isotropic, $Cov(\mathbf{h})$ depends only on the magnitude of $\mathbf{h}$, and the resulting $z(\mathbf{s})$ becomes isotropic. In this case, there exists a one-to-one relationship between the kernel $k(\mathbf{s})$ and $Cov(\mathbf{h})$, provided that either $\int_{\mathbb{R}^d} k(\mathbf{s})d\mathbf{s} < \infty$ and $\int_{\mathbb{R}^d} k^2(\mathbf{s})d\mathbf{s} < \infty$ or $Cov(\mathbf{s})$ is integrable and positive definite. This relationship is governed by the convolution theorem for Fourier transforms (Barry and Ver Hoef, 1996).

In practice, we would approximate the above convolution using a finite set

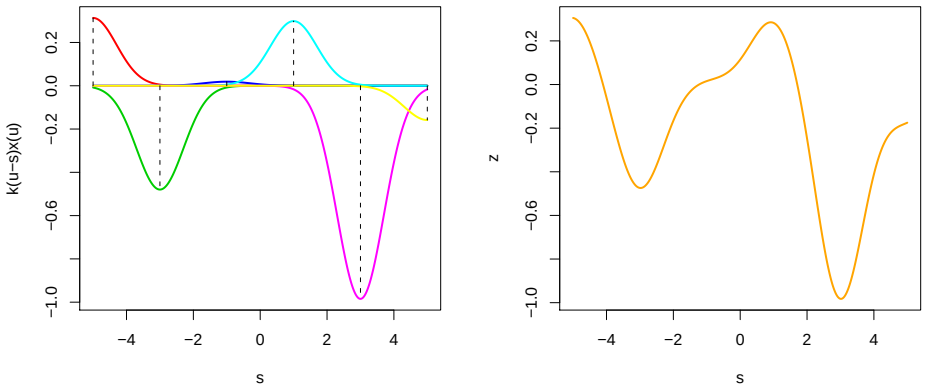


Figure 1.1: An example of a 1-d process convolution GP (*right*) constructed by summing six scaled Gaussian kernels (*left*)

of regularly spaced basis points $\mathbf{u}_1, \cdots, \mathbf{u}_m \in \mathcal{S}$ with $x(\mathbf{u}_j) \sim N(0, \lambda^{-1})$. A discrete approximation of $z(\mathbf{s})$ can be obtained as

$$z(\mathbf{s}) \approx \sum_{j=1}^{m} k(\mathbf{u}_j - \mathbf{s}) x(\mathbf{u}_j). \tag{1.33}$$

The expectation and covariance is given by

$$E[z(\mathbf{s}_i)] = \sum_{j=1}^{m} k(\mathbf{u}_j - \mathbf{s}_i) E[x(\mathbf{u}_j)] = 0, \tag{1.34}$$

$$Cov(z(\mathbf{s}_i), z(\mathbf{s}_{i'})) = \lambda^{-1} \sum_{j=1}^{m} k(\mathbf{u}_j - \mathbf{s}_i) k(\mathbf{u}_j - \mathbf{s}_{i'}). \tag{1.35}$$

Note that in Equation (1.35), the discrete convolution depends on the locations $\mathbf{s}_i$ and $\mathbf{s}_{i'}$ but not their vector distance. This nonstationarity is an artifact created by the discretization of the $\mathbf{u}$ coordinates. An example is shown in Figure 1.1, where the left panel shows six Gaussian kernels with different heights, and the right panel is a GP that results from summing these kernels. In almost any applications, only a finite set of

spatial sites $\mathbf{s}_1, \cdots, \mathbf{s}_n \in \mathcal{S}$ are of interest. In such cases, the above representation can be written in matrix form,

$$\mathbf{z} = \mathbf{K}\mathbf{x} \tag{1.36}$$

where $\mathbf{z} = (z(\mathbf{s}_1), \cdots, z(\mathbf{s}_n))^\top$, $\mathbf{x} = (x(\mathbf{u}_1), \cdots, x(\mathbf{u}_m))^\top$, and $\mathbf{K}$ is a $(n \times m)$ matrix with elements $\mathbf{K}_{ij} = k(\mathbf{u}_j - \mathbf{s}_i)$. The $(n \times n)$ covariance matrix $\mathbf{\Sigma}_z$ of $\mathbf{z}$ depends on the precision $\lambda$ and the kernel matrix $\mathbf{K}$: $\mathbf{\Sigma}_z = \lambda^{-1}\mathbf{K}\mathbf{K}^\top$. In general, the resolution of the resulting Gaussian process increases as the number of $\mathbf{u}_j$s increases. However, once a reasonable saturation is reached, there is diminishing return in adding more background points. Moreover, decreasing the size of the kernel decreases the smoothness and range of dependence of the resulting GP. A rule of thumb given by Higdon is to allow the marginal standard deviation of the kernel equal to the spacing between any two adjacent background points.

### 1.6.2 Modeling and Bayesian Estimation

Suppose that we have a set of observations $\{y(\mathbf{s}_i), i = 1, \cdots, n\}$. A process convolution GP model can be constructed as

$$\begin{aligned} y(\mathbf{s}_i) &= z(\mathbf{s}_i) + \epsilon(\mathbf{s}_i), \quad i = 1, \cdots, n, && (1.37)\\ z(\mathbf{s}_i) &= \sum_{j=1}^{m} k(\mathbf{u}_j - \mathbf{s}_i)x(\mathbf{u}_j), && (1.38)\\ \epsilon(\mathbf{s}_i) &\overset{\text{iid}}{\sim} N(0, \phi^{-1}), \quad i = 1, \cdots, n. && (1.39) \end{aligned}$$

The data is assumed to be an additive combination of the true underlying process $z$ and some Gaussian noise $\epsilon$. The underlying process $z$ is formulated via the process

convolution approach. The above model can be written in matrix form:

$$\mathbf{y} = \mathbf{K}\mathbf{x} + \boldsymbol{\epsilon}, \tag{1.40}$$

where $\mathbf{y} = (y(\mathbf{s}_1), \cdots, y(\mathbf{s}_n))^\top$, $\mathbf{x} = (x(\mathbf{u}_1), \cdots, x(\mathbf{u}_m))^\top$, $\boldsymbol{\epsilon} = (\epsilon(\mathbf{s}_1), \cdots, \epsilon(\mathbf{s}_n))^\top$, and $\mathbf{K}$ is a kernel matrix. The likelihood is given by

$$L(\mathbf{x}, \phi|\mathbf{y}) \propto \phi^{n/2} \exp\Big\{ -0.5\phi(\mathbf{y} - \mathbf{K}\mathbf{x})^\top(\mathbf{y} - \mathbf{K}\mathbf{x})\Big\}. \tag{1.41}$$

In the Bayesian setting, a Gaussian prior is imposed on $x$, and Gamma priors are imposed on the precision parameters:

$$\begin{aligned} P(\mathbf{x}|\lambda) &\propto \lambda^{m/2} \exp\{-0.5\lambda \mathbf{x}^\top \mathbf{x}\}, & (1.42)\\ P(\lambda) &\propto \lambda^{a_x - 1} \exp\{b_x \lambda\}, & (1.43)\\ P(\phi) &\propto \phi^{a_y - 1} \exp\{b_y \phi\}. & (1.44) \end{aligned}$$

The resulting conditional posterior distributions are given by

$$\begin{aligned} \mathbf{x}|\lambda, \phi, \mathbf{y} &\sim N((\phi\mathbf{K}^\top\mathbf{K} + \lambda\mathbf{I}_m)^{-1}\phi\mathbf{K}^\top\mathbf{y}\ ,\ (\phi\mathbf{K}^\top\mathbf{K} + \lambda\mathbf{I}_m)^{-1}), & (1.45)\\ \lambda|\mathbf{x}, \mathbf{y} &\sim G(m/2 + a_x\ ,\ 0.5\mathbf{x}^\top\mathbf{x}), & (1.46)\\ \phi|\mathbf{x}, \mathbf{y} &\sim G(n/2 + a_y\ ,\ 0.5(\mathbf{y} - \mathbf{K}\mathbf{x})^\top(\mathbf{y} - \mathbf{K}\mathbf{x}) + b_y). & (1.47) \end{aligned}$$

The Gibbs sampler can be used to obtain posterior samples of $\mathbf{x}$, $\lambda$, and $\phi$.

## 1.7 Treed Models

In general, treed models partition the predictor space into disjoint subsets in which the distribution of the response variable becomes more homogeneous and a simpler

submodel can be fitted in each partition. A popular example is the Classification and Regression Trees (CART) by Breiman et al. (1984). Bayesian formulations of CART can be found in the papers by Chipman et al. (1998) and Denison et al. (1998). The model presented in the rest of the dissertation follows the tree generating process by Chipman et al. (1998). Specifically, partitioning results in a binary tree structure such that each internal node is associated with a splitting rule that determines the location of partitioning. Each measurement of the response variable is assigned to only one of the terminal nodes (resulting partitions) where a conditional distribution is determined. Consider a problem with a $(n \times 1)$ observation vector $\mathbf{y}$ and a $(n \times p)$ design matrix $\mathbf{F}$ as in the linear regression setting. The partitioning process proceeds by choosing one of the terminal nodes, an available predictor from the chosen terminal node, and makes a split at one of the available values of the chosen predictor. If the model splits at value $v$ of the $l^{th}$ predictor, it assigns to the left child node the subset of $\{\mathbf{F}, \mathbf{y}\}$ in which values of the $l^{th}$ predictor is $\leq v$, and to the right child node the complement set of the data. Recursively splitting in this process results in a binary tree $\mathcal{T}$ with $b$ terminal nodes and each terminal node is associated with a subset, denoted by $\{\mathbf{F}_\nu, \mathbf{y}_\nu\}$, of the original data set $\{\mathbf{F}, \mathbf{y}\}$ such that $\mathbf{F} = (\mathbf{F}_1^\top, \cdots, \mathbf{F}_b^\top)^\top$ and $\mathbf{y} = (\mathbf{y}_1^\top, \cdots, \mathbf{y}_b^\top)^\top$. An example of treed models is shown in Figure 1.2. The tree $\mathcal{T}$ associates a separate model for each partition, that is, the sampling distribution of $\mathbf{y}_\nu$ can be described by a separate parametric model $P(\mathbf{y}_\nu|\mathbf{F}_\nu, \boldsymbol{\theta}_\nu, \mathcal{T})$, where $\boldsymbol{\theta}_\nu$ denotes the parameter vector associated with partition $\nu$. Let $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \cdots, \boldsymbol{\theta}_b)$, a treed model is fully characterized by $(\boldsymbol{\Theta}, \mathcal{T})$. An important assumption of this type of treed model is that data within the

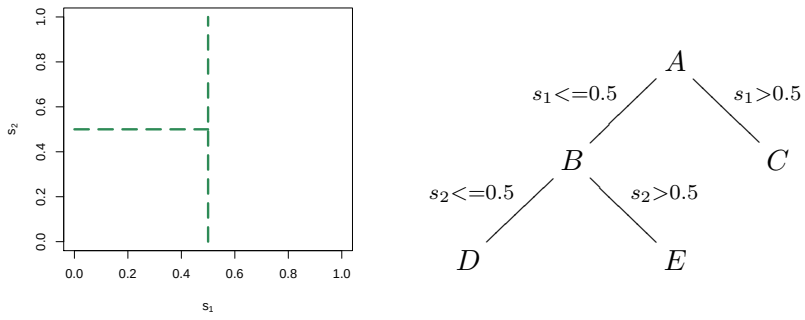


Figure 1.2: An exmaple of binary treed partitioning over a 2-d domain

same partition are i.i.d. and data across partitions are independent. Hence, the sampling distribution of $\mathbf{y}$ can obtained as

$$P(\mathbf{y}|\mathbf{F}, \boldsymbol{\Theta}, \mathcal{T}) = \prod_{\nu=1}^{b} P(\mathbf{y}_\nu|\mathbf{F}_\nu, \boldsymbol{\theta}_\nu, \mathcal{T}).$$

Implemention of treed models via a Bayesian approach requires prior specification for $(\boldsymbol{\Theta}, \mathcal{T})$. A convenient way would be to use the relationship $P(\boldsymbol{\Theta}, \mathcal{T}) = P(\boldsymbol{\Theta}|\mathcal{T})\mathrm{p}(\mathcal{T})$ and specify the tree prior $P(\mathcal{T})$ and the conditional parameter prior $P(\boldsymbol{\Theta}|\mathcal{T})$ separately. Following Chipman et al. (1998), $P(\mathcal{T})$ is specified through an implicit tree generating process so that each realization from this process is considered a random draw from this prior; $P(\boldsymbol{\Theta}|\mathcal{T})$ is specified by imposing conjugate priors on terminal node parameters while assuming conditional independence of parameters across terminal nodes. More details of prior specification will be discussed in the formulation of the treed process convolution GP model in the next chapter.

Recent work making use of Bayesian binary treed models includes fitting a linear model (Chipman et al., 2002), or a Gaussian process model (Gramacy and Lee,

2008) in each partition. Alternatively, the predictor space can be partitioned via a Voronoi tessellation. An example of such can be found in the work of Kim et al. (2005), which fits a piecewise GP within each Voronoi partition. However, using Voronoi tessellation may lead to complex partitioning of the predictor space and produces a final model that might be difficult to interpret. Binary treed models tend to produce fewer number of partitions leading to a simpler final model.

# Chapter 2

# Simulation Studies on Process Convolution GP Models

## 2.1 Introduction

Gaussian process (GP) models are widely used for statistical modeling of point-referenced and functional data in many scientific applications. While they are powerful models for their nonparametric flexibility, the trade-off is increasing computing requirements as the sample size increases. As shown in Section 1.6, a process convolutions representation provides a computationally efficient alternative for lower-dimensional problems (Higdon, 1998, 2002). This convolution approach requires additional specification in implementation, in particular, a kernel function $k(\cdot)$, and a basis grid $\{\mathbf{u}_1, \cdots, \mathbf{u}_m\}$ (discrete) for defining the latent process $x(\mathbf{u}_j)$. In general, the kernel shape and size is related to the smoothness and range of dependence of the process being modeled,

respectively. Thus the kernel choice depends on the application at hand, and a rule of thumb given by Hidgon for Gaussian kernels is to set the marginal standard deviation equal to the spacing between two adjacent basis points (assuming that the basis points are regularly spaced). However, little is known about how to choose the grid size $m$. In general, if this grid is dense enough, then the approximation will be good, otherwise, the approximation may be quite poor. The problem exhibits a saturation effect, so using additional points tends to have little additional benefit. Thus it would be helpful to establish rules of thumb for choosing the grid size $m$ in order to maintain accurate approximations without using too many unnecessary points. This chapter presents a series of simulation studies to inform this question.

## 2.2 Simulation setup

Three widely used kernels are considered in the study: *Gaussian*, *Bézier*, and *Exponential*. Other examples of kernel functions for process convolutions can be found in Kern (2000). For consistency, these kernels are parametrized in terms of the Mahalanobis distance $D_M(\mathbf{u}, \mathbf{s}, \mathbf{Q}) = \sqrt{(\mathbf{u}-\mathbf{s})^\top \mathbf{Q}(\mathbf{u}-\mathbf{s})}$, where $\mathbf{Q}^{-1}$ denotes the covariance matrix. These three kernels are defined as follows.

**Gaussian**:

$$k(\mathbf{u}-\mathbf{s};\mathbf{Q}) = (2\pi)^{-d/2}|\mathbf{Q}^{-1}|^{-1/2}\exp\Big\{-\frac{1}{2}D_M(\mathbf{u},\mathbf{s},\mathbf{Q})^2\Big\}, \tag{2.1}$$

where the resulting GP under this kernel is infinitely differentiable.

**Bézier**:

$$k(\mathbf{u}-\mathbf{s};\mathbf{Q}) = \begin{cases} (1 - D_M(\mathbf{u},\mathbf{s},\mathbf{Q})^2)^\kappa & \text{if } D_M(\mathbf{u},\mathbf{s},\mathbf{Q}) < 1 \\ 0 & \text{otherwise,} \end{cases} \tag{2.2}$$

where $\kappa$ is a smoothness parameter such that the resulting GP is $2\kappa$ differentiable (Brenning, 2001). Using a compactly supported kernel ensures that $z(\mathbf{s})$ is related only to the nearby values. In the matrix representation of the model (see Section 1.6), since $\mathbf{K}_{ij} = k(\mathbf{u}_j - \mathbf{s}_i)$, using a compactly supported kernel makes $\mathbf{K}$ a sparse matrix. Dedicated sparse matrix computation methods can be used to speed up computation of the likelihood.

**Exponential**:

$$k(\mathbf{u}-\mathbf{s};\mathbf{Q}) = \exp\Big\{ - D_M(\mathbf{u},\mathbf{s},\mathbf{Q})\Big\}. \tag{2.3}$$

Note that unlike the Gaussian kernel which induces a Gaussian correlation function (Kern, 2000), the induced correlation function of the Exponential kernel can not be obtained in closed form. In fact, the induced correlation function is differentiable at the origin, different from the standard Exponential correlation function which is not differentiabe at the origin.

As mentioned in (Kern, 2000), the correlation structure of $z$ constructed from continuous process convolutions is proportional to the convolution of the kernel with

itself (provided that a White noise process is specified for $x$). In fact, for any kernel, the corresponding correlation function can be obtained by convolving the kernel with itself and rescaling. Examples of correlation functions induced by the Gaussian, Bézier ($\kappa = 3$), and Exponential kernels are shown in Figure 2.1. Only isotropic kernels are considered for the simulation study. Anisotropy can be accounted by rotating and stretching the coordinate axes of the spatial domain. For isotropic kernels, the covariance matrix $\mathbf{Q}^{-1}$ has identical values along the diagonal and zeros elsewhere, and the diagonal elements determine the size of the kernel. Specifically, kernel size in the rest of this chapter is defined to be the square root of the diagonal element of $\mathbf{Q}^{-1}$. For instance, the kernel size for a Gaussian kernel is the marginal standard deviation. For the Bézier kernel, the kernel size equals to the radius of the compact support.

To carry out the simulations, a set of 1-dimensional data on the interval $[0, 1]$, and a set of 2-d data over the $[0, 1]^2$ domain are generated. In each case, 200 data points are generated by adding $N(0, \text{sd} = 0.01)$ noise to a simulated Gaussian process based on the *Matérn ($\kappa = 7$)*, *Spherical*, and *Exponential* correlation functions, and correlation functions induced by the three kernels mentioned above. These correlation functions are shown in Figure 2.1 with practical ranges (i.e., the distance at which the correlation = 0.05) set to 0.1, 0.2, and 0.3. The 1-d data and the interpolated surface of the 2-d data are shown in Figures 2.2 and 2.3 for the Gaussian and Exponential correlation functions, and those induced by the Bézier ($\kappa = 3$) and Exponential kernels. The Matérn ($\kappa = 7$) and Spherical data are not shown since they resemble the Gaussian and Exponential data, respectively.

The default kernel size used in the study is the one such that the practical range of the resulting correlation function matches that of the data. The effect of having a kernel larger and smaller than the default size is also of interest in the study. Thus each simulation is performed for a set of five different kernel sizes determined by multiplying the default size with $\{0.6, 0.8, 1, 1.5, 2\}$. The number of bases used in the study are $\{5, 6, \cdots, 50\}$ for the 1-d simulations, and $\{2^2, 3^2, \cdots, 30^2\}$ for the 2-d simulations. The significant increase in the computational time for using more than $30^2$ bases prevents us from doing so since there is a large set of repetitive 2-d simulations. A Bayesian approach is used for inference. Under the usual model setup, a Gamma distribution $G(1, 0.001)$ is specified as the prior, which is fairly noninformative, for both the measurement error precision $\phi$ and the background points precision $\lambda$.

In order to lower the variation in the simulations, a 10-fold cross-validation approach is taken by randomly splitting the 200 observations into 10 groups of equal size. Then, 9 of the 10 groups are chosen to be the training dataset to which the model is applied, while the other group is treated as a validation set on which prediction is evaulated. This forms a total of 10 independent simulations. This set of simulations is run in R (R Development Core Team, 2011) and is repeated for 10 different starting random seeds. In total, for each combination of kernel and correlation function, there are 100 simulations whose model fitting and prediction performances would be averaged. The log likelihood is the first model fitting performance meansure considered. However, it is an increasing function of the number of parameters, that is, increasing $m$ will always improve the model fitting, but also increases the chance of overfitting. The Bayesian

information criterion (BIC) can be used to address the potential problem of overfitting. However, the penalty term is dominating due to the large number of bases in 2-d, which renders BIC not useful for identifying the sufficient number of bases. Alternatively, the Deviance information criterion (DIC) is considered, which is more suitable for Bayesian model selection problems. For most simulations, the curve of DIC v.s. the number of bases shows a saturation behavior, and near the saturation point can be considered where the model has reached a stage such that further increase in $m$ results in negligible improvement in the model fitting. This point of saturation will be reffered as the *critical point* in the rest of this chapter. The sufficient number of bases is chosen to be the one associated with the critical point. For convenience, $\hat{m}$ denotes the sufficient number of bases. The prediction performance is evaluated based on the mean squared error (MSE) resulted by predicting against the validation dataset. For each combination of kernel and correlation function, both DIC and MSE are obtained from averaging the 100 simulations. To further reduce the variation in the averaged values, a five-period exponential moving average is used to smooth out the DIC and MSE curves. As a result, any value on the resulting DIC and MSE curve is an average of the original value at that point and the previous four values to the left, except for the first four points where the original values are retained. Discussion of the results are given in the following sections.

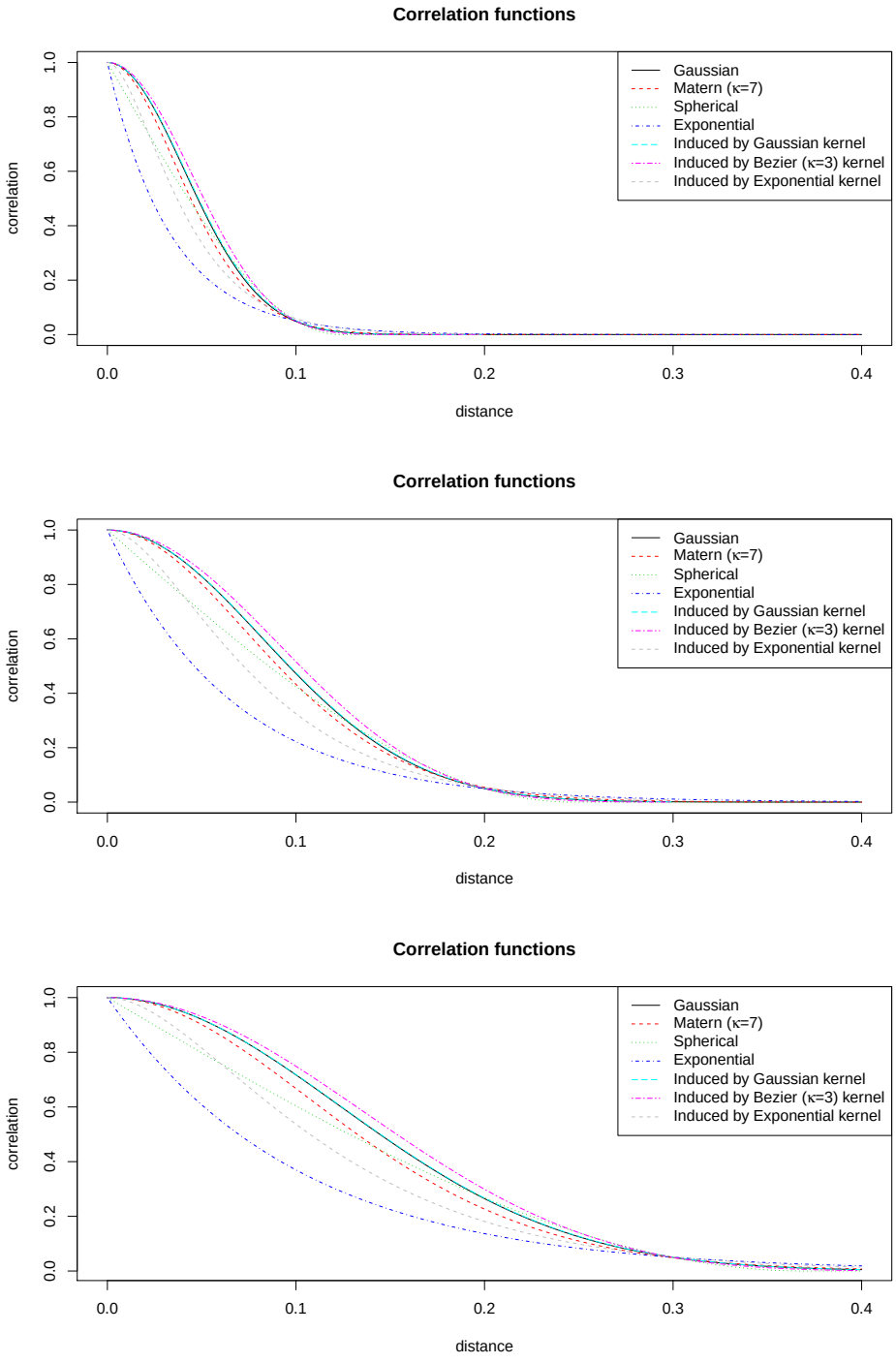


Figure 2.1: Correlation functions with practical range 0.1 (*top*), 0.2 (*middle*), and 0.3 (*bottom*)

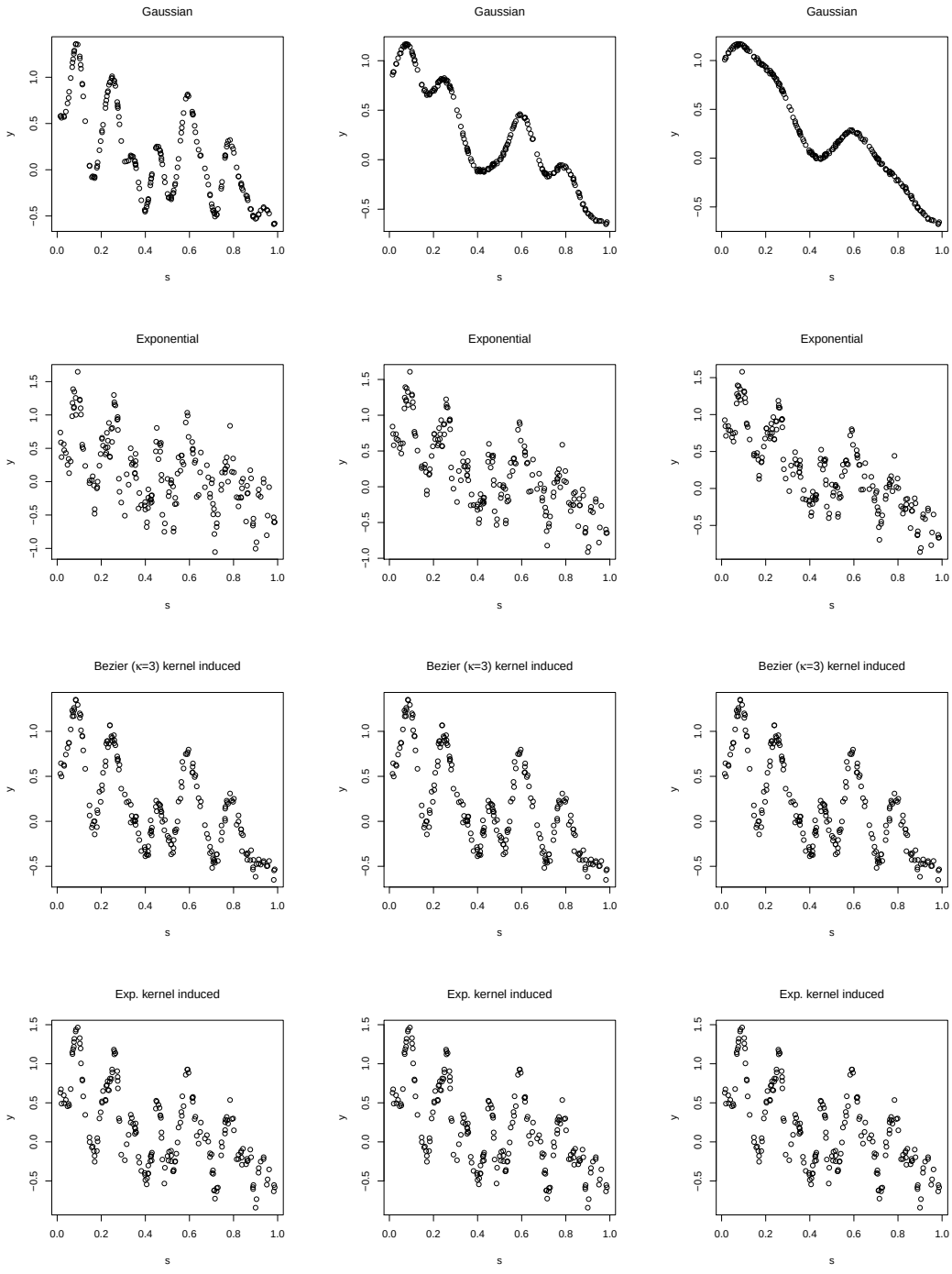


Figure 2.2: 1-d Gaussian data generated based on the Gaussian (*top*) and Exponential (*second row*) correlation functions, and those induced by the Bézier ($\kappa = 3$) (*third row*) and Exponential (*bottom*) kernels, with practical ranges = 0.1 (*left*), 0.2 (*center*), and 0.3 (*right*)

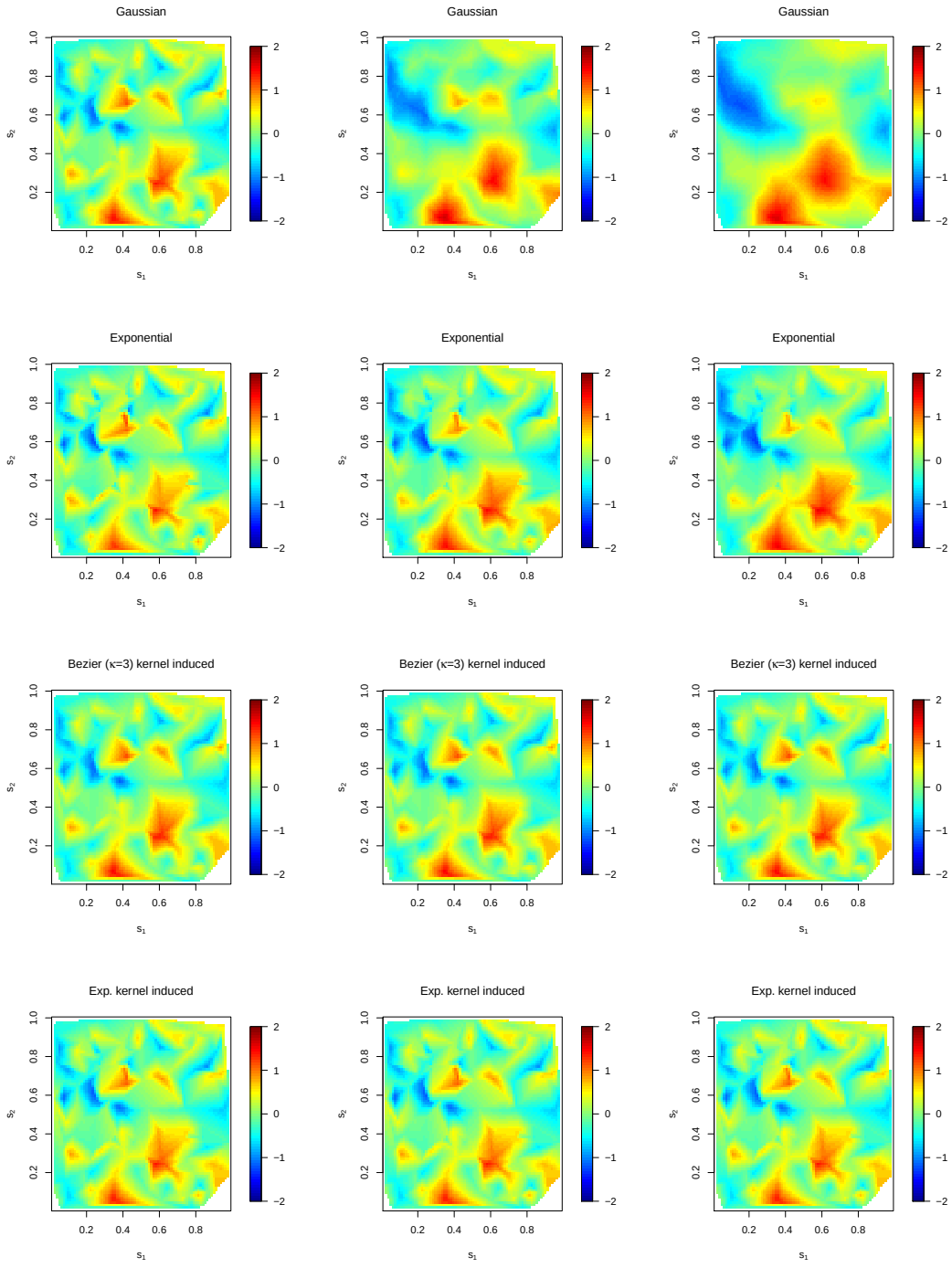


Figure 2.3: Interpolated 2-d Gaussian realizations generated based on the Gaussian (*top*) and Exponential (*second row*) correlation functions, and those induced by the Bézier ($\kappa = 3$) (*third row*) and Exponential (*bottom*) kernels, with practical ranges = 0.1 (*left*), 0.2 (*center*), and 0.3 (*right*)

## 2.3 Simulation results

The model performance measures (DIC and MSE) are plotted against $m$ (see Figures 2.4 through 2.15) for cases where the practical range of the true process is 0.1, 0.2, and 0.3. In almost all cases, the DIC decreases initially as $m$ increases, then saturates after a certain $m$ is reached. The critical point of DIC for some of the 2-d datasets with practical range 0.1 is not very obvious because the maximum number of bases is limited to $30^2$ for reasons explained above. However, the omitted cases do not prevent us from determining a rule of thumb for the sufficient number of bases.

### 2.3.1 Gaussian kernel

First, the Gaussian kernel is applied to 1-d and 2-d datasets generated based on the Gaussian correlation function. The model performance meaures are shown in Figure 2.4. For both 1-d and 2-d results, the lowest critical point of the DIC curves corresponds to the default kernel size (whose practical range matches that of the data). This makes sense because the induced correlation function from a Gaussian kernel is exactly the Gaussian correlation function. MSE curves decrease as $m$ increases, and saturate after the critical point. An exception occurs for the 2-d case when the practical range is 0.1, where the MSE curves are non-decreasing and that of the largest kernel is lower than the others. A possible reason is that the 2-d data with practical range = 0.1 is quite scattered like random noise, so fitting a relatively flat surface is more preferred by the model. In general, the further away the kernel size is from the default, the worse

the model fitting and prediction performance.

For comparison, the Gaussian kernel is also applied to data generated under the Matérn ($\kappa = 7$), Spherical, and Exponential kernel. Results are shown in Figure 2.5, 2.6, and 2.7, respectively. In the Matérn case, the lowest critical point corresponds to a kernel with size equal to 0.8 or 0.6 times the default size. This is expected because a process based on the Gaussian correlation function is infinitely differentiably, whereas a process based on the Matérn ($\kappa = 7$) correlation function is only 6 times differentiable. Moreover, the Matérn ($\kappa = 7$) correlation function decreases faster than that of the Gaussian, therefore shrinking the Gaussian kernel is needed to better capture the true correlation structure. Nonetheless, the corresponding $\hat{m}$'s are similar to those of the Gaussian data. Also, the MSE curves show that prediction performance is fairly stable when the kernel induced correlation function is similar to the truth.

For the Spherical and Exponential data, the lowest critical point corresponds to a kernel with size equal to 0.4 times the default size. This is reasonable since the Spherical and Exponential correlation functions are non-differentiable, and the corresponding datasets look fairly scattered. Therefore, a much smaller Gaussian kernel and alot more bases are needed to better capture the high variation data. In the 1-d case, the MSE has a similar decreasing behavior below the critical point, and the curve that corresponds to the smallest kernel scale has the lowest MSE after the critical point. In the 2-d case, the opposite is true, i.e., lower MSE pairs with larger kernel. A possible reason might be that learning the additional dimension of information in the 2-d case by shrinking the Gaussian kernel easily causes over-fitting and bad predictions. In any

case, just as what this example shows, it is not a good idea to use a kernel whose induced correlation function differs much from the truth.

### 2.3.2 Bézier kernel

Next, the Bézier ($\kappa = 3$) kernel is applied to 1-d and 2-d datasets generated based on the Bézier ($\kappa = 3$) kernel. The model performance meaures are shown in Figure 2.8. For 1-d results, the DIC values are generally very similar after the critical point regardless of the the kernel size. As expected, the lowest critical point actually corresponds to the default kernel size. In the 2-d case, the default kernel size provides a better balance between model fitting (DIC) and prediction (MSE), therefore is used to find $\hat{m}$.

Results based on the Matérn ($\kappa = 7$), Spherical, and Exponential correlation function are shown in Figure 2.9, 2.10, and 2.11. The corresponding model performance resemble those based on the Gaussian kernel. However, some of the DIC's for the 1-d cases are higher or lower than they should be (e.g., when the kernel scale is 1.5 and 2, and the practical range is 0.3 in the Matérn case). This instability might be a consequence of the compact support of the kernel because the number of data points covered by the kernel centered at each basis location may not be the same. DIC's from 2-d are more stable than those from 1-d, possibly because a 2-d kernel at each basis point overlaps with more surrounding kernels, which makes the model more stiff and offsets the instability caused by the compact support. Nonetheless, the fact that the model finds a similar $\hat{m}$ for kernel sizes close to the default satisfies our main interest.

### 2.3.3 Exponential kernel

Lastly, the Exponential kernel is applied to 1-d and 2-d datasets generated based on a correlation function induced by the Exponential kernel, Matérn ($\kappa = 7$), Spherical, and Exponential correlation function. The model performance measures are shown in Figure 2.12, 2.13, 2.14, and 2.15, respectively. Considering the Maťern case, both 1-d and 2-d simulations show that larger kernels provide better model fitting and prediction. This is because Matérn ($\kappa = 7$) correlation function generally has a higher magnitude than the the correlation function induced by the Exponential kernel at the same distance, hence requiring a larger size than the default.

Simulation results for the other three correlation structures are similar to each other. The 1-d results show that the DIC critical points are very similar to one another, and the DIC's are almost the same after the critical point regardless of the kernel size. However, 2-d results show that the smaller kernel sizes produce lower DIC's, whereas the MSE's are quite high for those small kernels. This battle between model fitting and prediction might suggest that process convolution GP model is not suitable for data with low smoothness. Whenever the data appears to be scattered, the model would desire smaller kernels and more bases. But unless the kernel really captures the true process, decreasing the kernel size and increasing the number of bases is likely to cause overfitting and bad prediction performance. From a conceptually point of view, process convolution GP is unlikely to well capture a process with very low smoothness because it acts like a moving average or low-pass filter such that the high frequency components

in the data are ignored.

### 2.3.4 A rule of thumb for the number of bases

Recall that the main goal of this study is to come up with a rule of thumb for setting the number of bases for applying process convolution GP model. A summary of the above simulations are given in Figure 2.16 and 2.17, where the *basis spacings* at the critical points are plotted against the *practical ranges* of the correlation functions from which the datasets are generated. Simulations corresponding to the Exponential and Spherical correlation functions, and Exponential kernel, are excluded because the focus is on cases where the kernel induced correlation function is fairly similar to the true correlation function. Having a large difference between the two is discouraged in practice. The smallest practical range tested in the 1-d simulations is 0.05, whereas that of the 2-d simulations is 0.1, because a smaller range requires alot more bases which significantly slows down the 2-d simulations. For both cases in general, the basis spacing increases (less bases) as the practical range increases. The 1-d results suggest a smaller basis spacing than that of the 2-d results for the same practical range. This not surprising because increasing $m$ is more likely to overfit 1-d data in practice. On the other hand, the behavior of the curves for the larger kernel sizes in 2-d are not consistent with those of the smaller kernels. Since the DIC's with respect to these large kernels are fairly high, they can be ignored from the purpose of this study. The rationale is again that a kernel whose induced correlation function similar to the truth should be used in practice. A diagonal line (*gray*) is fitted through the points in each plot. The line used

for 1-d is determined by

$$spacing = 0.01 + 0.1 \times range,$$

and the one used for 2-d is given by

$$spacing = 0.03 + 0.1 \times range.$$

The formation of these functions is solely based on visual judgement and do not represent any optimal fittings. However, the points in the plots seem to be well fitted by these linear functions, thus they can be used as a general rule for choosing the basis spacing (equivalently, the number of bases) when the range is $\geq 0.1$. For a range less than 0.1, further increase in $m$ can lead to better model performance, however, the validity of using process convolutions for modeling such a short range dependence should also be questioned.

## 2.4 Conclusion

In this chapter, a series of simulation studies on process convolution GP models are provided, with a particular interest in finding the sufficient number of bases required for promising model performance. Both 1-d and 2-d simulations are considered based on the Gaussian, Bézier ($\kappa = 3$), and Exponential kernels on datasets generated via several different correlation functions. Various combinations of kernel sizes and number of bases are tested. Results show that for datasets whose estimated range of dependence is $\geq 0.1$, basis spacing (equivalently, the number of basis) can be chosen using a linear

formula. For range $< 0.1$, more bases are needed in general, but also switching to a different model capable of handling short dependence range might be more appropriate.

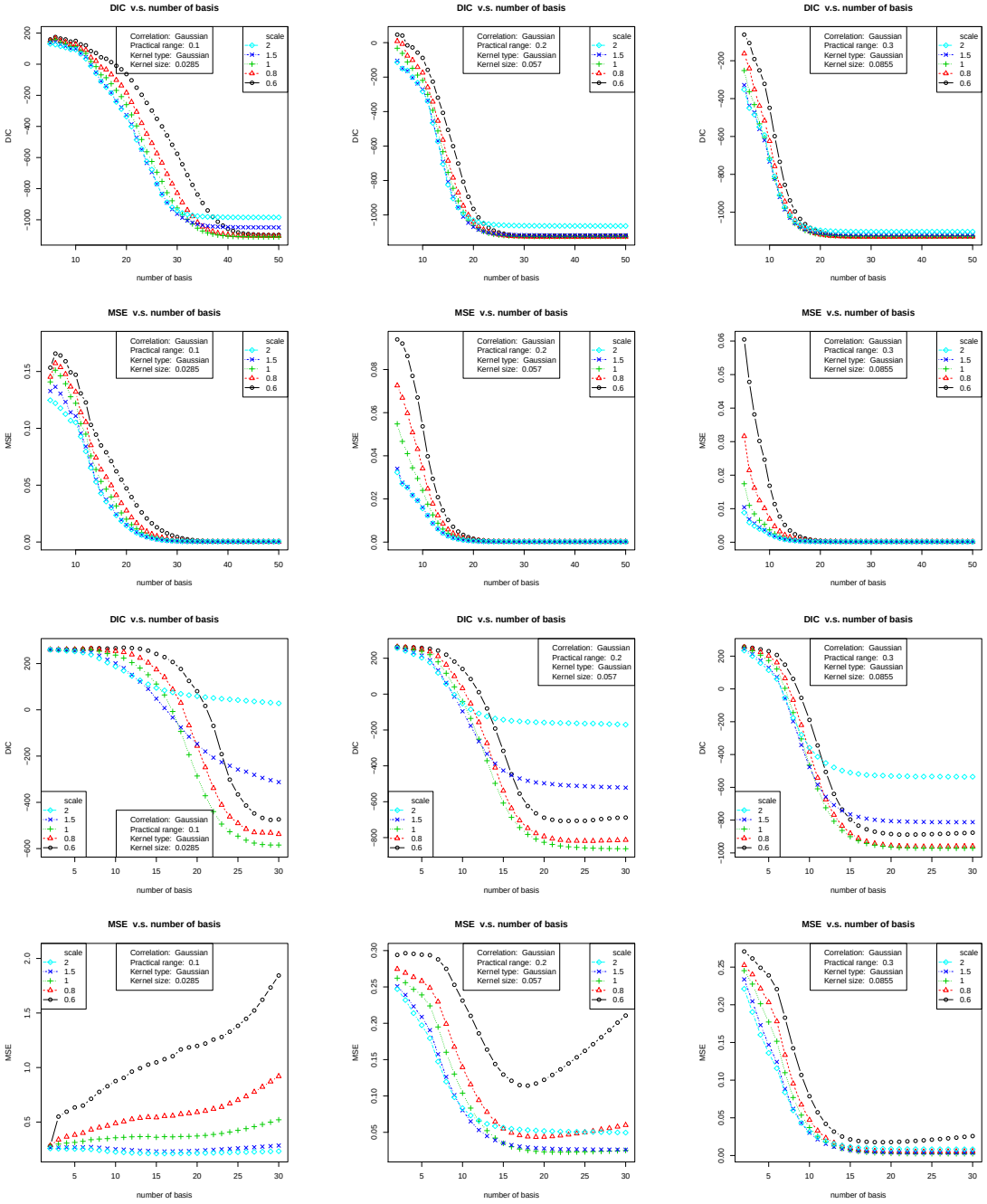


Figure 2.4: Process convolution GP model performance based on a Gaussian kernel for 1-d (*top two rows*) and 2-d (*bottom two rows*) data generated via a Gaussian correlation function

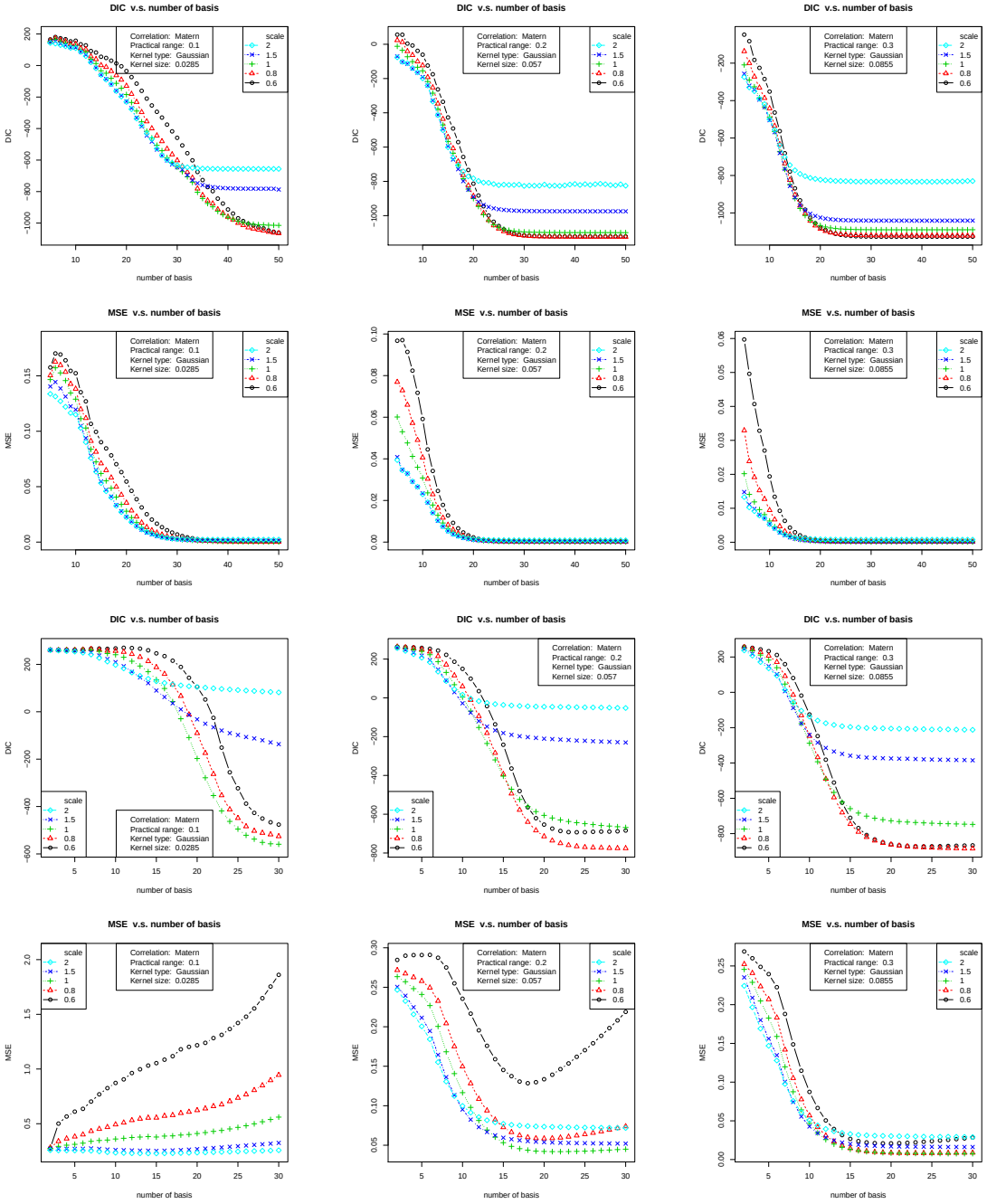


Figure 2.5: Process convolution GP model performance based on a Gaussian kernel for 1-d (*top two rows*) and 2-d (*bottom two rows*) data generated via a Matérn ($\kappa = 7$) correlation function

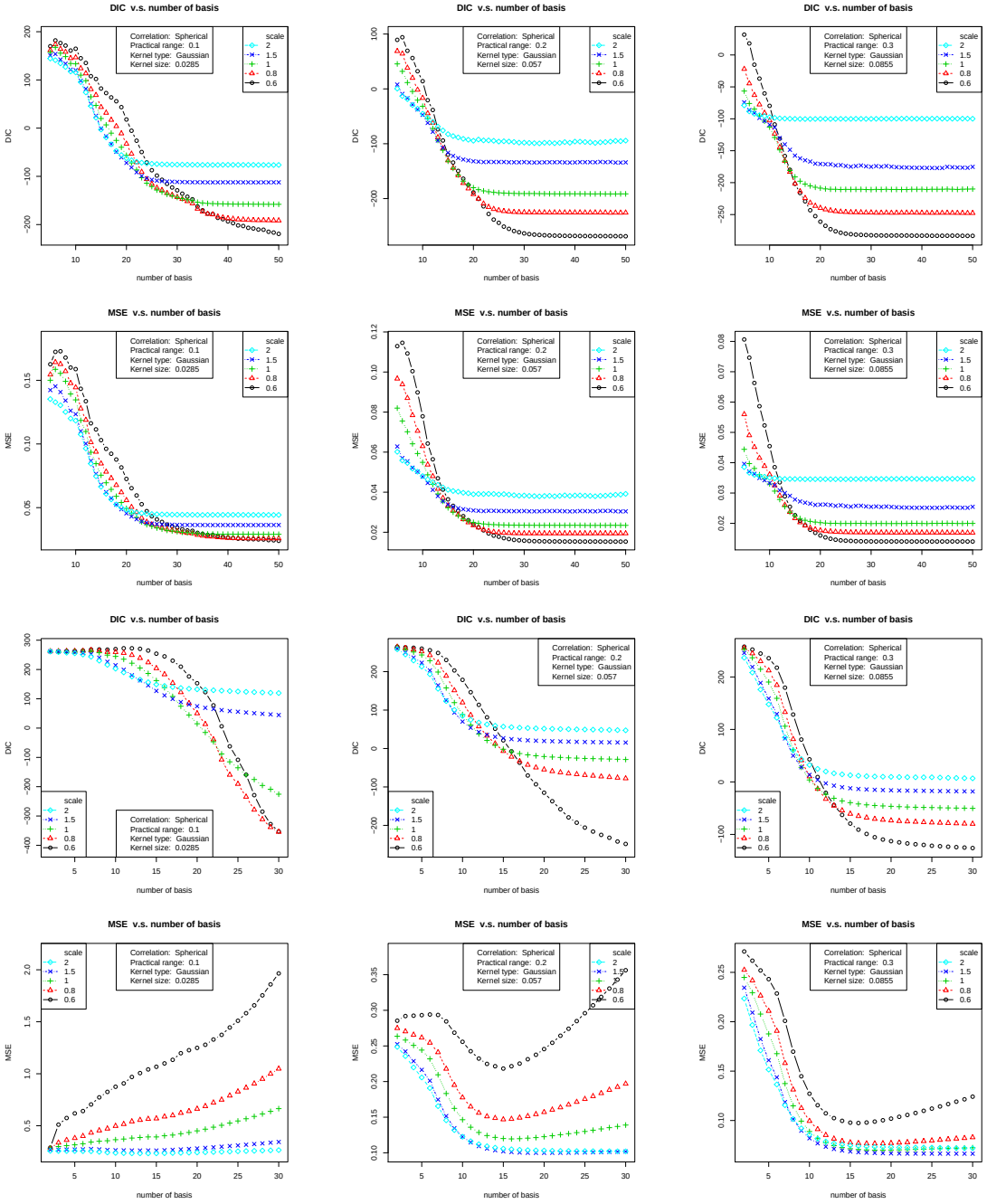


Figure 2.6: Process convolution GP model performance based on a Gaussian kernel for 1-d (*top two rows*) and 2-d (*bottom two rows*) data generated via a Spherical correlation function

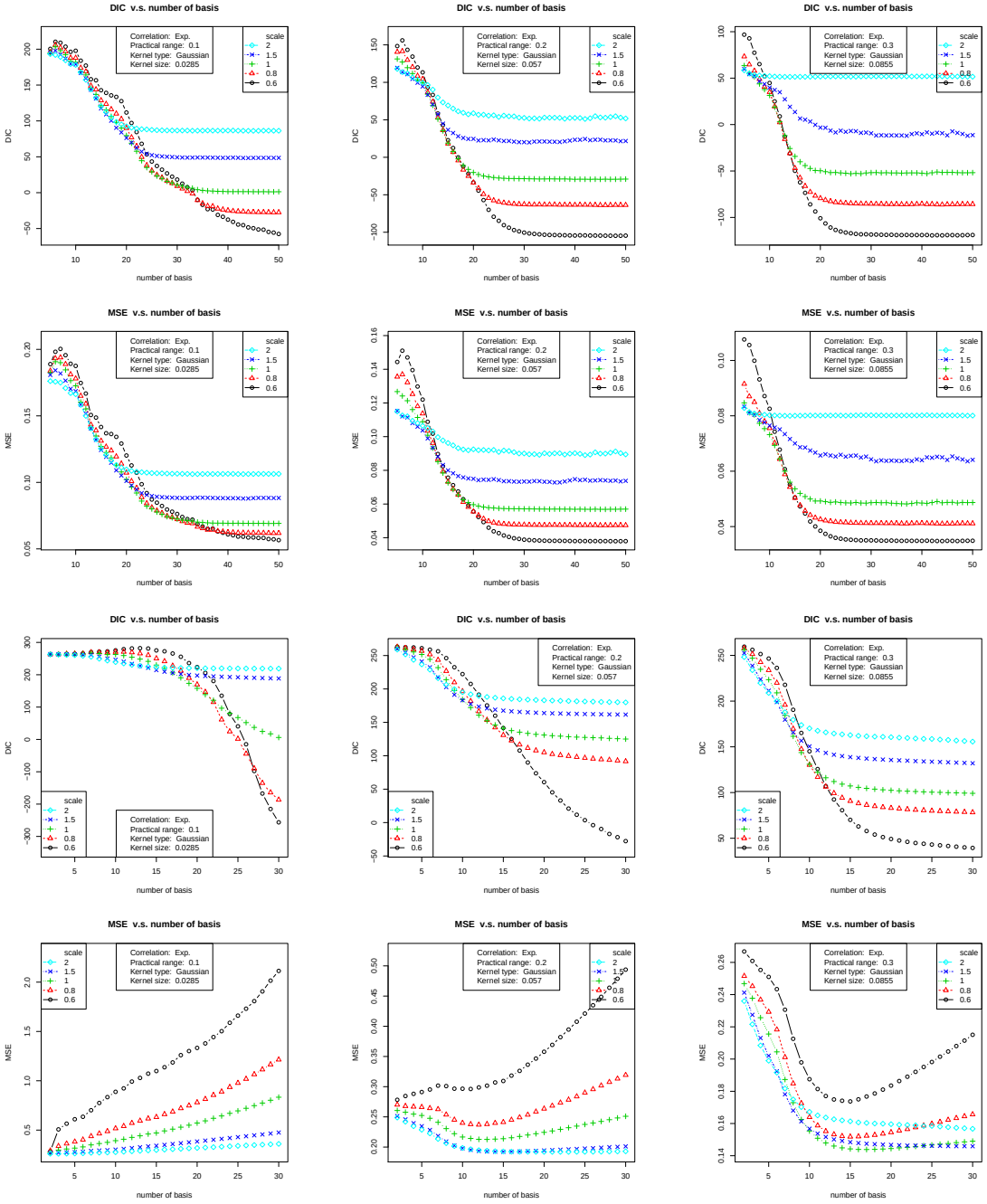


Figure 2.7: Process convolution GP model performance based on a Gaussian kernel for 1-d (*top two rows*) and 2-d (*bottom two rows*) data generated via an Exponential correlation function

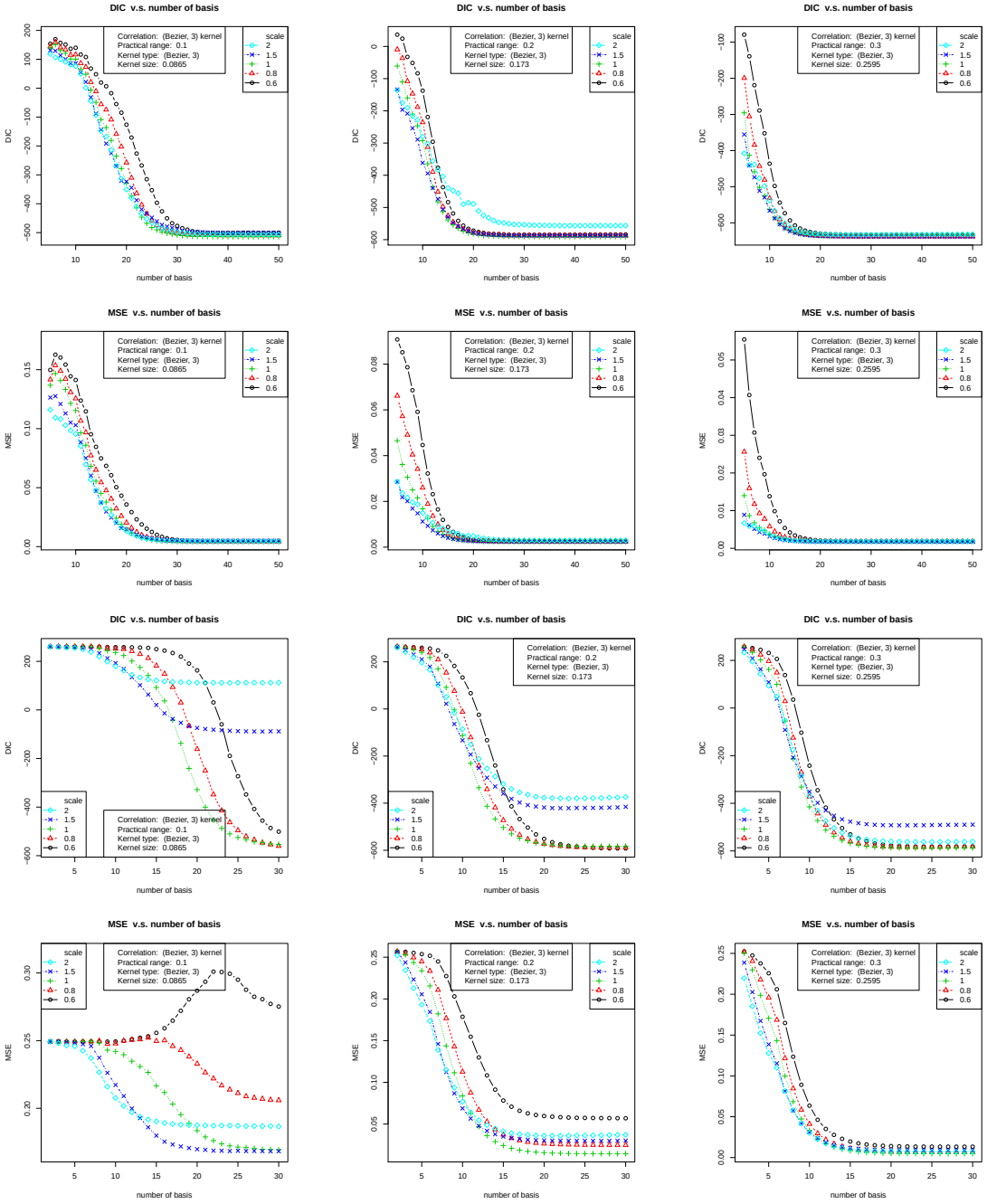


Figure 2.8: Process convolution GP model performance based on a Bézier ($\kappa = 3$) kernel for 1-d (*top two rows*) and 2-d (*bottom two rows*) data generated via a correlation function induced by the Bézier ($\kappa = 3$) kernel

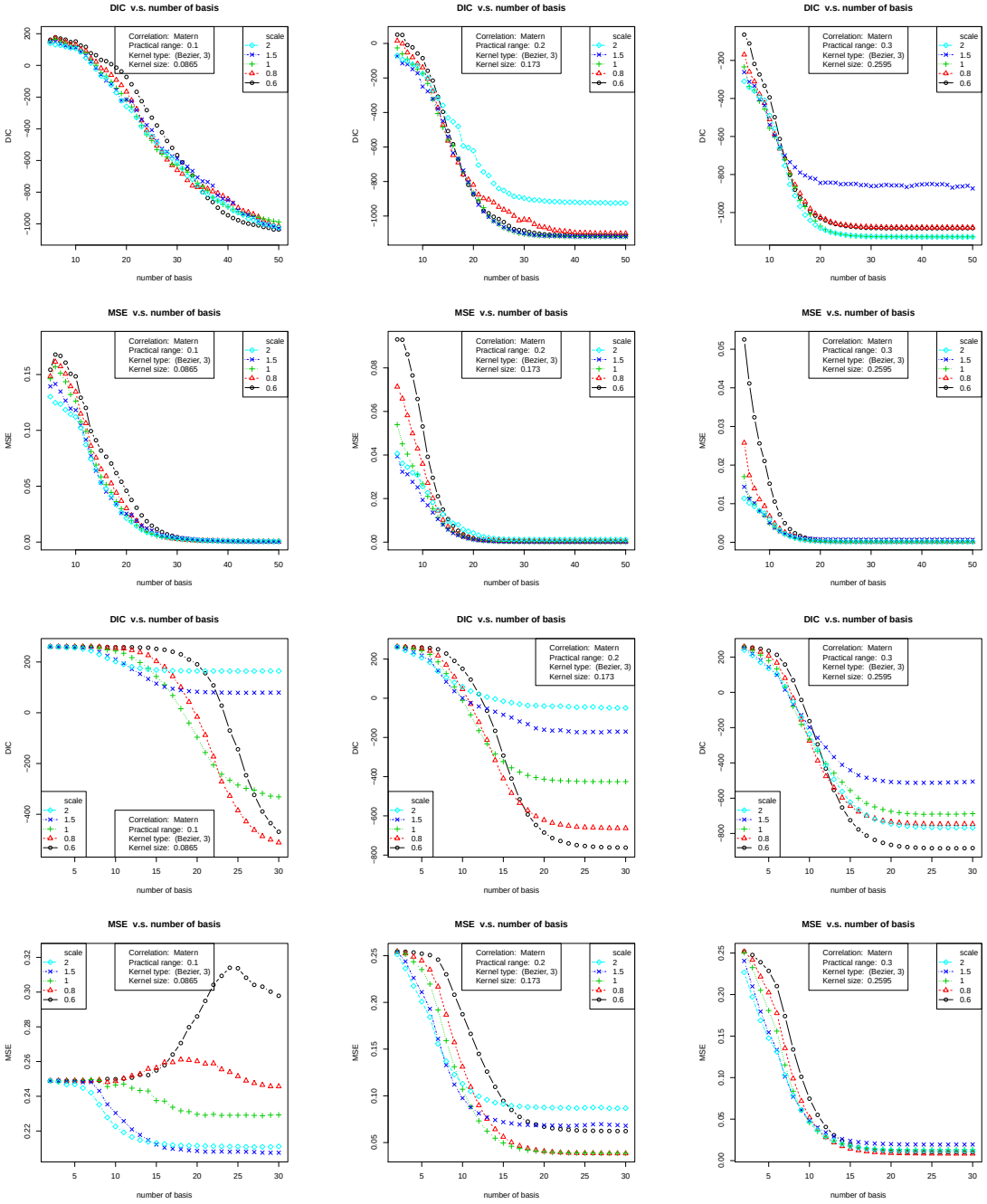


Figure 2.9: Process convolution GP model performance based on a Bézier ($\kappa = 3$) kernel for 1-d (*top two rows*) and 2-d (*bottom two rows*) data generated via a Matérn ($\kappa = 7$) correlation function

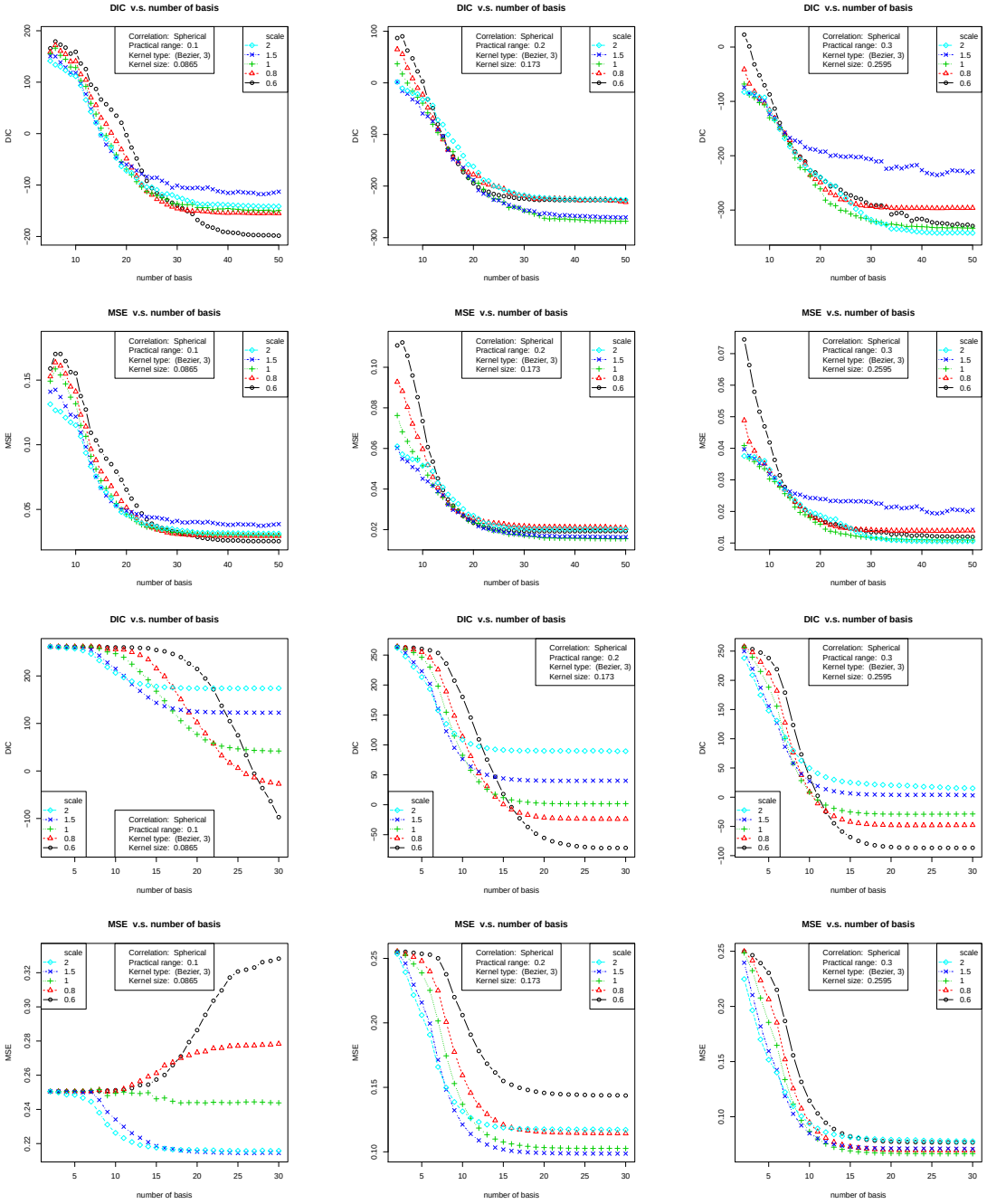


Figure 2.10: Process convolution GP model performance based on a Bézier ($\kappa = 3$) kernel for 1-d (*top two rows*) and 2-d (*bottom two rows*) data generated via a Spherical correlation function

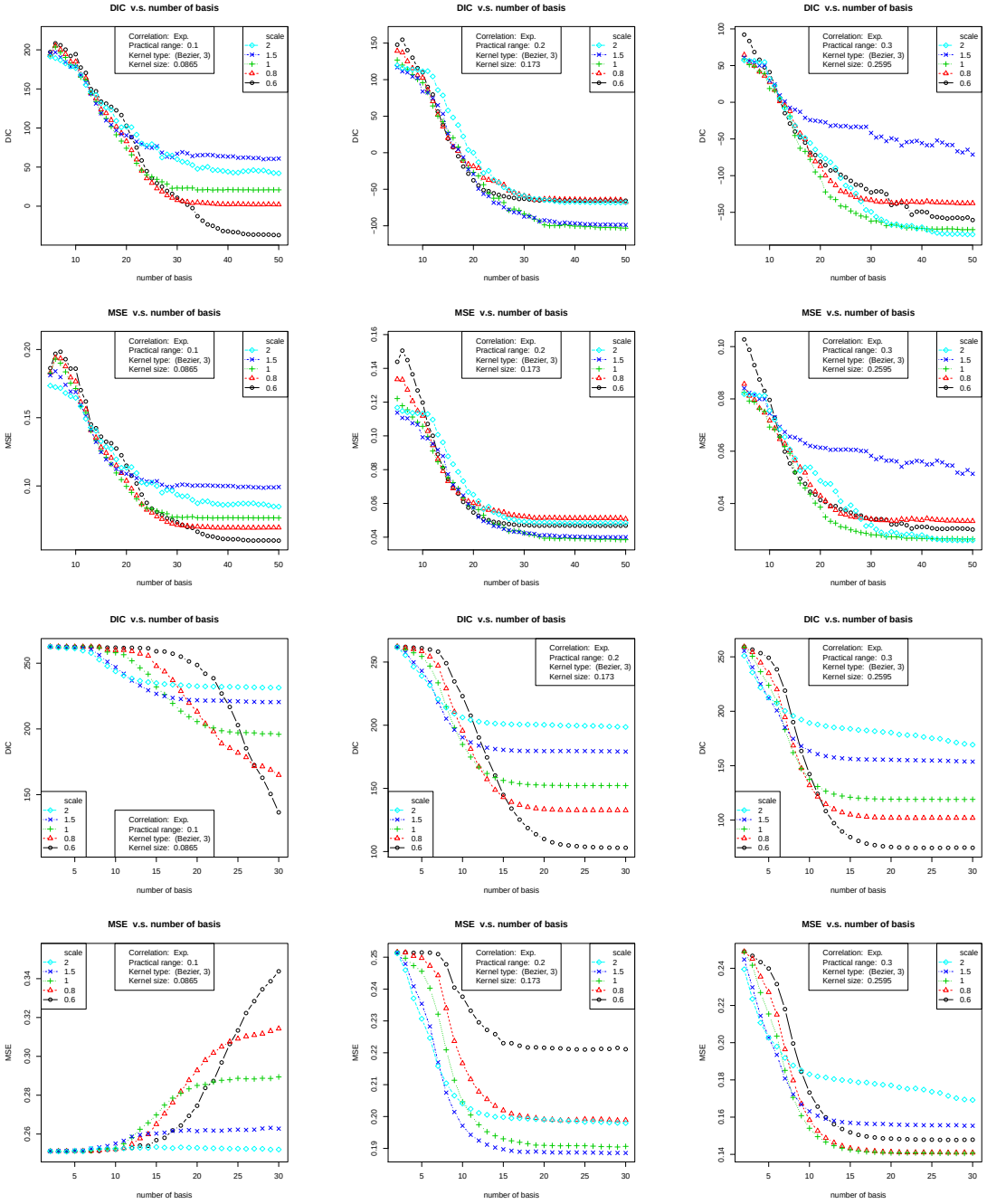


Figure 2.11: Process convolution GP model performance based on a Bézier ($\kappa = 3$) kernel for 1-d (*top two rows*) and 2-d (*bottom two rows*) data generated via an Exponential correlation function

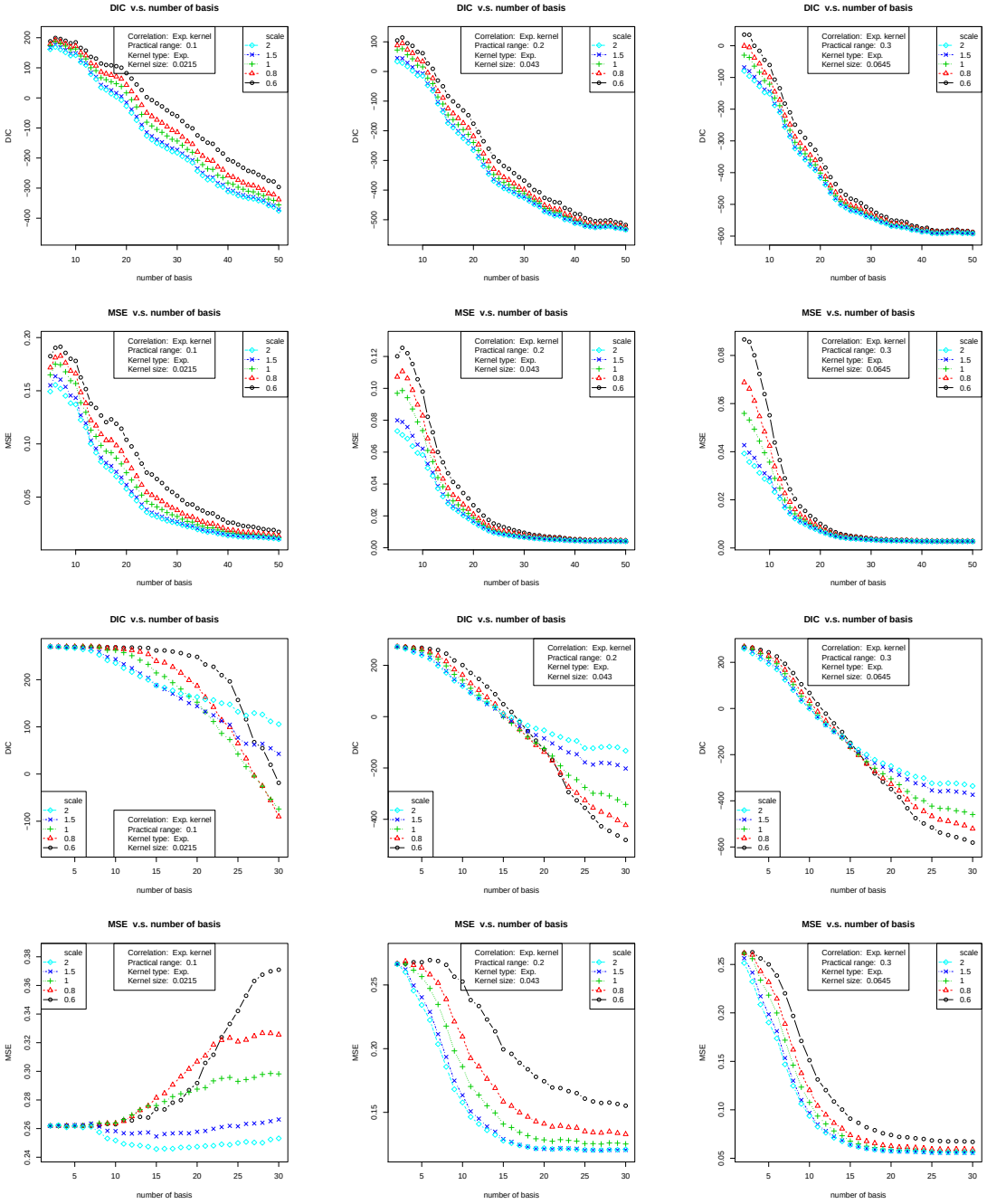


Figure 2.12: Process convolution GP model performance based on an Exponential kernel for 1-d (*top two rows*) and 2-d (*bottom two rows*) data generated via a correlation function induced by the Exponential kernel

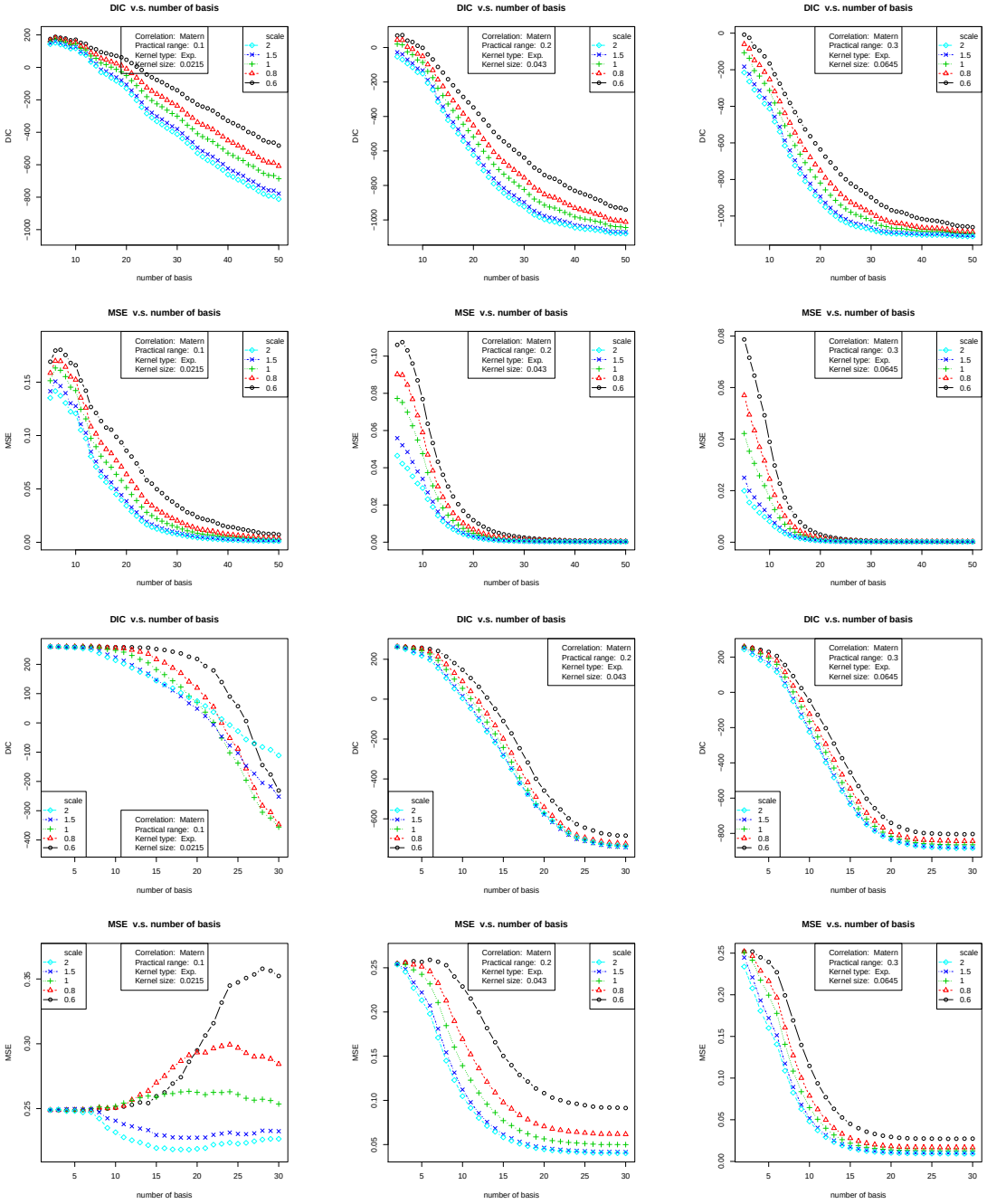


Figure 2.13: Process convolution GP model performance based on an Exponential kernel for 1-d (*top two rows*) and 2-d (*bottom two rows*) data generated via a Matérn ($\kappa = 7$) correlation function

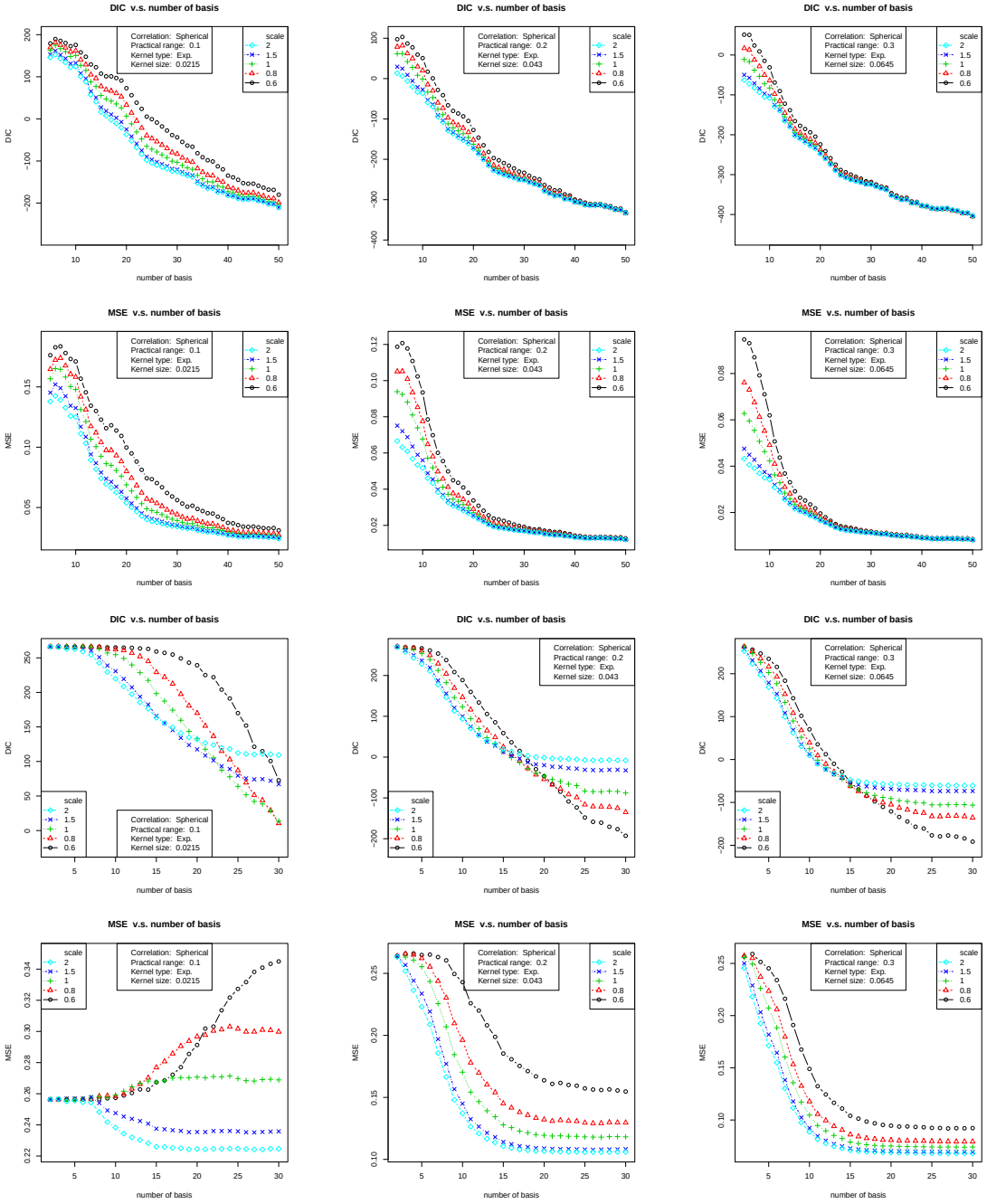


Figure 2.14: Process convolutions GP model performance based on an Exponential kernel for 1-d (*top two rows*) and 2-d (*bottom two rows*) data generated via an Spherical correlation function

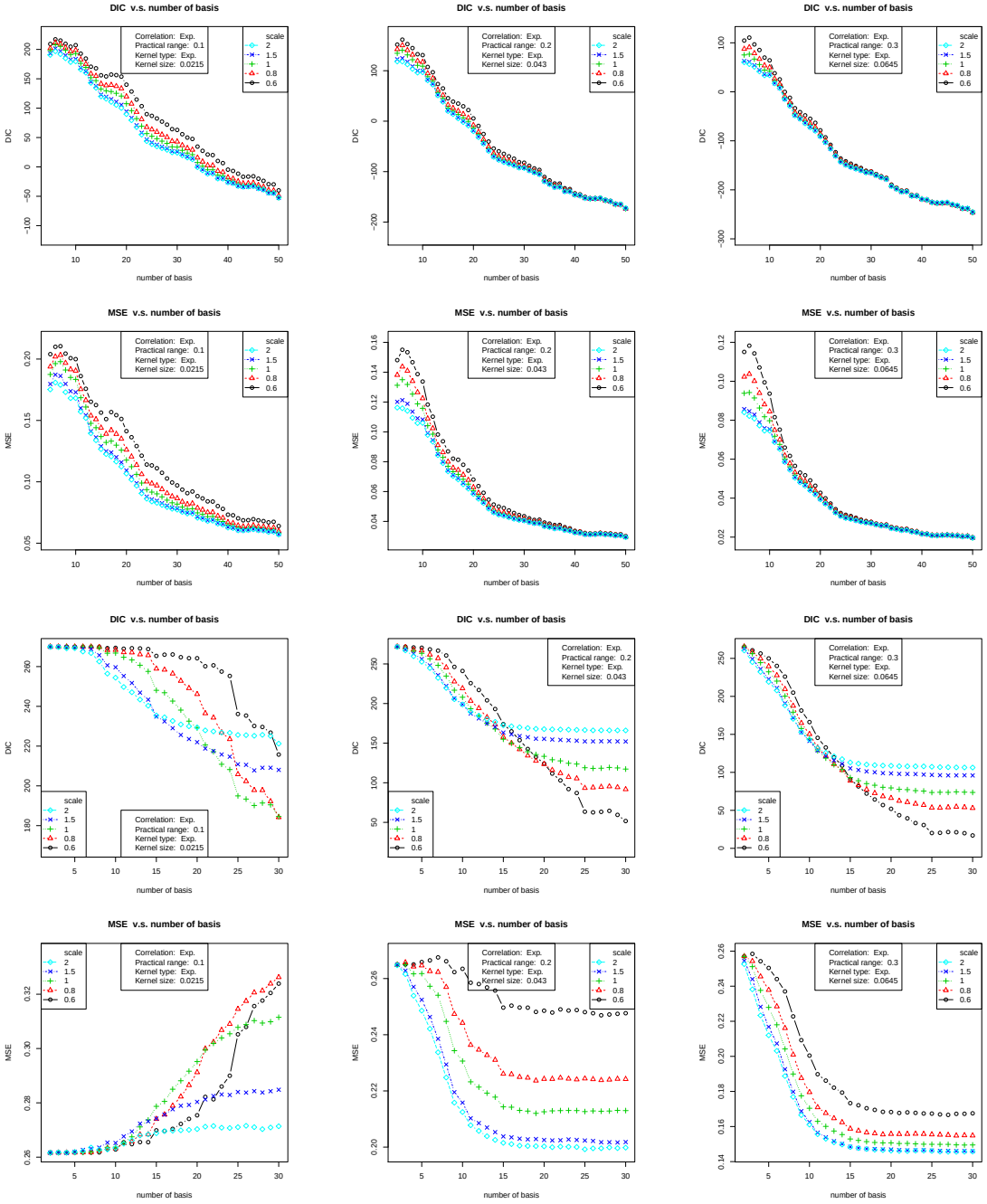


Figure 2.15: Process convolution GP model performance based on an Exponential kernel for 1-d (*top two rows*) and 2-d (*bottom two rows*) data generated via an Exponential correlation function

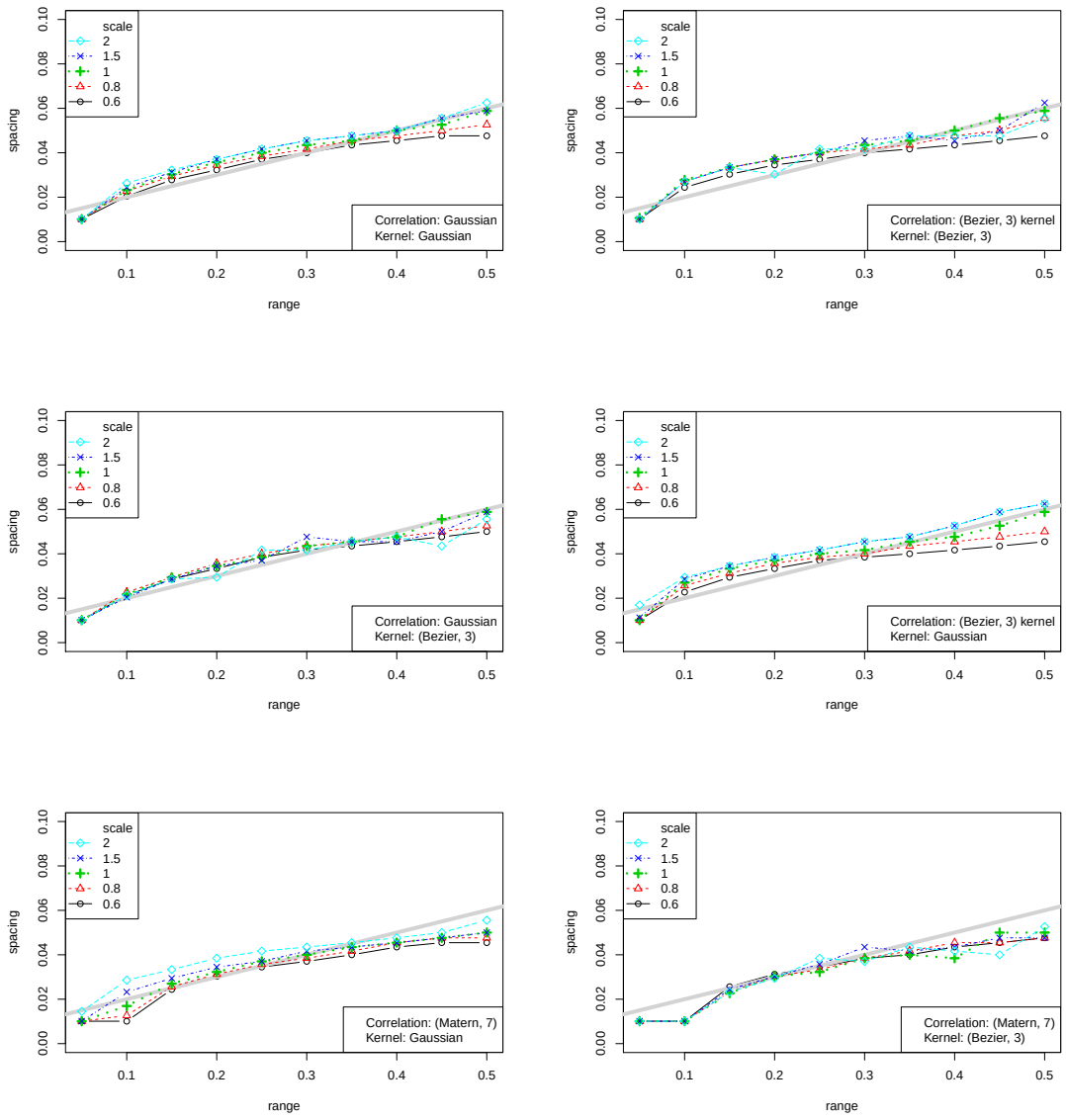


Figure 2.16: Basis spacing v.s. practical range of correlation function for 1-d data

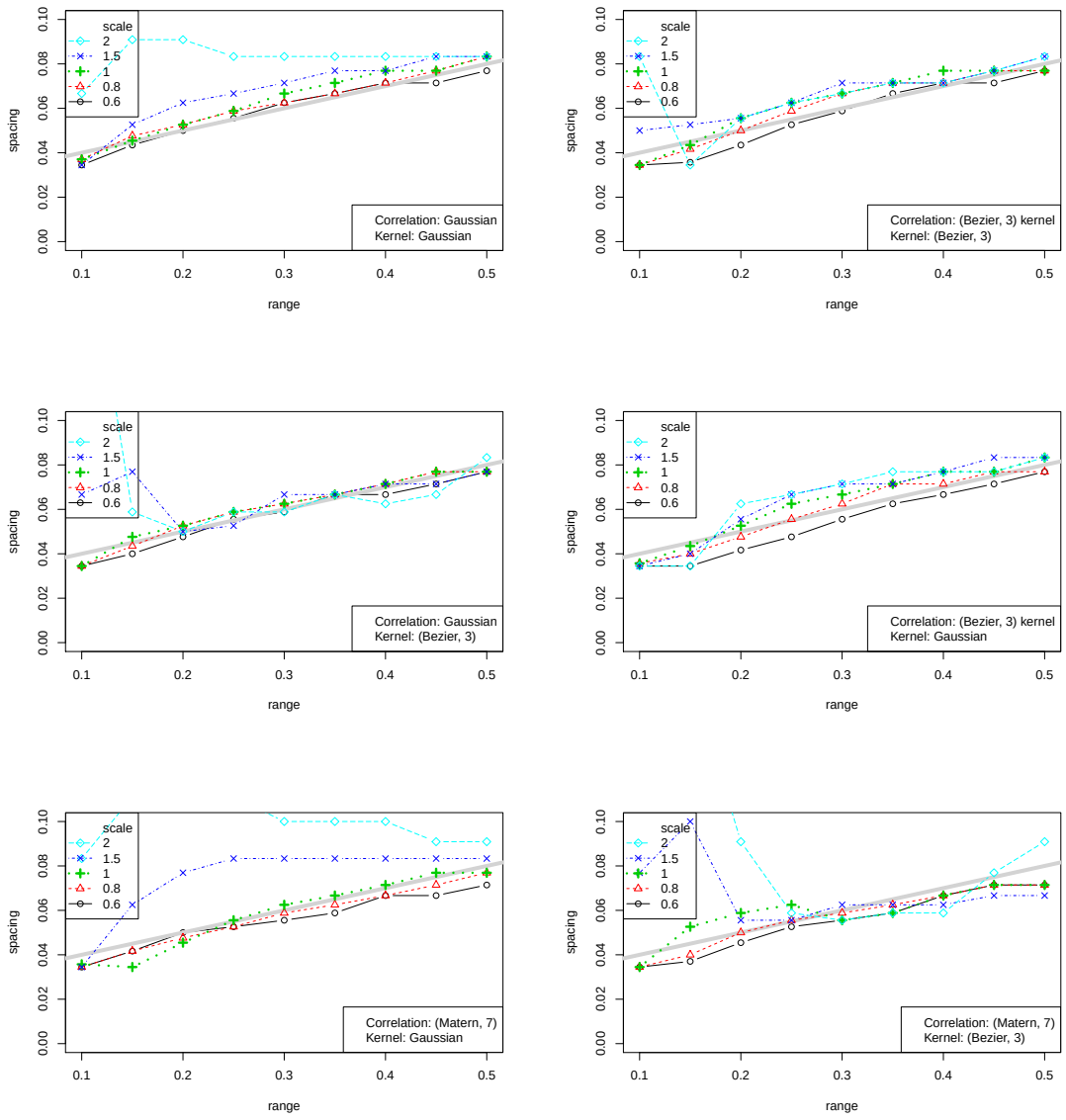


Figure 2.17: Basis spacing v.s. practical range of correlation function for 2-d data

# Chapter 3

# Treed Process Convolution GP model

## 3.1 Introduction

In spatial modeling, observations can be related to the underlying process $z$ using two different approaches, *interpolation* or *smoothing*. Interpolation in the spatial modeling setting is commonly known as *Kriging* (Matheron, 1963), where the value of $z$ at a location $\mathbf{s}$ is a weighted average of surrounding observations. Smoothing assumes each observation $y$ to be the additive combination of $z$ and random measurement error $\epsilon$. In traditional GP models, smoothing can be achieved by adding a so-called *nugget* term in the definition of the correlation function for $z$ (Gramacy, 2005), which is quite popular in the Geostatistical community. This is mathematically equivalent to adding a measurement error term $\epsilon$ to the stochastic component, as shown by Equation (1.20). However, the nugget term has less than satisfactory statistical interpretation and many authors have advised against its use for this reason. In the following sections, a de-

tail formulation of the treed process convolution GP model (TPCGP) is presented. It follows the smoothing approach because it is the common choice in the literature, and measurement error is often present in most spatial applications to which TPCGP can be applied. A key feature of TPCGP is nonstationarity induced by partitioning the spatial domain and having a separate latent process for each partition. The partitioning methodology follows a binary tree generating scheme from that of Bayesian CART model (Chipman et al., 1998). A Bayesian approach is used to explore the treed model space and estimate the parameters simultaneously.

## 3.2 Model setup

As mentioned before, the model structure of TPCGP follows the process convolution approach. Using the same notation as in Section 1.6.2, the observation at location $\mathbf{s}$ is denoted by $y(\mathbf{s})$, where $\mathbf{s} \in \mathcal{S} \subseteq \mathbb{R}^d$. To smooth the data, $y(\mathbf{s})$ is decomposed as

$$y(\mathbf{s}) = \mu(\mathbf{s}) + z(\mathbf{s}) + \epsilon(\mathbf{s}), \tag{3.1}$$

where $\mu(\mathbf{s})$ denotes the mean function, $z(\mathbf{s}) = \int_{\mathcal{S}} k(\mathbf{u}-\mathbf{s})x(\mathbf{u})d\mathbf{u}$ is a stochastic process generated via process convolutions, and $\epsilon(\mathbf{s}) \sim N(0, \phi^{-1})$ denotes the Gaussian measurement error at $\mathbf{s}$ with $\phi$ being the precision. In general, the mean function is specified as a linear function of covariates (spatial locations and/or attributes). Although the mean function can be specified in higher order, complicated trend characteristics are usually better described through the stochastic component $z$. The covariance between

$z(\mathbf{s}_i)$ and $z(\mathbf{s}_{i'})$ is given by

$$Cov(z(\mathbf{s}_i), z(\mathbf{s}_{i'})) = \int_{\mathcal{S}} \int_{\mathcal{S}} k(\mathbf{s}_i - \mathbf{u}_j) k(\mathbf{s}_{i'} - \mathbf{u}_{j'}) Cov(x(\mathbf{u}_j), x(\mathbf{u}_{j'})) d\mathbf{u}_j d\mathbf{u}_{j'}. \tag{3.2}$$

Suppose that $\mathcal{S}$ is partitioned into $b$ disjoint regions $\{\mathcal{S}_\nu : \nu = 1, \cdots, b\}$ as described in Section 1.7, and assuming that each partition has a separate latent process. Then, $z$ is clearly nonstationary except when all latent processes are White noise with the same marginal variance. Otherwise, convolving the kernel with a set of latent processes with different variability creates a nonstationary GP model whose covariance (or correlation) structure is governed by that of the latent processes and the kernel.

In practice, discrete process convolutions (DPC) is used for modeling since computers can only store a finite set of numbers. As shown in Section 1.6.2, a finite set of regularly spaced basis points $\{\mathbf{u}_1, \cdots, \mathbf{u}_m\}$ is fixed in $\mathcal{S}$ and $z$ can be approximated by $z(\mathbf{s}) = \sum_{j=1}^{m} k(\mathbf{u}_j - \mathbf{s}) x(\mathbf{u}_j)$. As mentioned before, DPC is nonstationary by construction. Partitioning generalizes this model by allowing more flexiblity to the background points $\{x(\mathbf{u}_1), \cdots, x(\mathbf{u}_m)\}$. Given a finite set of samples $\{y(\mathbf{s}_1), \cdots, y(\mathbf{s}_n)\}$, the model can be written in matrix/vector notation:

$$\mathbf{y} = \boldsymbol{\mu} + \mathbf{z} + \boldsymbol{\epsilon}, \tag{3.3}$$

where $\mathbf{y} = (y(\mathbf{s}_1), \cdots, y(\mathbf{s}_n))^\top$; $\boldsymbol{\mu} = \mathbf{F}\boldsymbol{\beta}$ such that $\mathbf{F}$ denotes a $(n \times (p+1))$ design matrix, $\boldsymbol{\beta}$ denotes a $((p+1) \times 1)$ coefficient vector, $p$ denotes the number of covariates, and the additional dimension corresponds to the intercept; the vector $\mathbf{z}$ and $\boldsymbol{\epsilon}$ contain the values of the process $z$ and $\epsilon$, respectively, i.e., $\mathbf{z} = (z(\mathbf{s}_1), \cdots, z(\mathbf{s}_n))^\top$, and $\boldsymbol{\epsilon} =$

$(\epsilon(\mathbf{s}_1), \cdots, \epsilon(\mathbf{s}_n))^\top$. As shown in Section 1.6, $\mathbf{z}$ can be written in matrix form as

$$\mathbf{z} = \mathbf{K}\mathbf{x}, \tag{3.4}$$

where $\mathbf{x} = (x(\mathbf{u}_1), \cdots, x(\mathbf{u}_m))^\top$ and $\mathbf{K}$ is a $(n \times m)$ kernel matrix with elements $\mathbf{K}_{ij} = k(\mathbf{u}_j - \mathbf{s}_i)$. Suppose that $\mathcal{S}$ is partitioned into $b$ disjoint regions $\{\mathcal{S}_1, \cdots, \mathcal{S}_b\}$, a separate mean function $\boldsymbol{\mu}_\nu$, latent process component $\mathbf{x}_\nu$, and observation error precision $\phi_\nu$ are assumed for each partition. This leads to the partitioning of $\{\mathbf{y}, \boldsymbol{\mu}, \mathbf{F}, \mathbf{z}, \mathbf{K}, \mathbf{x}, \boldsymbol{\epsilon}\}$ such that

$$\mathbf{y} = (\mathbf{y}_1^\top, \cdots, \mathbf{y}_b^\top)^\top, \quad \boldsymbol{\epsilon} = (\boldsymbol{\epsilon}_1^\top, \cdots, \boldsymbol{\epsilon}_b^\top)^\top, \quad \boldsymbol{\epsilon}_\nu \sim N(0, \phi_\nu^{-1}\mathbf{I}_{n_\nu}),$$

$$\boldsymbol{\mu} = (\boldsymbol{\mu}_1^\top, \cdots, \boldsymbol{\mu}_b^\top)^\top, \quad \mathbf{F} = (\mathbf{F}_1^\top, \cdots, \mathbf{F}_b^\top)^\top, \quad \boldsymbol{\mu}_\nu = \mathbf{F}_\nu\boldsymbol{\beta}_\nu$$

$$\mathbf{z} = (\mathbf{z}_1^\top, \cdots, \mathbf{z}_b^\top)^\top, \quad \mathbf{K} = (\mathbf{K}_1^\top, \cdots, \mathbf{K}_b^\top)^\top, \quad \mathbf{x} = (\mathbf{x}_1^\top, \cdots, \mathbf{x}_b^\top)^\top, \quad \mathbf{z}_\nu = \mathbf{K}_\nu\mathbf{x},$$

where $n_\nu$ denotes the number of observations in $\mathcal{S}_\nu$, $\mathbf{I}_{n_\nu}$denotes the $(n_\nu \times n_\nu)$ identity matrix, and $\{\mathbf{y}_\nu, \boldsymbol{\mu}_\nu, \mathbf{F}_\nu, \boldsymbol{\beta}_\nu, \mathbf{z}_\nu, \mathbf{K}_\nu, \mathbf{x}_\nu, \boldsymbol{\epsilon}_\nu, \phi_\nu\}$ are the corresponding matrix/vector components associated with $\mathcal{S}_\nu$. Partitions are generated by recursively choosing a dimension within the parent partition and split at one of the available values. Spliting is allowed only within the parent partition, i.e., a split can not go across the boundary of the parent partition. This results in a binary tree structure as shown in Section 1.7 where the internal nodes represent the parent partitions generated during the splitting process and the terminal nodes represent the final partitions. Together, a chosen dimension and the associated splitting value forms a splitting rule, and each internal node of the treed model is assosicated with a unique splitting rule. For the rest of this dissertation, the set of all splitting rules in TPCGP is denoted by $\boldsymbol{\rho}$ and the treed model

itself by $\mathcal{T}$. Under this construction, the sampling distribution of $\mathbf{y}_\nu$ is given by

$$\mathbf{y}_\nu|\mathbf{x}, \boldsymbol{\beta}_\nu, \phi_\nu, \boldsymbol{\rho}, \mathcal{T}, \mathbf{K}_\nu, \mathbf{F}_\nu, \quad \sim \quad N_{n_\nu}(\mathbf{F}_\nu\boldsymbol{\beta}_\nu + \mathbf{K}_\nu\mathbf{x}\,, \phi_\nu^{-1}\mathbf{I}_{n_\nu}). \tag{3.5}$$

Following the convention used in Classification and Regression Trees, terminal node parameters are assumed to be independent conditional on the tree structure. The full likelihood of the treed model $\mathcal{T}$ and its parameters is given by

$$\begin{aligned} & L(\mathbf{x}, \{\boldsymbol{\beta}_\nu, \phi_\nu\}_{\nu=1}^b, \boldsymbol{\rho}, \mathcal{T}|\mathbf{y}, \mathbf{K}, \mathbf{F}) \\ = & \prod_{\nu=1}^{b} N_{n_\nu}(\mathbf{F}_\nu\boldsymbol{\beta}_\nu + \mathbf{K}_\nu\mathbf{x}\,, \phi_\nu^{-1}\mathbf{I}_{n_\nu}) \\ = & \prod_{\nu=1}^{b} \Big(\frac{\phi_\nu}{2\pi}\Big)^{n_\nu/2} \exp\Big\{ -\frac{\phi_\nu}{2}(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})^\top(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})\Big\}. \end{aligned} \tag{3.6}$$

## 3.3 Bayesian Estimation

Given a treed model $\mathcal{T}$ with $b$ partitions, the set of unknown parameters is denoted by $\boldsymbol{\Theta} = \{\{\mathbf{x}_\nu, \boldsymbol{\beta}_\nu, \phi_\nu\}_{\nu=1}^b, \boldsymbol{\rho}\}$. Using a Bayesian approach to explore the posterior space of treed models and simultaneously make inference about the model paramters requires prior specification for $(\boldsymbol{\Theta}, \mathcal{T})$. Following Chipman et al. (1998), the conditional relationship $P(\boldsymbol{\Theta}, \mathcal{T}) = P(\boldsymbol{\Theta}|\mathcal{T})P(\mathcal{T})$ is used to specify the priors separately. First, $P(\mathcal{T})$ is specified through a tree generating process as follows.

1. Initialize $\mathcal{T}$ consisting of a single root node denoted $\eta$.

2. Split the terminal node $\eta$ with probability $P_{split}(\eta, \mathcal{T})$.

3. Assign a splitting rule $\rho$ with probability $P_{rule}(\rho|\eta, \mathcal{T})$ to $\eta$ if it splits, and create

its left and right children nodes (new terminal nodes). Let $\mathcal{T}$ denote the new tree, then repeat steps 2 and 3.

The probability of splitting a node $\eta$ is determined by $P_{split}(\eta, \mathcal{T}) = \alpha_T(1 + d_\eta)^{-\beta_T}$, where $d_\eta$ is the number of splits above $\eta$, $0 \leq \alpha_T \leq 1$ controls the shape (balance) of the tree, and $\beta_T \geq 0$ controls the size of the tree. In general, decreasing $\beta_T$ increases the probability of having a tree with more terminal nodes. The treed model prior, $P(\mathcal{T})$, is determined by

$$P(\mathcal{T}) = \prod_{\eta \in \mathcal{I}} P_{split}(\eta, \mathcal{T}) \prod_{\eta \in \mathcal{L}} [1 - P_{split}(\eta, \mathcal{T})], \tag{3.7}$$

where $\mathcal{I}$ and $\mathcal{L}$ denote the set of internal nodes and terminal nodes in $\mathcal{T}$, respectively. The splitting rule probability $P_{rule}(\rho|\eta, \mathcal{T})$ is usually taken to be a uniform distribution on the set of available splitting dimensions and values. The prior for the set of all splitting rules $P(\boldsymbol{\rho}|\mathcal{T})$, is given by

$$P(\boldsymbol{\rho}|\mathcal{T}) = \prod_{\eta \in \mathcal{I}} P_{rule}(\rho|\eta, \mathcal{T}). \tag{3.8}$$

To specify $P(\boldsymbol{\Theta}|\mathcal{T})$, suggestions from Chipman et al. (1998) are followed by imposing conjugate priors on the terminal node parameters and assuming conditional independence of parameters across terminal nodes. Doing this allows us to analytically marginalize the terminal node parameters out from the joint posterior. The conjugate priors are specified as

$$\boldsymbol{\beta}_\nu|\phi_\nu, \boldsymbol{\beta}_0, \mathbf{C}, \boldsymbol{\rho}, \mathcal{T} \sim N_{p+1}(\boldsymbol{\beta}_0\,, (\phi_\nu \mathbf{C})^{-1}), \quad \phi_\nu|b_y, \boldsymbol{\rho}, \mathcal{T} \sim G(a_y\,, b_y),$$

$$\mathbf{x}_\nu|\lambda_\nu, \boldsymbol{\rho}, \mathcal{T} \sim N_{m_\nu}(\mathbf{0}\,, (\lambda_\nu \mathbf{I}_{m_\nu})^{-1}),$$

where $G$ denotes the Gamma distribution. Note that $\mathbf{x}_\nu$ is given a Gaussian distributed prior each having a separate precision $\lambda_\nu$. As a result, this puts a nonstationary Gaussian prior on $\mathbf{z}$. To extend the hierarchy of inference one step further, the following hyperpriors are imposed on the hyperparameters:

$$\lambda_\nu | b_x, \boldsymbol{\rho}, \mathcal{T} \sim G(a_x\,, b_x) \quad \boldsymbol{\beta}_0 \sim N(\boldsymbol{\mu}, \mathbf{B}^{-1}), \quad \mathbf{C} \sim W((\varphi \mathbf{V})^{-1}, \varphi),$$

$$b_y \sim G(\tau_y, \xi_y), \quad b_x \sim G(\tau_x, \xi_x),$$

where $\{a_x,\ a_y,\ \boldsymbol{\mu},\ \mathbf{B},\ \varphi,\ \mathbf{V},\ \tau_x,\ \xi_x,\ \tau_y,\ \xi_y\}$ are constants, and $W$ denotes the Wishart distribution. The kernel covariance matrix $\mathbf{Q}^{-1}$ is assumed to be fixed in this chapter, and this assumption will be relaxed in the next chapter. Together with $P(\mathcal{T})$, $P(\boldsymbol{\rho}|\mathcal{T})$, and the likelihood given by (3.6), the conditional posterior distribution for $(\mathcal{T}, \boldsymbol{\rho})$ can be obtained as

$$\begin{aligned} P(\mathcal{T}, \boldsymbol{\rho} | \cdots) \ \propto \ & \Bigg( \prod_{\nu=1}^{b} \Big(\frac{1}{2\pi}\Big)^{(n_\nu + m_\nu)/2} |(\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})|^{-1/2} \frac{b_y^{a_y}}{\Gamma(a_y)} \frac{b_x^{a_x}}{\Gamma(a_x)} \times \\ & \Gamma(n_\nu/2 + a_y) \Big( b_y + \frac{1}{2} \Big( s_\nu^2 + (\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}_\nu)^\top \mathbf{R}_\nu^{-1} (\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}_\nu) \Big) \Big)^{-(n_\nu/2 + a_y)} \times \\ & \Gamma(m_\nu/2 + a_x) \Big( \frac{1}{2} \mathbf{x}_\nu^\top \mathbf{x}_\nu + b_x \Big)^{-(m_\nu/2 + a_x)} \Bigg) P(\boldsymbol{\rho}|\mathcal{T}) P(\mathcal{T}), \end{aligned} \tag{3.9}$$

where $\mathbf{R}_\nu = \mathbf{C}^{-1} + (\mathbf{F}_\nu^\top \mathbf{F}_\nu)^{-1}$ and this posterior can not be obtained in closed form. The partially marginalized conditionals for the model parameters are obtained as

$$\mathbf{x}|\cdots \sim N_m\Big((\mathbf{K}^\top\mathbf{\Phi}\mathbf{K}+\mathbf{\Lambda})^{-1}\mathbf{K}^\top\mathbf{\Phi}\mathbf{w}\ ,\ (\mathbf{K}^\top\mathbf{\Phi}\mathbf{K}+\mathbf{\Lambda})^{-1}\Big), \tag{3.10}$$

$$\lambda_\nu|\cdots \sim G\Big(\frac{m_\nu}{2}+a_x\ ,\ \frac{1}{2}\mathbf{x}_\nu^\top\mathbf{x}_\nu+b_x\Big), \tag{3.11}$$

$$\boldsymbol{\beta}_\nu|\cdots \sim N_{p+1}\Big((\mathbf{F}_\nu^\top\mathbf{F}_\nu+\mathbf{C})^{-1}(\mathbf{F}_\nu^\top\mathbf{v}_\nu+\mathbf{C}\boldsymbol{\beta}_0)\ ,\ (\phi_\nu(\mathbf{F}_\nu^\top\mathbf{F}_\nu+\mathbf{C}))^{-1}\Big), \tag{3.12}$$

$$\phi_\nu|\cdots \sim G\Big(\frac{n_\nu}{2}+a_y\ ,\ b_y+\frac{1}{2}(s_\nu^2+(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}})^\top\mathbf{R}_\nu^{-1}(\boldsymbol{\beta}_0-\hat{\boldsymbol{\beta}}))\Big), \tag{3.13}$$

$$\boldsymbol{\beta_0}|\cdots \sim N_{p+1}\Big(\mathbf{G}\Big(\mathbf{C}\sum_{\nu=1}^{b}\phi_\nu\boldsymbol{\beta}_\nu+\mathbf{B}\boldsymbol{\mu}\Big)\ ,\ \mathbf{G}\Big), \tag{3.14}$$

$$\mathbf{C}|\cdots \sim W\Big(\Big(\sum_{\nu=1}^{b}\phi_\nu(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)^\top+\varphi\mathbf{V}\Big)^{-1}\ ,\ \varphi+b\Big), \tag{3.15}$$

$$b_y|\cdots \sim G\Big(a_yb+\tau_y\ ,\ \sum_{\nu=1}^{b}\phi_\nu+\xi_y\Big), \tag{3.16}$$

$$b_x|\cdots \sim G\Big(a_xb+\tau_x\ ,\ \sum_{\nu=1}^{b}\lambda_\nu+\xi_x\Big), \tag{3.17}$$

where

$$\mathbf{\Phi}=\begin{pmatrix}\mathbf{\Phi}_1 & \mathbf{0} & \dots & \mathbf{0}\\ \mathbf{0} & \mathbf{\Phi}_2 & \dots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{\Phi}_b\end{pmatrix},\quad \mathbf{\Lambda}=\begin{pmatrix}\mathbf{\Lambda}_1 & \mathbf{0} & \dots & \mathbf{0}\\ \mathbf{0} & \mathbf{\Lambda}_2 & \dots & \mathbf{0}\\ \vdots & \vdots & \ddots & \vdots\\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{\Lambda}_b\end{pmatrix},$$

$$\mathbf{\Phi}_\nu=\phi_\nu\mathbf{I}_{n_\nu},\quad \mathbf{\Lambda}_\nu=\lambda_\nu\mathbf{I}_{m_\nu},\quad \mathbf{w}=\mathbf{y}-\mathbf{F}\boldsymbol{\beta},\quad \hat{\boldsymbol{\beta}}=(\mathbf{F}^\top\mathbf{F})^{-1}\mathbf{F}^\top\mathbf{v},$$

$$\mathbf{v}=\mathbf{y}-\mathbf{K}\mathbf{x},\quad s_\nu^2=(\mathbf{v}_\nu-\mathbf{F}_\nu\hat{\boldsymbol{\beta}})^\top(\mathbf{v}_\nu-\mathbf{F}_\nu\hat{\boldsymbol{\beta}}),\quad \mathbf{G}=\Big(\mathbf{C}\sum_{\nu=1}^{b}\phi_\nu+\mathbf{B}\Big)^{-1}.$$

Reversible jump Markov Chain Monte Carlo (RJ-MCMC (Green, 1995)) is used to explore the posterior distributions of the treed model and unknown parameters. Specifically, the Metropolis-Hastings algorithm is used to explore the posterior space of $(\mathcal{T},\boldsymbol{\rho})$

while the Gibbs sampler is used to draw samples from the posterior distributions of the other parameters. The sampling procedure is summarized as follows.

1. Initialize counter $t = 0$ and parameters. Denote the single root node tree by $\mathcal{T}^0$.

2. Propose a new treed model $(\mathcal{T}^*, \boldsymbol{\rho}^*)$ from $(\mathcal{T}^t, \boldsymbol{\rho}^t)$ with distribution $q(\mathcal{T}^*, \boldsymbol{\rho}^* | \mathcal{T}^t, \boldsymbol{\rho}^t)$.

3. Compute

$$\alpha = \frac{P(\mathcal{T}^*, \boldsymbol{\rho}^* | \cdots) q(\mathcal{T}^t, \boldsymbol{\rho}^t | \mathcal{T}^*, \boldsymbol{\rho}^*)}{P(\mathcal{T}^t, \boldsymbol{\rho}^t | \cdots) q(\mathcal{T}^*, \boldsymbol{\rho}^* | \mathcal{T}^t, \boldsymbol{\rho}^t)},$$

   and set $(\mathcal{T}^{t+1}, \boldsymbol{\rho}^{t+1})$=$(\mathcal{T}^*, \boldsymbol{\rho}^*)$ with probability $min(\alpha, 1)$, otherwise keep the previous sample.

4. For $\nu = 1, \cdots, b$, sample $\boldsymbol{\beta}_\nu^{t+1}$, $\phi_\nu^{t+1}$ and $\lambda_\nu^{t+1}$ from (3.12), (3.13), and (3.11), respectively.

5. Sample $\mathbf{x}^{t+1}$, $\boldsymbol{\beta}_0^{t+1}$, $\mathbf{C}^{t+1}$, $b_y^{t+1}$, and $b_x^{t+1}$ from (3.10), (3.14), (3.15), (3.16), and (3.17), respectively.

6. Increment $t$ and repeat steps 2 through 5.

The distribution, $q(\mathcal{T}^*, \boldsymbol{\rho}^* | \mathcal{T}^t, \boldsymbol{\rho}^t)$, is a proposal of going from treed model $\mathcal{T}^t$ with spliting rules $\boldsymbol{\rho}^t$ to a new treed model $\mathcal{T}^*$ with splitting rules $\boldsymbol{\rho}^*$. Details of this proposal are given in the following section.

## 3.4 Treed Model Proposals

The treed model proposal is implemented via a set of tree modification operations called *Grow*, *Prune*, *Change* and *Swap* as discussed by Chipman et al. (1998). Each time when a new tree is proposed, a single operation is randomly selected with equal probability from these four operations. Consider the case where the *Grow* operation is chosen for the current treed model $\mathcal{T}^t$. A leaf node $\eta$ at depth $d_\eta$ is uniformly selected among the available candidates. Suppose that the selected leaf node corresponds to partition $\mathcal{S}_\nu$. Within this leaf node, a spatial dimension and a splitting value is uniformly selected from the available candidates. Availability is ensured when the split does not lead to empty terminal nodes. The selected leaf splits as a new parent node and is assigned a new splitting rule $\rho^+$. The data within this new parent node (a leaf node before splitting) is divided among its newly created children nodes according to the splitting rule. Denote this proposed treed model by $\mathcal{T}^*$. Notice that going from $\mathcal{T}^t$ to $\mathcal{T}^*$, there is an additional parameter, $\rho^+$. This change in the dimension of the parameter space is the main reason of using RJ-MCMC. A good primer on the application of RJ-MCMC is given by Gelman et al. (2004). In general, RJ-MCMC requires incorporating the probability of generating $\rho^+$ in $q(\mathcal{T}^*, \boldsymbol{\rho}^* | \mathcal{T}^t, \boldsymbol{\rho}^t)$. Since $\rho^+$ is sampled from the prior (a uniform distribution on both the spliting dimension and values), the Jacobian term that usually exists in the acceptance ratio can be omitted. Let $\mathcal{G}$ denote the set of growable leaves of $\mathcal{T}^t$ and $\mathcal{P}$ denote the set of pruneable nodes of $\mathcal{T}^*$. Going

from $\mathcal{T}^t$ to $\mathcal{T}^*$, the proposal probability is given by

$$q(\mathcal{T}^*, \boldsymbol{\rho}^* | \mathcal{T}^t, \boldsymbol{\rho}^t) = \frac{P_{rule}(\rho^+ | \eta, \mathcal{T}^t)}{|\mathcal{G}|}, \quad \text{where} \quad \boldsymbol{\rho}^* = (\boldsymbol{\rho}^t, \rho^+). \tag{3.18}$$

Going back from $\mathcal{T}^*$ to $\mathcal{T}^t$, the proposal is simply

$$q(\mathcal{T}^t, \boldsymbol{\rho}^t | \mathcal{T}^*, \boldsymbol{\rho}^*) = \frac{1}{|\mathcal{P}|}. \tag{3.19}$$

When the *Prune* operation is chosen, a parent node whose children are terminal nodes is selected uniformly from the available candidates. The children nodes are collapsed and their data are absorbed by the selected parent, which returns to be a terminal node. Since the *Prune* and *Grow* operations are counterparts of one another, the proposal distribution of *Prune* is just the reverse of that of the *Grow* operation. An example of the *Grow* and *Prune* operations are shown in Figure 3.1.

When the *Change* operation is chosen, an internal node is chosen uniformly from the available candidates in the current tree $\mathcal{T}^t$. While keeping the chosen splitting dimension and everything else unchanged, the spliting value is replaced by randomly sampling a new value from the available candidates. As a general rule, this new split value must not yield any empty terminal nodes in the subtree. After a new spliting value is obtained, data at the terminal nodes of the subtree are rearranged in order to form the new treed model $\mathcal{T}^*$. Going from $\mathcal{T}^t$ to $\mathcal{T}^*$ or vice versa, the same internal node has to be chosen and the number of available split values are the same. Therefore,

$$q(\mathcal{T}^*, \boldsymbol{\rho}^* | \mathcal{T}^t, \boldsymbol{\rho}^t) = q(\mathcal{T}^t, \boldsymbol{\rho}^t | \mathcal{T}^*, \boldsymbol{\rho}^*), \tag{3.20}$$

which can be omitted from the computation of the MH acceptance ratio. An example of the *Change* operation is shown in Figure 3.2. This *Change* proposal is the standard

approach that have been adopted in many cases (Chipman et al., 1998, 2002; Gramacy and Lee, 2008). A more flexible approach would be to use non-uniform proposals by giving higher probability weights to points closer to the current split point. This can be achieved by computing the distance between the current split point and the available points, then recale each distance value with the sum of all distance values. The resulting numbers are then employed as the proposal probabilities. The rationale behind this is that the current split point represents a place where there is a significant change in the structure of the data. Therefore, proposing points nearby helps to explore more "good" trees than proposing points far away. As a result, mixing of the treed models is improved by changing the local partition boundaries. This modification shrinks the variation of the *Change* proposal, much like reducing the variance of a Gaussian proposal to increase acceptance rate in a typical MCMC setting. In the rest of the dissertation, usage of the non-uniform *Change* proposal will be explicitly mentioned, otherwise the uniform version is assumed.

When the *Swap* operation is chosen, a parent-child pair of internal nodes are uniformly selected among the available candidates in the current tree $\mathcal{T}^t$. Then, the parent node's splitting rule is swapped with that of the child node. When the parent and child nodes split at different dimensions, there are the following cases of swapping:

- If the child node is on the right, the left subtree of the parent node swaps with the left subtree of the child node (see Figure 3.3).

- If the child node is on the left, the right subtree of the parent node swaps with

the right subtree of the child node (see Figure 3.4).

- If both children nodes have the same splitting rule, their splitting rules are swapped with that of the parent node, and the right subtree of the left child node swaps with the left subtree of the right child node (see Figure 3.5).

When the parent and child nodes split at the same dimension, swapping their splitting rules would yield empty terminal nodes. Instead of swapping, as noted by Gramacy (2005), a rotation scheme is used in this situation. If the child node is on the right, a left rotation is done, or vice versa. An example of rotations is shown in Figure 3.6. In all cases, data at the terminal nodes are adjusted to form the new treed model $\mathcal{T}^*$. Since both the current and the proposed treed model have the same number of parent-child internal nodes pairs, the proposal is symmetric. Therefore,

$$q(\mathcal{T}^*, \boldsymbol{\rho}^* | \mathcal{T}^t, \boldsymbol{\rho}^t) = q(\mathcal{T}^t, \boldsymbol{\rho}^t | \mathcal{T}^*, \boldsymbol{\rho}^*), \tag{3.21}$$

which can be omitted from the MH acceptance ratio. In the following sections, TPCGP is illustrated on a set of 1-d synthetic sinusoidal data and a set of 2-d real precipitatoin data.

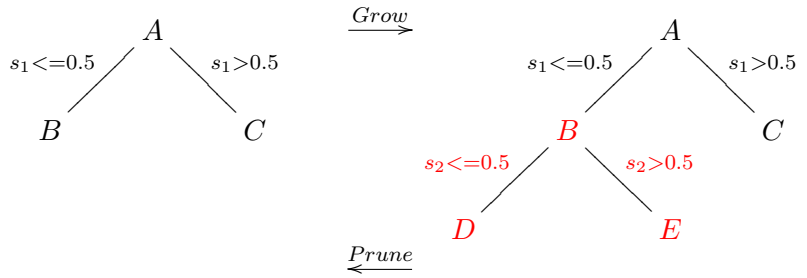


Figure 3.1: An example of the *Grow* and *Prune* operations

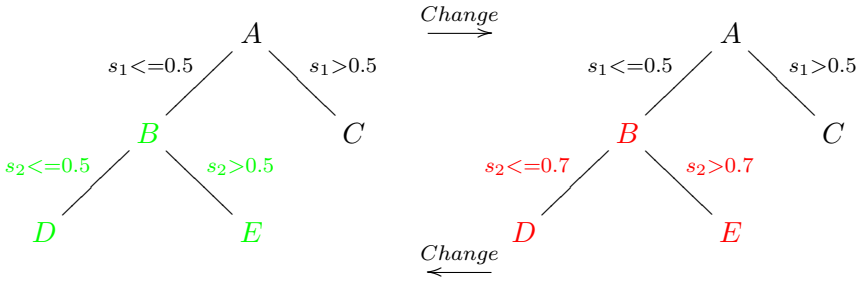


Figure 3.2: An example of the *Change* operation

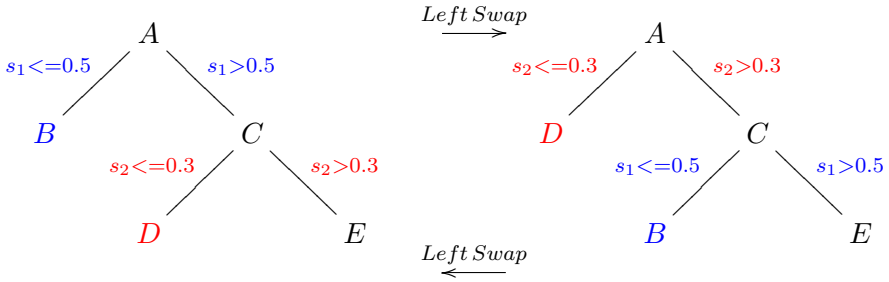


Figure 3.3: An example of the left *Swap* operation

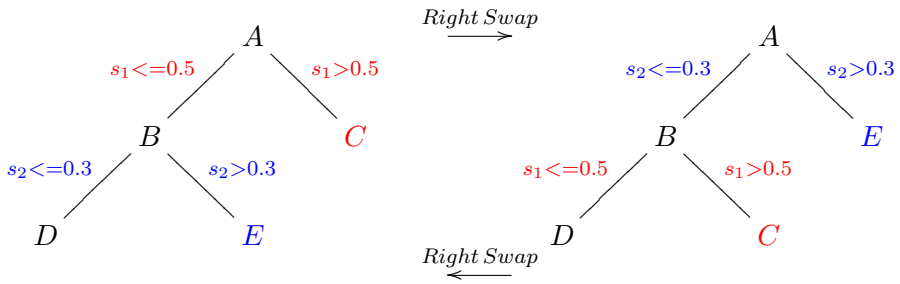


Figure 3.4: An example of the right *Swap* operation

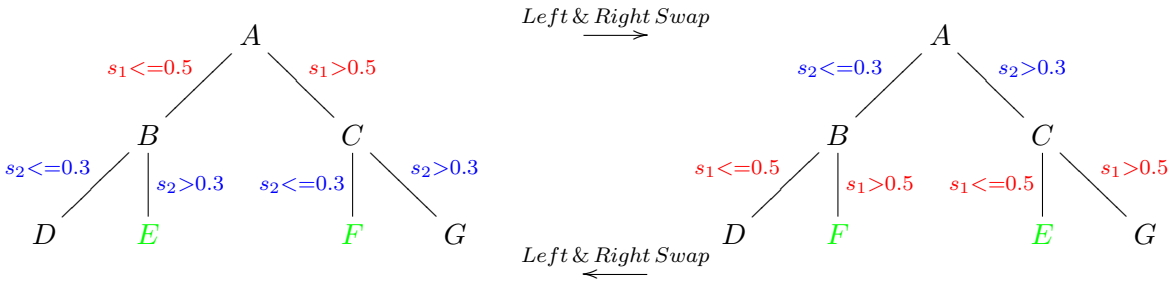


Figure 3.5: An example of the left & right *Swap* operation

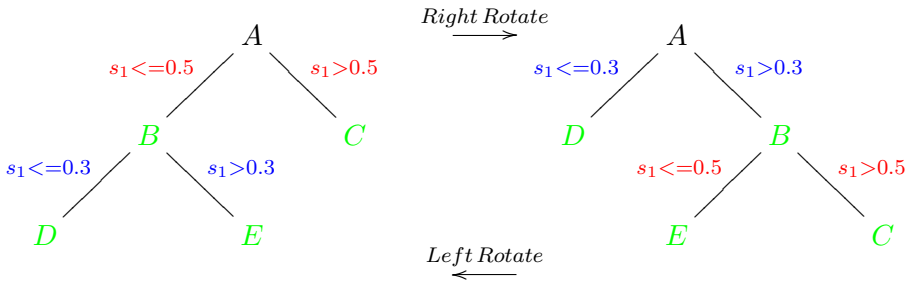


Figure 3.6: An example of tree rotations

## 3.5 Illustration

### 3.5.1 1-d Synthetic Sinusoidal Data

A set of 200 data points $y(\mathbf{s})$ (shown in Figure 3.7) is generated by sampling from the following response,

$$z(s) = \begin{cases} 2\sin(\frac{\pi s}{5}) + 1.4\cos(\pi s) & s < 9.6, \\ 0.2\sin(\frac{\pi s}{2}) & \text{otherwise}, \end{cases} \tag{3.22}$$

and adding $N(0, 0.1^2)$ noise to the sampled points, that is,

$$y(s) = z(s) + \epsilon, \quad \epsilon \sim N(0, 0.1^2). \tag{3.23}$$

There are 160 data points located at $s < 9.6$ and 40 at $s \geq 9.6$. TPCGP and the

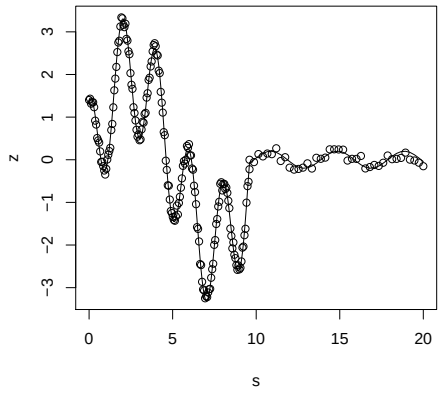


Figure 3.7: 1-d sinusoidal data (*circles*) generated by adding $N(0, 0.1^2)$ noise to Equation (3.23)

standard, non-partitioning process convolution GP model (PCGP) are applied to this dataset based on a Gaussian kernel with standard deviation following the Higdon's rule

of thumb (Section 2.1). The tree prior $P(\mathcal{T})$ has parameters $\alpha_T = 0.95$ and $\beta_T = 0.5$. The number of terminal nodes given by this prior has a mean of 7 and a range between 1 to 30 with non-negligible probability. The other conjugate priors are specified as

$$\phi_\nu | b_y, \boldsymbol{\rho}, \mathcal{T} \sim G(a_y = 1,\ b_y),$$

$$\mathbf{x}_\nu | \lambda_\nu, \boldsymbol{\rho}, \mathcal{T} \sim N_{m_\nu}(\mathbf{0},\ (\lambda_\nu \mathbf{I}_{m_\nu})^{-1}), \quad \lambda_\nu | b_x, \boldsymbol{\rho}, \mathcal{T} \sim G(a_x = 1,\ b_x),$$

$$b_y \sim G(\tau_y = 1,\ \xi_y = 1000), \quad b_x \sim G(\tau_x = 1,\ \xi_x = 1000).$$

The linear term is not being fitted since the data has zero mean. According to Chipman et al. (1998), the posterior distribution of a treed model is often multimodal, and RJ-MCMC is likely to get stuck in a local mode. One suggested solution is to restart the RJ-MCMC multiple times (starting with a different random seed) and average the results. Having restarts might seem time consuming, however, computers nowadays have multiple cores/threads per CPU, so multiple RJ-MCMCs can be run in parallel on a single chip. Following this approach, eight independent runs of TPCGP and PCGP are performed. The simulations are repeated for a number of bases ranging from 10 to 30. Each run has 3000 iterations with the first 2000 as burn-in, giving a total of 1000 posterior samples. Figure 3.8 displays the posterior predictive mean surfaces (*solid*) with the corresponding 90% intervals (*dashed*) for $m = 10$, 20, and 30. The green dotted lines depict the partition boundary of the MAP (*maximum a posteriori*) treed model, which is the one with the highest conditional posterior probability among the RJ-MCMC samples. For $m = 10$, PCGP produces a larger posterior predictive interval for $s > 9.6$, which does not match the data variability in that region. In contrast, TPCGP provides

a more reasonable interval for the same region. For $m = 20$, PCGP still produces a posterior predictive interval as large as that of $m = 10$, but TPCGP has an interval well matching the data variability. For $m = 30$, both models seem to have comparable results, but if one looks closely, PCGP's interval is still a little too large compared to that of TPCGP for $s > 9.6$. To better quantify the performance of the two models, the Deviance Information Criterion (DIC) is computed for each model for $m = 10$ through $m = 30$, and is shown in Figure 3.9. In general, TPCGP has lower DIC values, or better model fitting, than PCGP. However, the difference in model performance diminshes as the number of bases increases. In practical applications, the number of bases may be under-specified, in this case TPCGP seems to be a better choice for maintaining reasonable model performance. Furthermore, since a separate mearsurement error term can be assumed for each partition, TPCGP can handle heteroscedasticity (input-dependent noise), whereas PCGP would not be able to capture the variation in noise. In summary, TPCGP is at least as good as PCGP when the number of bases is large, and works better if the number of bases is low and/or in the presence of heteroscedasticity.

Figure 3.10 shows DIC v.s. parameter values for each prior. The green circles represent the tested cases and the red circle depicts the default value that produced the results above. These results are based on a single run of the model. Note that TPCGP is not very sensitive to the prior parameters unless a high value is specified for $a_x$, $a_y$, and the shape parameters of the Gamma priors of $b_x$, and $b_y$. Increasing $a_x$ and $a_y$ reduces the prior mean and variance of $\lambda^{-1}$ and $\phi^{-1}$, respectively. Since $\lambda^{-1}$ and $\phi^{-1}$ represent the variance of background points and measurement error, too strong of a

prior would lead to bad model performance. On the other hand, $b_x$ and $b_y$ represent the prior rates of $\lambda$ and $\phi$, respectively. When the shape parameters of the Gamma priors of $b_x$ and $b_y$ increase, so do the prior mean and variance of $\lambda^{-1}$ and $\phi^{-1}$. Based on this dataset, it is unlikely for $\lambda^{-1}$ and $\phi^{-1}$ to be larger than 1. Consider the shape parameter of the tree prior, most of the DIC values are constant at the same value except there is a downward spike (better model performance) near 0.8 and 1. This may be removed by averaging the results over more runs of the model. Since majority of the DIC values are constant, it is fair to say that the model is not very sensitive to the shape parameter of the tree prior. For the kernel width, the DIC goes up for widths less than 0.75 and bigger than 1.5. This is expected because the kernel width directly affects the model fitting, too large or too small of a value would render the fitting too smooth or too rough, respectively. In general, TPCGP is not very sensitive to the prior parameters except for the extreme cases described above.

### 3.5.2 2-d Real Precipitation Data

Next, TPCGP is illustrated on a set of precipitation data obtained from an R package called *KriSp* (Furrer, 2006). This dataset contains total precipitation counts (in milliliter) recorded at $11,918$ locations over the contiguous U.S for April 1948. Observations have been standardized on the square-root scale by the *KriSp* package so that they are more closer to a Gaussian distribution (Johns et al., 2003). Each observation is referenced by a location recoreded in longitude and latitude. In order to apply TPCGP, the longitude/latitude representation is converted to a planar representation in kilome-

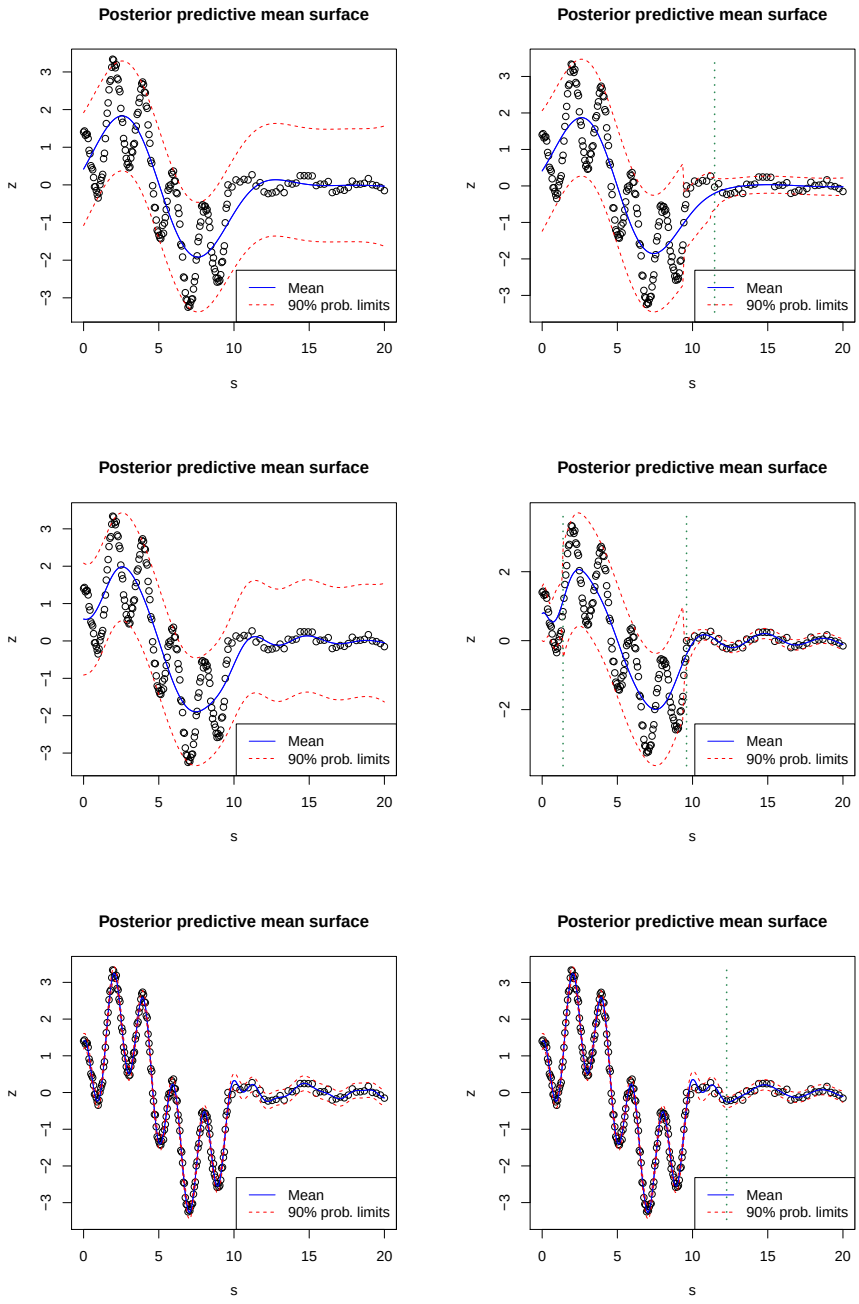


Figure 3.8: Posterior predictive summary from modeling of 1-d sinusoidal data based on a fixed Gaussian kernel using 10 (*top*), 20 (*middle*), and 30 (*bottom*) bases, with partitioning (*right*) and non-partitioning (*left*)

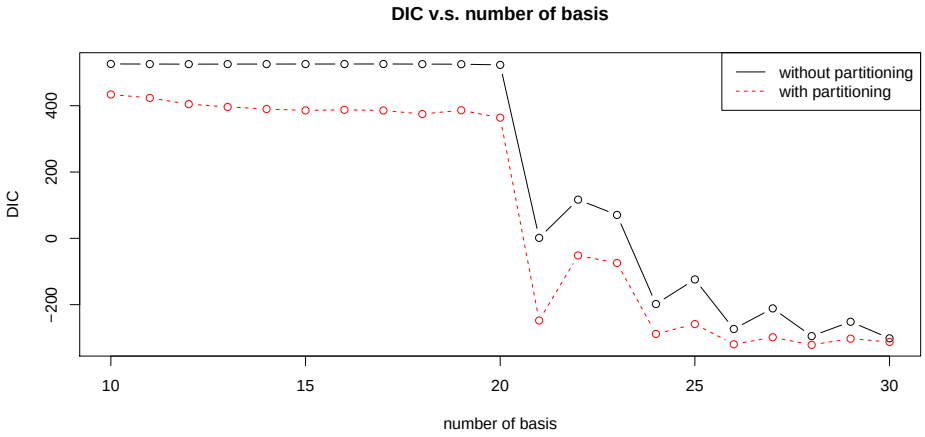


Figure 3.9: DIC v.s. number of basis from modeling of 1-d sinusoidal data

ters using the Universal Transverse Mercator coordinate system. Top panel of Figure 3.11 shows $10,000$ observations randomly sampled from the full dataset. The remaining $1,918$ observations are left out for prediction comparison. A basis grid of $3,646$ evenly spaced points (*solid black*) is fixed over the observation domain and the spacing between any two adjacent (vertically or horizontally) basis points is 50 kilometers. The bottom panel of Figure 3.11 shows the elevation over the contiguous U.S., which is incoporated into TPCGP as a covariate along with an intercept term. A Bézier kernel with $\kappa = 0.5$ is used to induce a sparse kernel matrix $\mathbf{K}$ so that dedicated matrix routines can be used to speed up the computation of the likelihood. The chosen value of $\kappa$ comes our experience of modeling this dataset, that is, decreasing $\kappa$ generally improves model fitting, but improvment diminishes for values lower than 0.5. The kernel covariance matrix $\mathbf{Q}^{-1}$ is specified via the Higdon's rule of thumb (described in Section 2.1), that is, as a diagonal matrix whose diagonal components are $(3 \times 50)^2$. This makes the compact

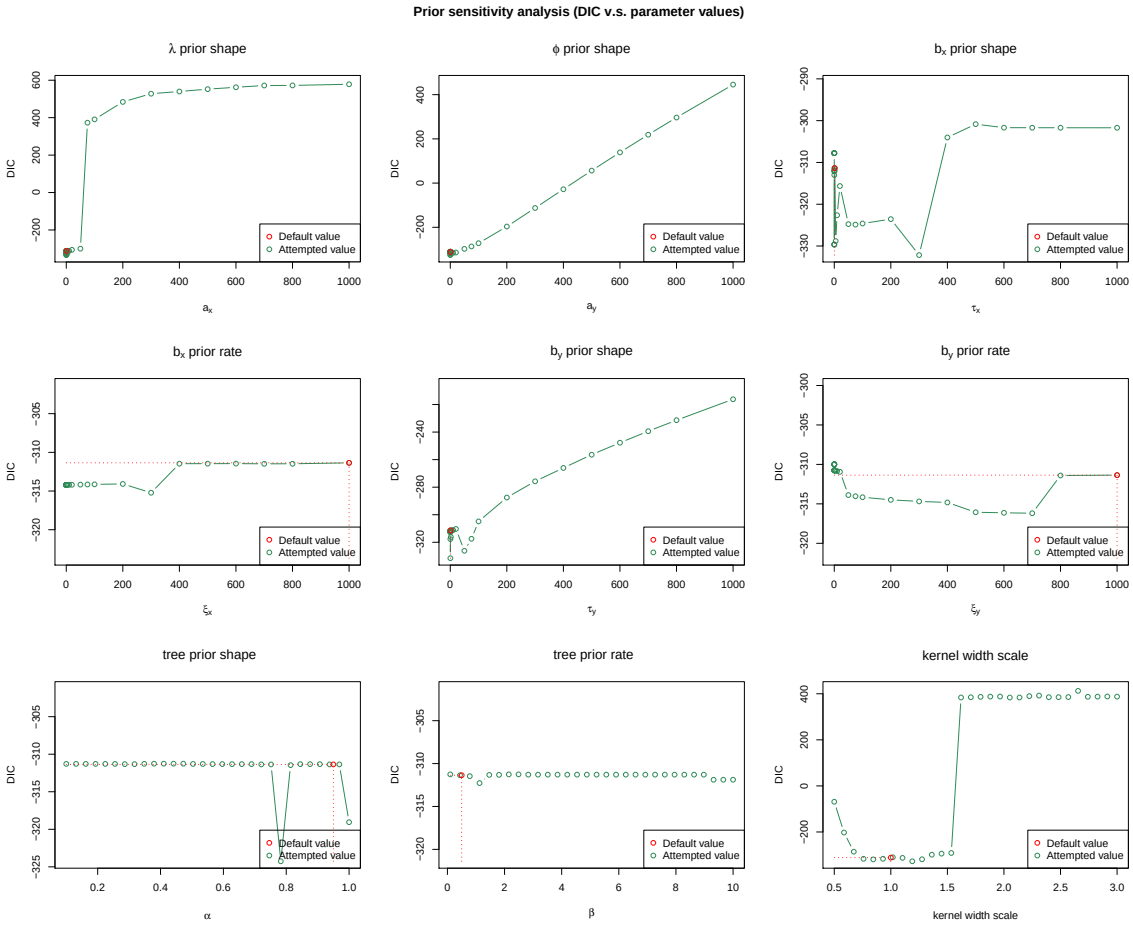


Figure 3.10: Sensitivity analysis of prior parameters of TPCGP with a fixed kernel on 1-d sinusoidal data

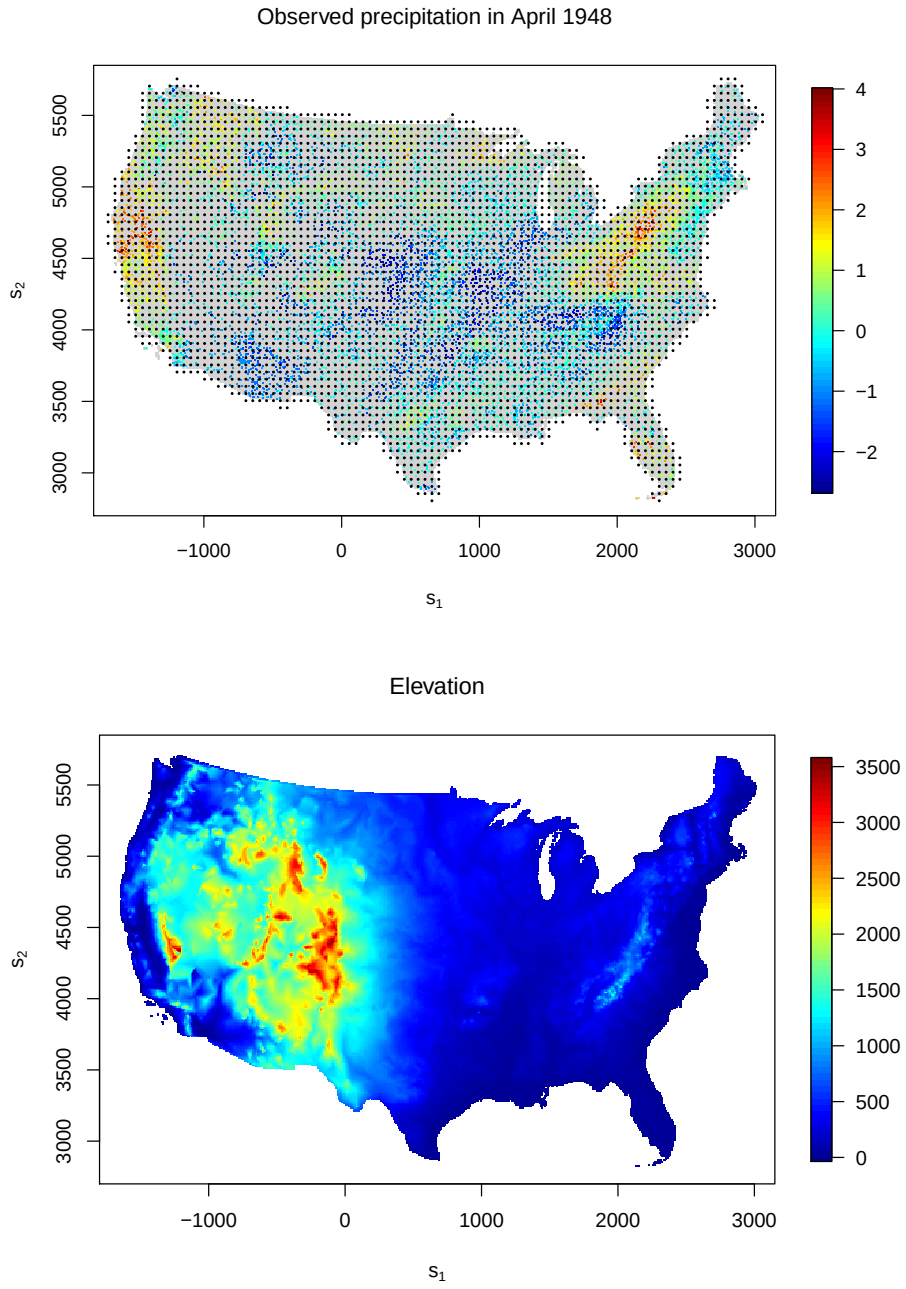


Figure 3.11: Total precipitation count for April 1948 (*top*) and elevation over the contiguous U.S. (*bottom*)

support of the kernel to have a radius of $3 \times 50\,\text{km} = 150\,\text{km}$. The tree prior $P(\mathcal{T})$ is given parameters $\alpha_T = 0.95$ and $\beta_T = 0.5$ as before. The other conjugate priors are specified as

$$\boldsymbol{\beta}|\boldsymbol{\beta}_0, \mathbf{C}, \boldsymbol{\rho}, \mathcal{T} \sim N_{p+1}(\boldsymbol{\beta}_0,\ \mathbf{C}^{-1}), \quad \phi_\nu|b_y, \boldsymbol{\rho}, \mathcal{T} \sim G(a_y = 10,\ b_y),$$

$$\mathbf{x}_\nu|\lambda_\nu, \boldsymbol{\rho}, \mathcal{T} \sim N_{m_\nu}(\mathbf{0},\ (\lambda_\nu \mathbf{I}_{m_\nu})^{-1}), \quad \lambda_\nu|b_x, \boldsymbol{\rho}, \mathcal{T} \sim G(a_x = 10,\ b_x),$$

$$\boldsymbol{\beta}_0 \sim N(\boldsymbol{\mu} = (0,0)^\top,\ \mathbf{B}^{-1} = 0.1\mathbf{I}_2), \quad \mathbf{C} \sim W((\varphi \mathbf{V})^{-1} = 10\mathbf{I}_2,\ \varphi = 100),$$

$$b_y \sim G(\tau_y = 1,\ \xi_y = 1000), \quad b_x \sim G(\tau_x = 1,\ \xi_x = 1000).$$

Note that instead of having a separate $\boldsymbol{\beta}$ for each partition, a single $\boldsymbol{\beta}$ is assumed over the entire domain. Using multiple $\boldsymbol{\beta}$s has been attempted, but improvement in the resulting modeling fitting is very small. In addition, assuming a single $\boldsymbol{\beta}$ keeps the resulting process continuous, which is a more reasonable behavior for environmental processes such as precipitation. The first attempt in applying TPCGP to this dataset shows that the treed model $\mathcal{T}$ quickly gravitates towards a particular tree structure (a local mode in the posterior) and becomes stuck there. Although the mixing of each parameter looks fairly good, the treed model posterior is not being well explored. In fact, fully exploring the treed model posterior is not necessary (and also very difficult) to obtain good model performance. Exploring a few "good" trees is usually enough to produce promising results. However, a quick look at the fitted residuals shows that there are some apparent features in the data that have not been captured by the model. Increasing the number of RJ-MCMC iterations does not solve this problem. A simple solution suggested by Chipman et al. (1998) is to restart the RJ-MCMC several times

with different random seeds and average the results. Following this approach, TPCGP is restarted on the same dataset for a total of 8 times. Each run has $3,000$ iterations, from which the last $1,000$ are saved as posterior samples. Together, these 8 independent runs produce a total of $8,000$ samples. To further improve results, a non-uniform version of the *Change* proposal (described in Section 3.4) is used for all simulations, which help to explore more "good" trees.

The posterior predictive mean surface is shown in the top panel of Figure 3.12 with the corresponding 90% interval width shown on the bottom. These surfaces are obtainted by averaging all $8,000$ samples. The *white dashed* lines depict the partition boundary of the MAP treed model, and most of them occur at where the variability of the data changes. The mean and variance of the background points $\mathbf{x}$ are shown in Figure 3.13. In general, the mean and variance of $\mathbf{x}$ determine the magnitude and variation of the resulting fitted surface, respectively. Regions with higher data variability is often captured with higher variance in $\mathbf{x}$. Posterior mean of the linear term $\mathbf{F}\boldsymbol{\beta}$ is shown in the top panel of Figure 3.14, which ranges from about $-0.11$ to 0.05. Based on the magnitude of the posterior predictive mean surface (ranges from about $-2$ to 3), this shows that most of the features in the data have been captured by the stochastic component and very little is explained by the linear term. The observational error SD is shown in the bottom panel of Figure 3.14. The error term has a tendency of picking up patterns unexplained by the stochastic component and the linear term. Therefore, the estimated error SD is high whenever the nearby observations have high variation which is relatively harder to be explained by the stochastic component. Figure 3.15 shows the

log conditional of treed models $\log\{P(\mathcal{T}, \boldsymbol{\rho}|\cdots)\}$, number of partitions visited, and the accepted tree operations from all $8,000$ samples. The result from each run of the model is depicted by a different color. The number of partitions ranges from 4 to 7, however, most of the runs are stuck at a particular number of partitions. This is why averaging multiple runs of the model is needed to improve the model performance. Despite the bad mixing in the number of partitions visited, $\log\{P(\mathcal{T}, \boldsymbol{\rho}|\cdots)\}$ actually mixes reasonably well for most of the runs, and the acceptance rate of treed models is nearly 19% mostly due to the *Change* operation. Although results might be further improved by averaging over more runs, the model performance based on these 8 runs is sufficiently good. This can be shown by the fitted residuals shown in Figure 3.16. Visually, the fitted residuals appear to be randomly scattered about zero and no significant patterns are visible. Although residuals are sparse at higher elevations as shown in the bottom panel of Figure 3.16, the vertical spread is roughly constant from the lowest to highest elevation. The "Moran's I test" (Moran, 1950) is performed on the residuals to find out how correlated they are. The employed function is named *Moran.I* from the R package called *ape*. The computed p-value is about 0.09, which is higher than the commonly specified significance level of 0.05. Thus, there is NOT enough evidence to reject the null hypothesis that the residuals are uncorrelated at the 0.05 significance level. This confirms that TPCGP is working properly. Predictions are made at those $1,918$ observation locations that have been omitted from the model training. Residuals resulted from predictions are shown in Figure 3.17, and seem to be randomly scattered about zero with a mean squared error (MSE) of 0.138. For the computational speed,

each of the 8 runs is able to complete in less than 3 hours on a desktop computer with Intel Core i5 CPU at 2.8 GHz. This speed is much faster than that of a standard GP model fitted to this dataset.

## 3.6 Conclusion

This chapter provides a detail formulation of the treed process convolution GP model. Specifically, nonstationarity is induced in the resulting GP by partitioning the spatial domain via a binary tree generating procedure and allowing a separate latent process for each partition. A Bayesian approach is used to explore the treed model space and estimate the model parameters simultaneously. Illustrations of this model are provided on a 1-d synthetic sinusoidal dataset and a 2-d real precipitation dataset. Results show that TPCGP provides promising performance in terms of model fitting, prediction, and computational speed. In the next chapter, this basic setup of TPCGP will be extended to improve model performance by allowing kernels to vary across partitions. Comparison with other computationally efficient models in the literature will also be given.

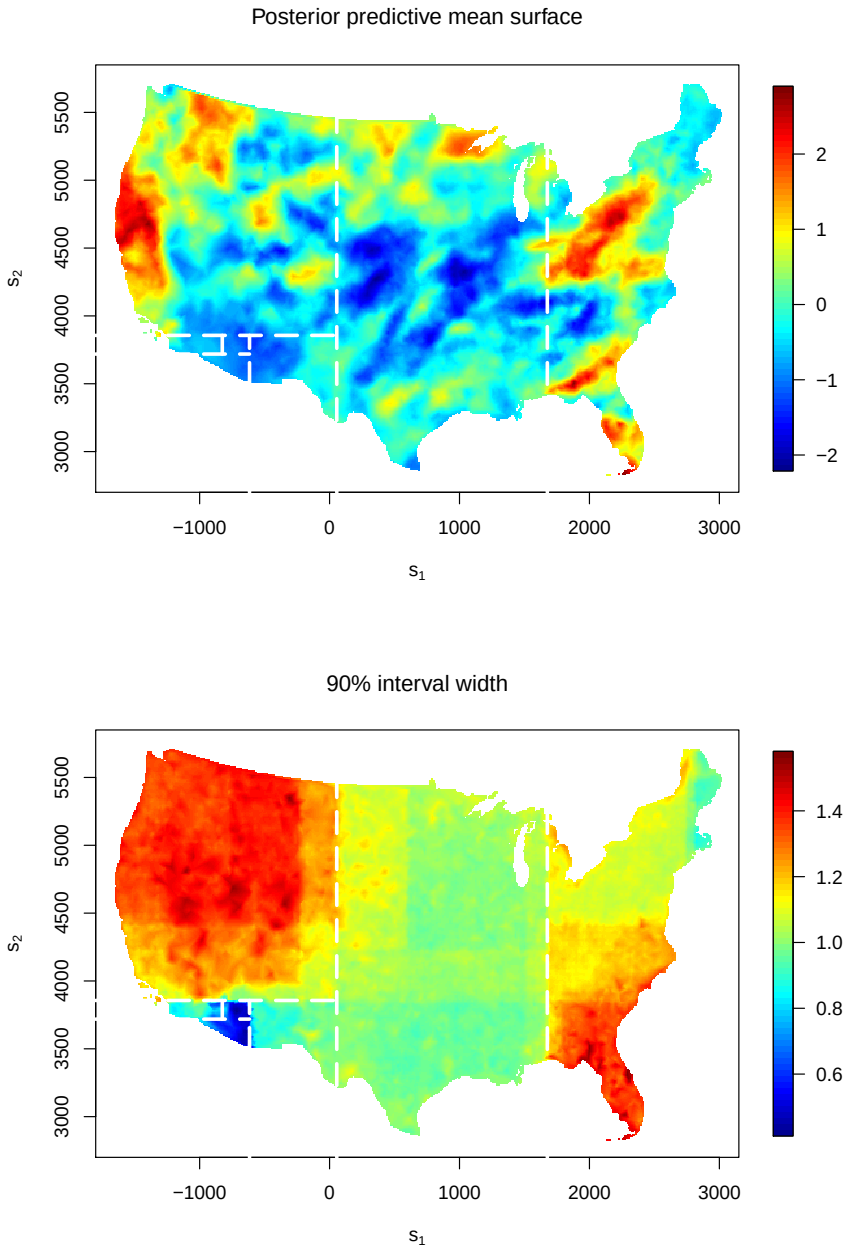


Figure 3.12: Posterior predictive mean surface (*top*) and 90% posterior predictive interval width (*bottom*)

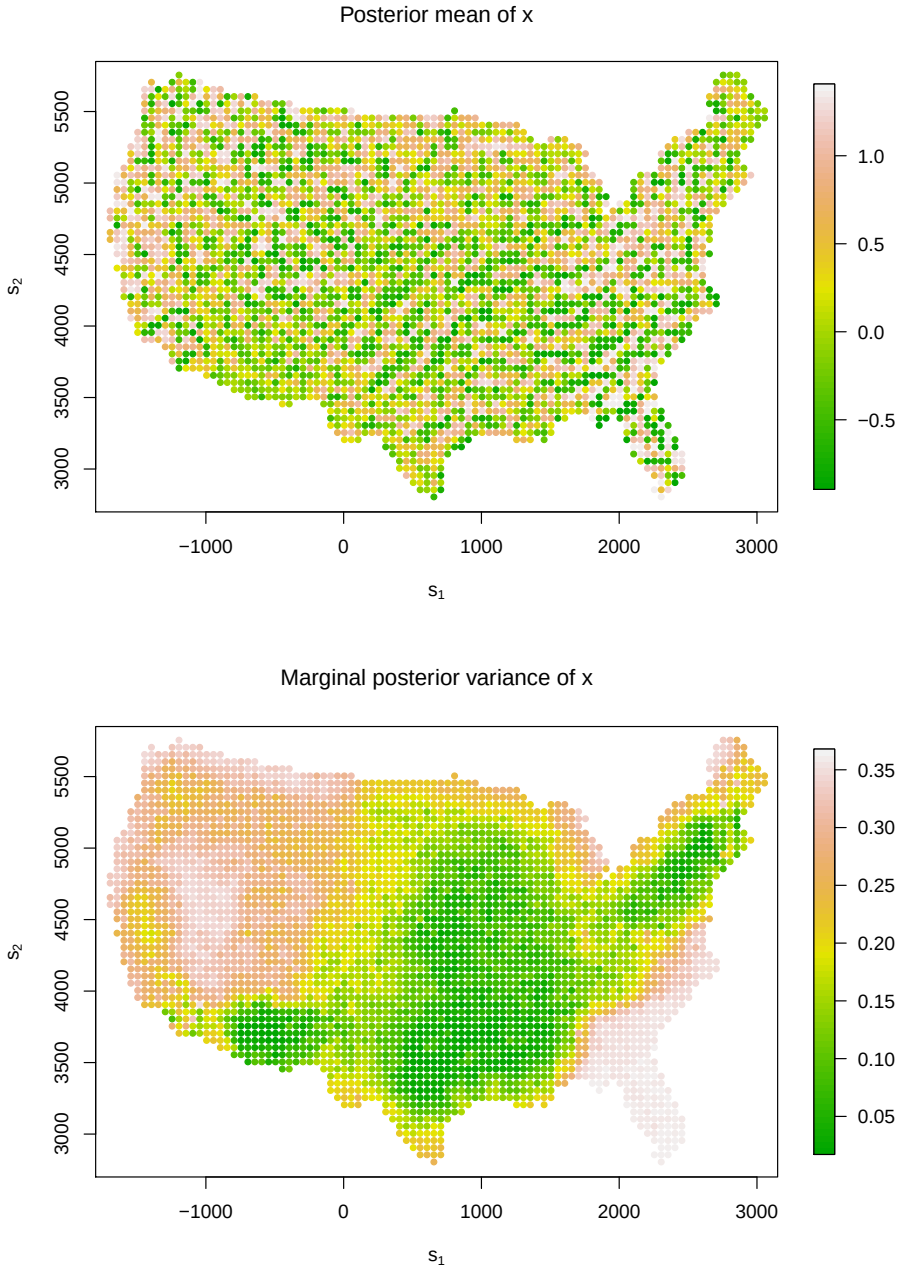


Figure 3.13: Posterior mean (*top*) and variance (*bottom*) of $\mathbf{x}$

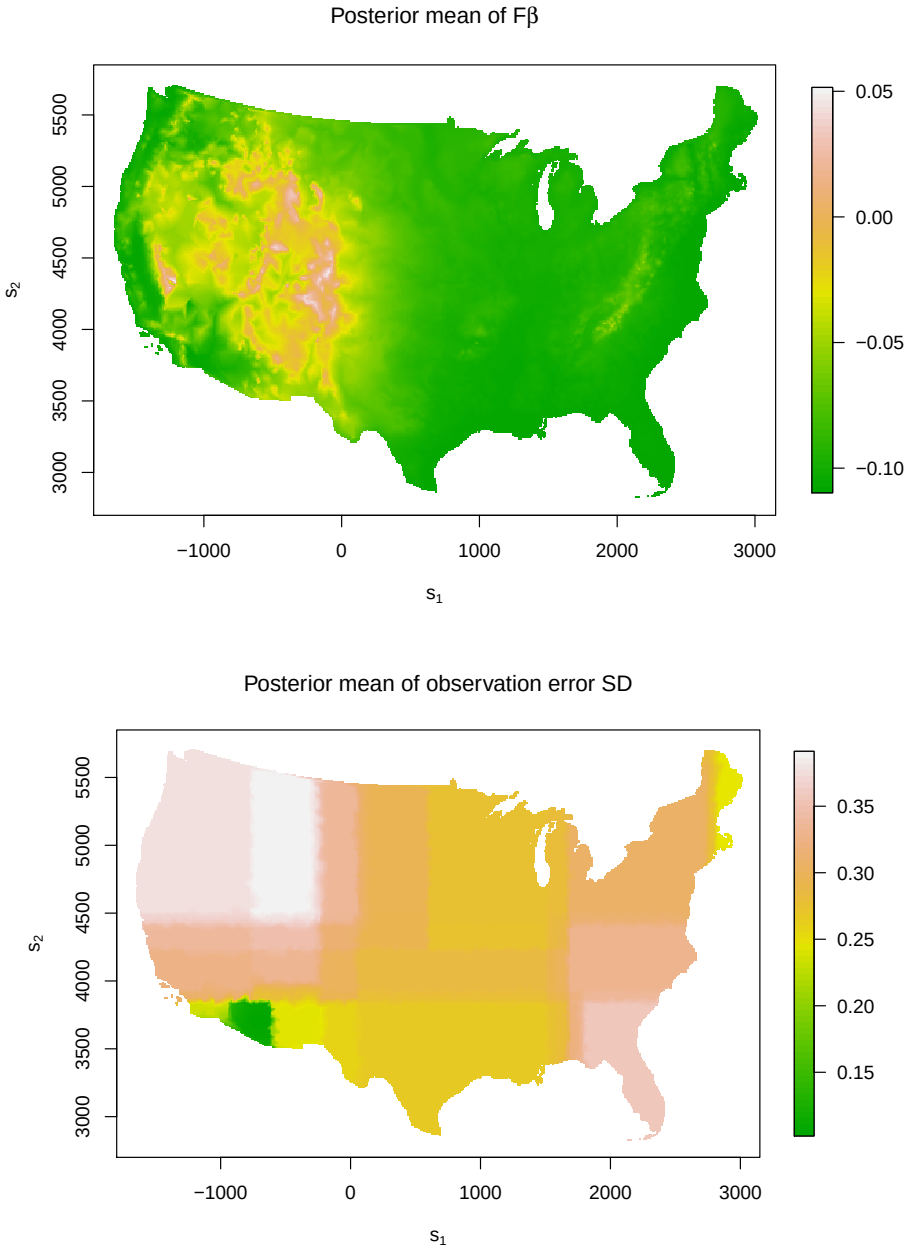


Figure 3.14: Posterior mean of $\mathbf{F}\boldsymbol{\beta}$ (*top*) and observation error standard deviation $\sqrt{1/\phi}$ (*bottom*)

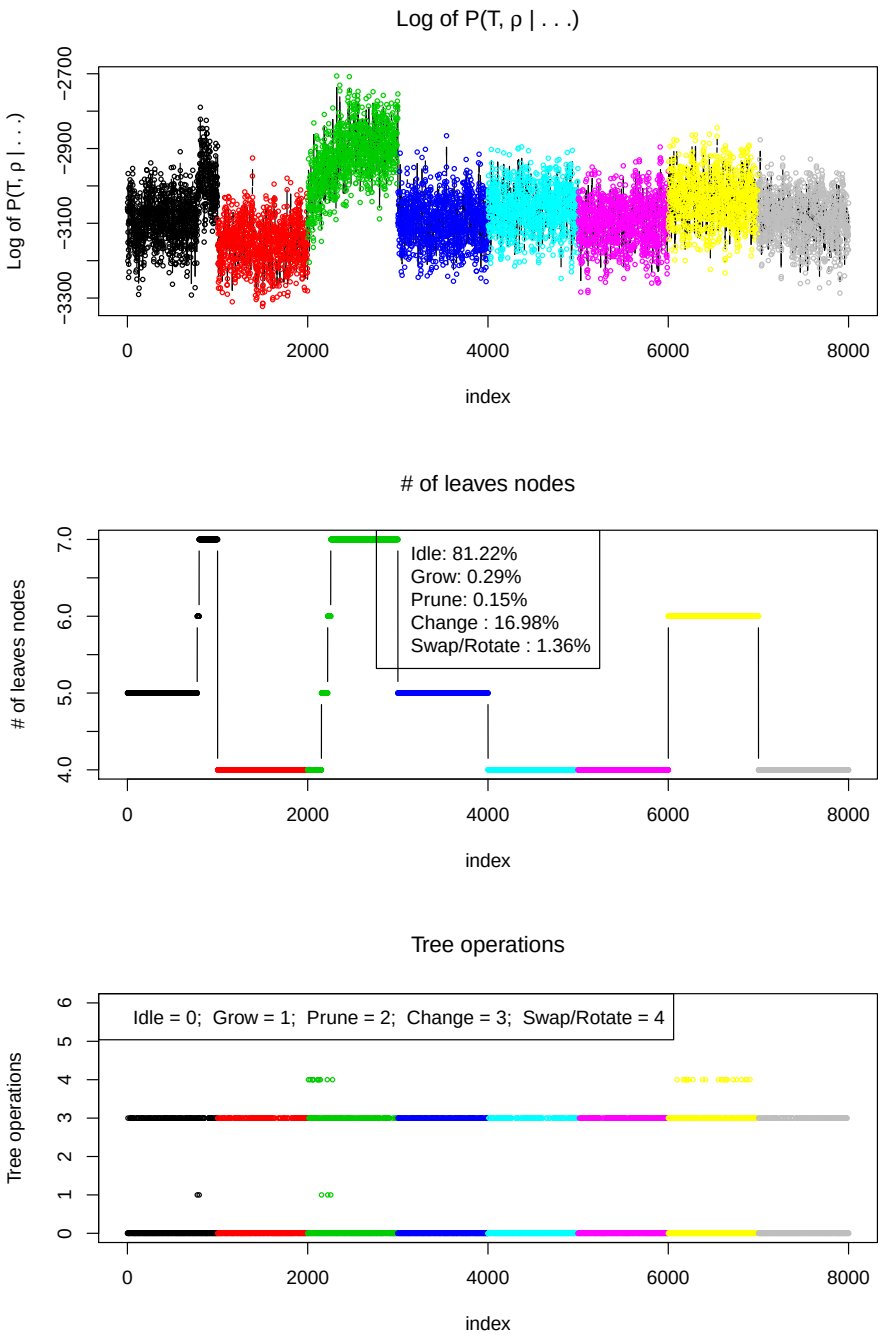


Figure 3.15: Log conditional of treed models (*top*), number of partitions visited (*middle*), and the accepted tree operations (*bottom*)

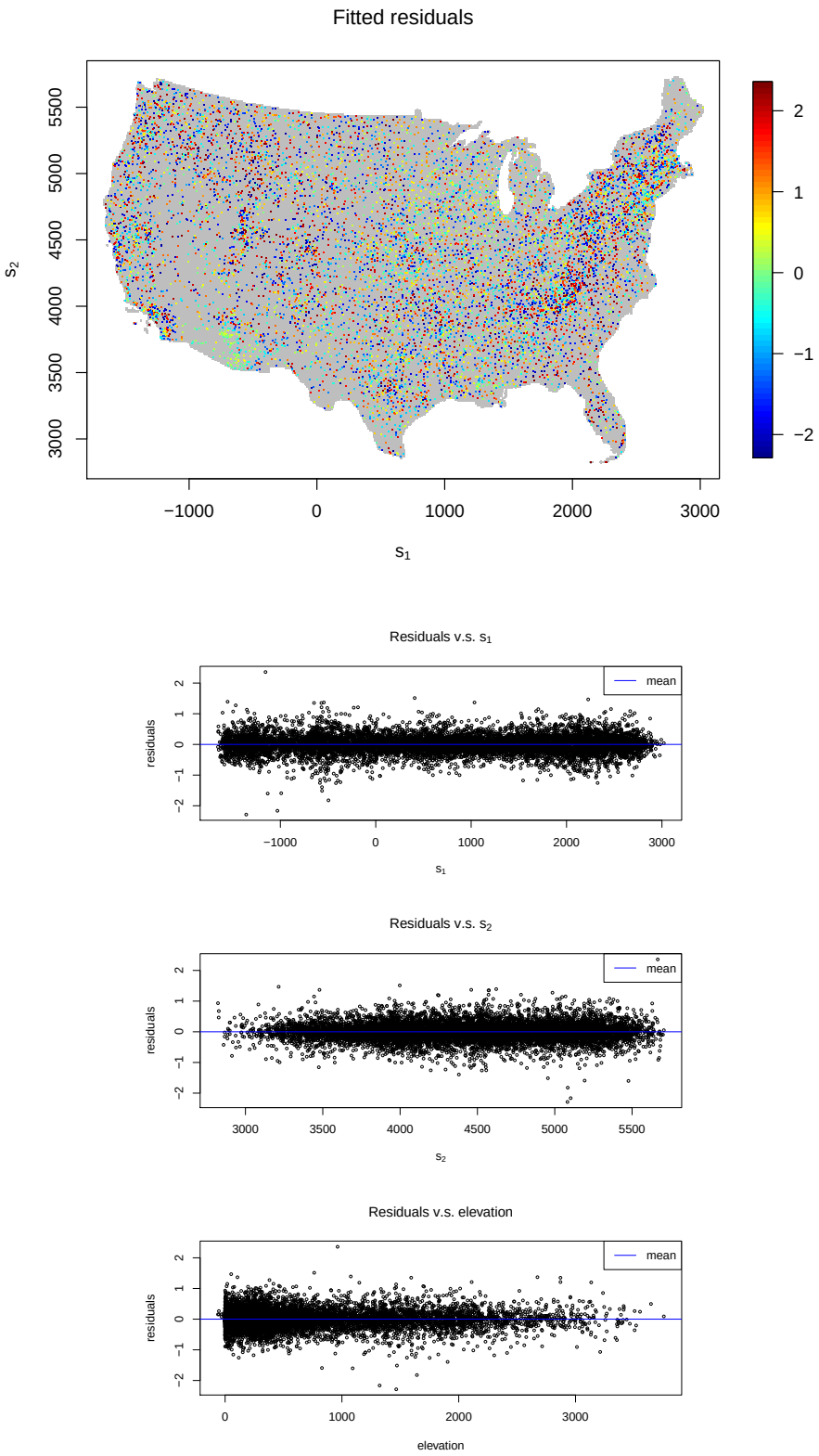


Figure 3.16: Fitted residuals from TPCGP with a fixed kernel

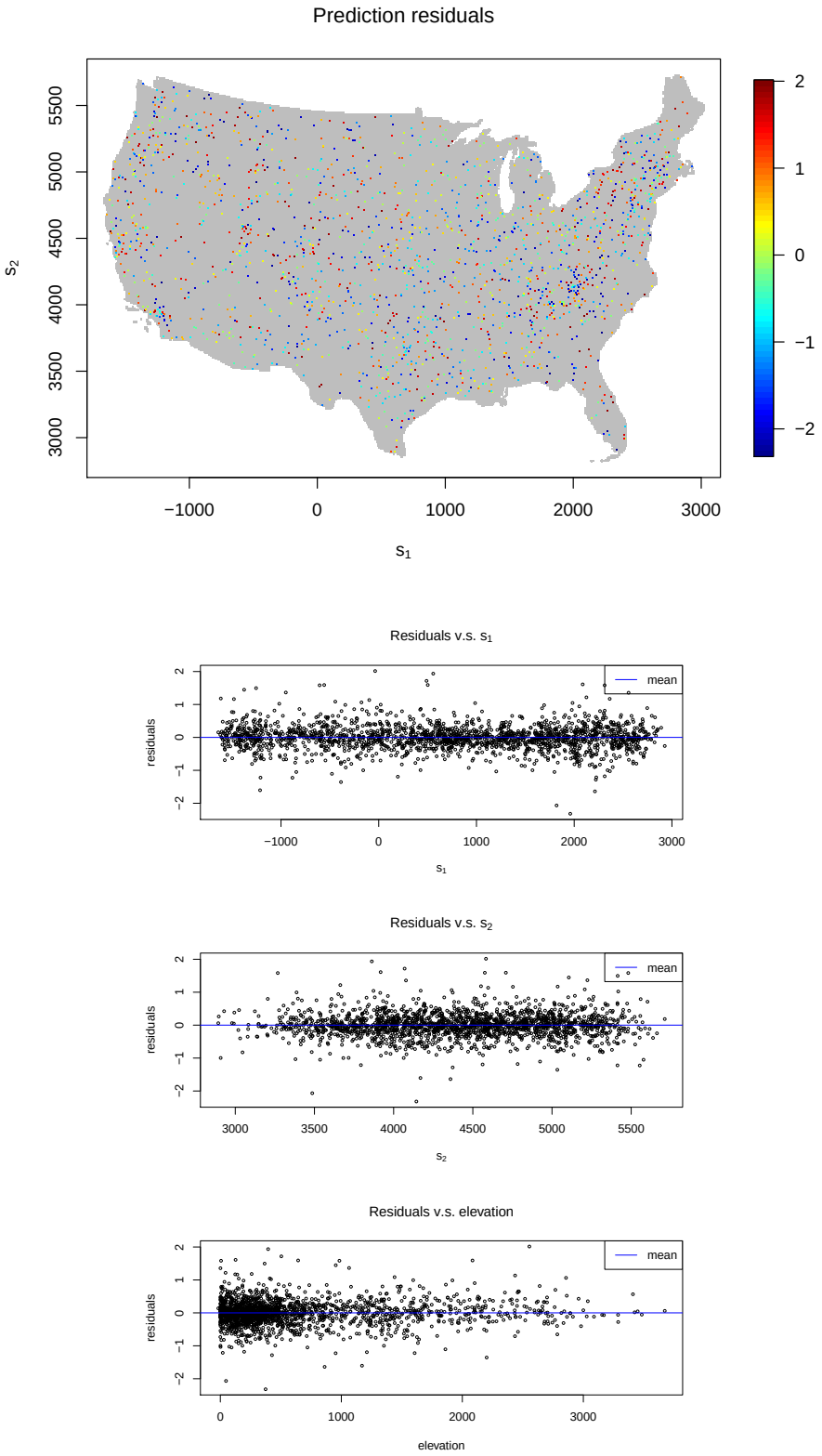


Figure 3.17: Prediction residuals from TPCGP with a fixed kernel

# Chapter 4

# Variable Kernels Across Partitions

## 4.1 Introduction

The model setup given in the previous section assumes that the kernel is fixed for all partitions. Although this may work well in many applications, there are situations where fixing the kernel is not good enough. For example, if the process of interest has high variability in one region but is smooth in another, fixing the kernel is unlikely to well model both regions. Furthermore, for applications whose dimension is larger than one, the anisotropic stucture may vary over space, and this can not be well captured with a fixed kernel. Improvement in the model can be made by treating the kernel as a random function of space and estimate it along with other parameters. A similar approach has been attempted by Higdon et al. (1999), where the size and orientation of each kernel depends on its spatial location $\mathbf{s}$, i.e., the kernel covariance matrix is re-parameterized in terms of elliptical projections and priors are imposed on the corresponding parameters.

The drawback is that some parameters have to be fixed, and overfitting and mixing problems are found when these parameters are allowed to vary (Swall, 1999). In contrast, TPCGP can be extended by associating a separate kernel precision matrix $\mathbf{Q}_\nu$ for each partition. Specifically, each basis point $\mathbf{u}$ is treated as the center of a kernel, and for all $\mathbf{u}$'s in the same partition, the corresponding kernels have the same precision matrix. The rationale behind this is that the partitions are suppose to figure out regions wherein the data structure is more homogeneous and the kernels in each partition would capture the local behavior. As a result, this extension provides more flexibility to the model. The kernels can be estimated using a Bayesian approach by imposing a prior distribution on the kernel precision matrix. A natural choice for the prior of $\mathbf{Q}_\nu$ is the Wishart distribution:

$$\mathbf{Q}_\nu|\boldsymbol{\rho}, \mathcal{T} \sim W((\psi\mathbf{H})^{-1}, \psi), \tag{4.1}$$

where $\psi$ and $\mathbf{H}^{-1}$ denotes the degrees of freedom and mean of the distribution, respectively. Under this prior, the full conditional posterior distribution for $\mathbf{Q}_\nu$ is given by

$$\begin{aligned} & P(\mathbf{Q}_\nu|\cdots) \\ \propto & \prod_{\nu=1}^{b} \exp\Big\{-\frac{\phi_\nu}{2}(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})^\top(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})\Big\} \times \\ & |\mathbf{Q}_\nu|^{(\psi-r-1)/2} \exp\Big\{-\frac{1}{2}\mathrm{tr}(\psi\mathbf{H}\mathbf{Q}_\nu)\Big\} \\ \propto & |\mathbf{Q}_\nu|^{(\psi-r-1)/2} \exp\Big\{-\frac{1}{2}\Big(\mathbf{x}^\top\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{K}\mathbf{x} - 2\mathbf{w}^\top\boldsymbol{\Phi}\mathbf{K}\mathbf{x} + \mathrm{tr}(\psi\mathbf{H}\mathbf{Q}_\nu)\Big)\Big\}. \end{aligned} \tag{4.2}$$

Note that $P(\mathbf{Q}_\nu|\cdots)$ can not be obtained in closed form, and the Metropolis-Hastings algorithm is used to obtain posterior samples of $\mathbf{Q}_\nu$. Also, note that $\mathbf{Q}_\nu$ can not be

integrated out from the treed model posterior, a Jacobian term should be incorporated into the MH acceptance ratio for the *Grow* and *Prune* step according to the rules of RJ-MCMC. However, the augmented $\mathbf{Q}_\nu$ is to be proposed from its prior, so the Jacobian term can be omitted. The conditional posterior distribution for $(\mathcal{T}, \boldsymbol{\rho})$ is given by

$$\begin{aligned} P(\mathcal{T}, \boldsymbol{\rho}|\cdots) \propto & \left( \prod_{\nu=1}^{b} \left(\frac{1}{2\pi}\right)^{(n_\nu+m_\nu)/2} |(\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})|^{-1/2} \frac{b_y^{a_y}}{\Gamma(a_y)} \frac{b_x^{a_x}}{\Gamma(a_x)} \times \right. \\ & \Gamma(n_\nu/2 + a_y)\Big(b_y + \frac{1}{2}\Big(s_\nu^2 + (\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}_\nu)^\top \mathbf{R}_\nu^{-1} (\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}_\nu)\Big)\Big)^{-(n_\nu/2+a_y)} \times \\ & \Gamma(m_\nu/2 + a_x)\Big(\frac{1}{2}\mathbf{x}_\nu^\top \mathbf{x}_\nu + b_x\Big)^{-(m_\nu/2+a_x)} \times \\ & \left. \frac{|\mathbf{Q}_\nu|^{(\psi-r-1)/2} |\psi \mathbf{H}|^{\psi/2}}{2^{\psi r/2} \Gamma_r(\psi/2)} \exp\Big\{ -\frac{1}{2} \mathrm{tr}(\psi \mathbf{H} \mathbf{Q}_\nu) \Big\} \right) P(\boldsymbol{\rho}|\mathcal{T}) P(\mathcal{T}), \end{aligned} \tag{4.3}$$

where $\mathbf{R}_\nu = \mathbf{C}^{-1} + (\mathbf{F}_\nu^\top \mathbf{F}_\nu)^{-1}$ and this posterior can not be obtained in closed form. Illustration of the extended TPCGP on simulated and real datasets are provided in the following sections.

## 4.2 Illustration

### 4.2.1 1-d Synthetic Sinusoidal Data

A set of 100 data points $y(s)$ (shown in Figure 4.1) are generated by sampling from the following response,

$$z(s) = \begin{cases} \sin(\frac{\pi s}{5}) + (\frac{-0.49s}{9.6} + 0.5)\cos(\pi s) & s < 9.6, \\ \frac{s}{9.6} - 1 & \text{otherwise}, \end{cases} \tag{4.4}$$

and adding $N(0, 0.1^2)$ noise to the sampled points, that is,

$$y(s) = z(s) + \epsilon, \quad \epsilon \sim N(0, 0.1^2). \tag{4.5}$$

The response $z(s)$ has high frequency for $s < 9.6$, but a smooth and increasing trend on the opposite side. A Gaussian kernel with 30 bases, and a tree prior $P(\mathcal{T})$ with

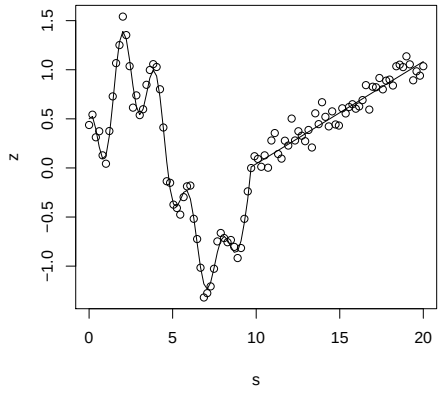


Figure 4.1: 1-d sinusoidal data (*circles*) generated from Equation (4.5)

parameters $\alpha_T = 0.95$ and $\beta_T = 0.5$ are chosen for the model. A separate linear term is allowed for each partition with covariates being the location $s$, along with an intercept term. Conjugate priors for the model parameters are specified as

$$\boldsymbol{\beta}_\nu | \boldsymbol{\beta}_0, \mathbf{C}, \boldsymbol{\rho}, \mathcal{T} \sim N_{p+1}(\boldsymbol{\beta}_0,\ (\phi_\nu \mathbf{C})^{-1}), \quad \phi_\nu | b_y, \boldsymbol{\rho}, \mathcal{T} \sim G(a_y = 1,\ b_y),$$

$$\mathbf{x}_\nu | \lambda_\nu, \boldsymbol{\rho}, \mathcal{T} \sim N_{m_\nu}(\mathbf{0},\ (\lambda_\nu \mathbf{I}_{m_\nu})^{-1}), \quad \lambda_\nu | b_x, \boldsymbol{\rho}, \mathcal{T} \sim G(a_x = 1,\ b_x),$$

$$\boldsymbol{\beta}_0 \sim N(\boldsymbol{\mu} = (0,0)^\top,\ \mathbf{B}^{-1} = \mathbf{I}_2), \quad \mathbf{C} \sim W((\varphi \mathbf{V})^{-1} = 0.1\mathbf{I}_2,\ \varphi = 10),$$

$$b_y \sim G(\tau_y = 1,\ \xi_y = 1000), \quad b_x \sim G(\tau_x = 1,\ \xi_x = 1000).$$

The prior for $\mathbf{Q}_\nu$ is specified as

$$\mathbf{Q}_\nu \sim W\Big((\psi \mathbf{H})^{-1} = 2.5\Big(\frac{20}{30-1}\Big)^{-2}\mathbf{I}_1,\ \psi = 1\Big),$$

where $\frac{20}{30-1}$ is the spacing between any two adjacent basis points, and the degree of freedom $\psi = 1$ makes this prior fairly non-informative. A Wishart distribution is used as the proposal for $\mathbf{Q}_\nu$ with a mean equal to the previous RJ-MCMC sample and the degrees of freedom equals to 100. The model results are shown in Figure 4.2, where the left panel corresponds to the full TPCGP with variable kernels; the middle and right panels corresponds to a TPCGP with a fixed kernel such that $\sqrt{\mathbf{Q}^{-1}}$ equals to 1 and 10 times the basis spacing, respectively. All three of them have similar MAP partitions, however, the full TPCGP with variable kernels is able to well capture both the high frequency and linear regions simultaneously. In fact, it fits the linear region much closer to the truth than the other two TPCGPs with a fixed kernel. This is achevied, as shown in the bottom left panel of Figure 4.2, by estimating a wider kernel for the linear region while keeping the kernel size small for the high frequency side. In contrast, using a fixed kernel can overfit the linear region if the width is too small or underfit the high frequency side if the width is too large. These results suggest that TPCGP with variable kernels is more flexible than its basic setup.

### 4.2.2 2-d Synthetic Exponential Data

For the next illustration, consider the following modified version of an exponential response surface $z(\mathbf{s})$ borrowed from Gramacy (2005),

$$z(\mathbf{s}) = 2s_1 \exp(-\mathbf{s}^\top \mathbf{\Sigma} \mathbf{s}), \quad \text{where} \quad \mathbf{s} = (s_1, s_2)^\top \in [-3, 6]^2,$$
$$\mathbf{\Sigma} = \begin{pmatrix} 1 & 0.7 \\ 0.7 & 1 \end{pmatrix} \quad \text{if} \quad s_1 < 3.5 \quad \text{and} \quad s_2 < 3.5, \quad \text{else} \quad \mathbf{\Sigma} = \mathbf{I}_2.$$

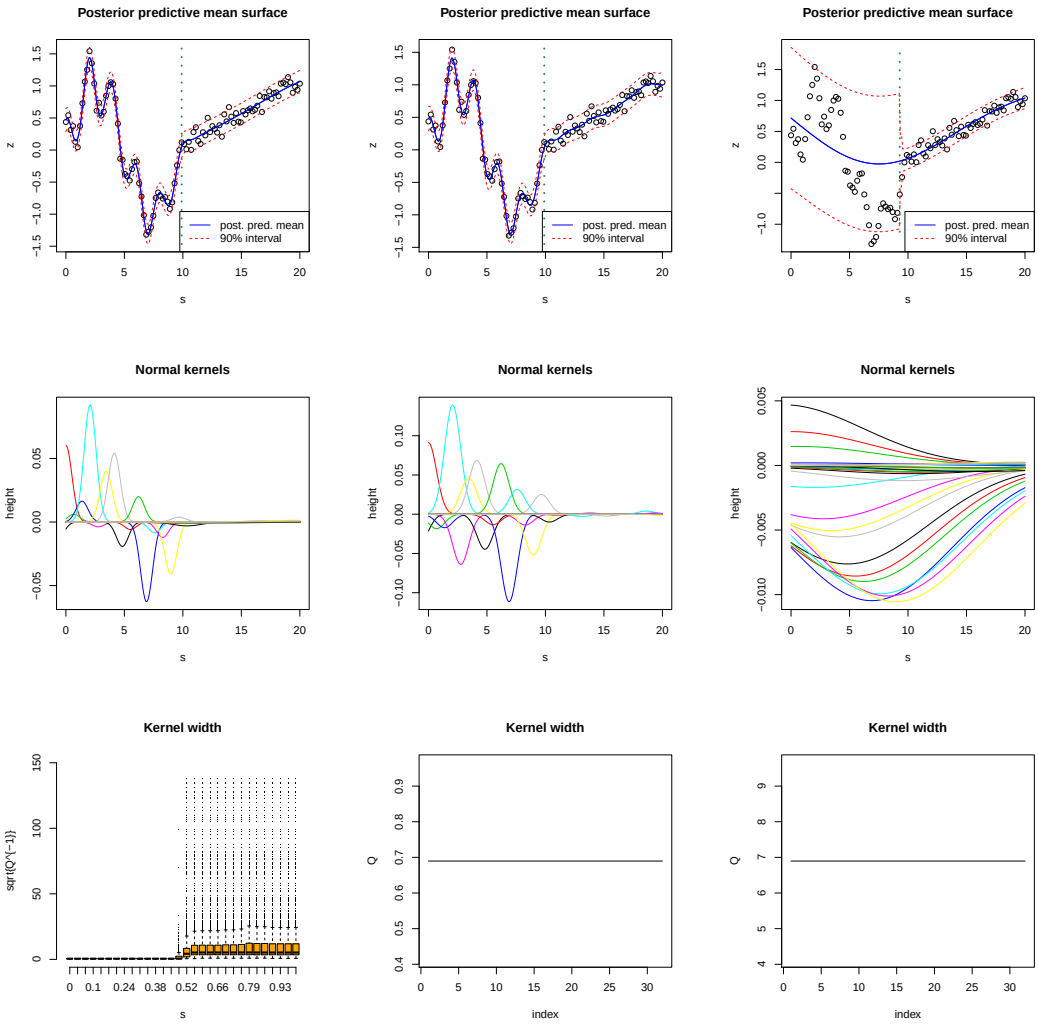


Figure 4.2: Posterior predictive summary of TPCGP with variable kernels (*left*), and with a fixed kernel such that $\sqrt{\mathbf{Q}^{-1}}$ equals to 1 (*middle*) and 10 (*right*) times the basis spacing

This response surface is shown in the top panels of Figure 4.3. The peak and trough of the surface are stretched towards the upper left direction, which is an example of an anisotropic response surface. A set of 500 data points $y(\mathbf{s})$ is generated by sampling from the response surface and adding $N(0, 0.1^2)$ noise to the samples,

$$y(\mathbf{s}) = z(\mathbf{s}) + \epsilon, \quad \epsilon \sim N(0, 0.1^2). \tag{4.6}$$

A grid of $20 \times 20$ bases is fixed over the spatial domain as shown in the top right panel of Figure 4.3. The same tree prior from the previous section is used along with the following conjugate priors,

$$\phi_\nu | b_y, \boldsymbol{\rho}, \mathcal{T} \sim G(a_y = 1,\ b_y),$$

$$\mathbf{x}_\nu | \lambda_\nu, \boldsymbol{\rho}, \mathcal{T} \sim N_{m_\nu}(\mathbf{0},\ (\lambda_\nu \mathbf{I}_{m_\nu})^{-1}), \quad \lambda_\nu | b_x, \boldsymbol{\rho}, \mathcal{T} \sim G(a_x = 1,\ b_x),$$

$$b_y \sim G(\tau_y = 1,\ \xi_y = 1000), \quad b_x \sim G(\tau_x = 1,\ \xi_x = 1000).$$

Note that the linear term is omitted since the data has mean zero. The extended TPCGP is applied to this dataset using a Bézier kernel with $\kappa = 3$ and a separate kernel precision matrix $\mathbf{Q}_\nu$ for each partition. The prior for $\mathbf{Q}_\nu$ is specified as

$$\mathbf{Q}_\nu \sim W\Big((\psi \mathbf{H})^{-1} = \frac{1}{3}\Big(\frac{3 \times 9}{20 - 1}\Big)^{-2} \mathbf{I}_2,\ \psi = 3\Big),$$

where $\frac{9}{20-1}$ is the spacing between any two adjacent basis points, and $\psi = 3$ makes this prior fairly non-informative ($\psi$ has to be at least the dimension of the spatial domain). A Wishart distribution is used as the proposal for $\mathbf{Q}_\nu$ with a mean equal to the previous RJ-MCMC sample and the degrees of freedom equals to 100. The resulting posterior predictive mean surface and the corresponding 90% interval width are shown in the

middle and bottom right panels of Figure 4.3, respectively. The mean surface closely resembles the true response, and the partition boundaries (*white dashed lines*) of the MAP treed model occur near where the anisotropy changes. The ellipses depict the orientation of the kernel support. They have been decreased in size for better visual illustration, and the actual size is 7 times larger. The orientation of these ellipses illustrates how the full TPCGP captures anisotropy by rotating/streching the kernels in the lower left region while keeping other regions fairly isotropic. The estimated mean of the measurement error standard deviation is shown in the bottom left of Figure 4.3. Although the values are different across partitions, they are close to the true value of 0.1. The log conditional of the treed models, $\log\{P(\mathcal{T}, \boldsymbol{\rho}|\cdots)\}$, and the number of partitions visited are shown in the top and middle panels of Figure 4.4. Majority of the visited models have 3 partitions, while some of them have 4 or 5. The acceptance rate is about 4.38%. Although the mixing of the treed models does not seem to be very good, the resulting model performance turns out to be fairly promising. The fitted residuals given in the bottom panel of Figure 4.4 do not show any pattern, which is a good indication that features in the data have been well explained by the full TPCGP model.

### 4.2.3 2-d Real Precipitation Data

Let us revisit the precipitation dataset from the previous chapter. The model setup and computing resource are the same as before except that a separate kernel precision matrix $\mathbf{Q}_\nu$ is assumed for each partition, and the proposal for $\mathbf{Q}_\nu$ follows the

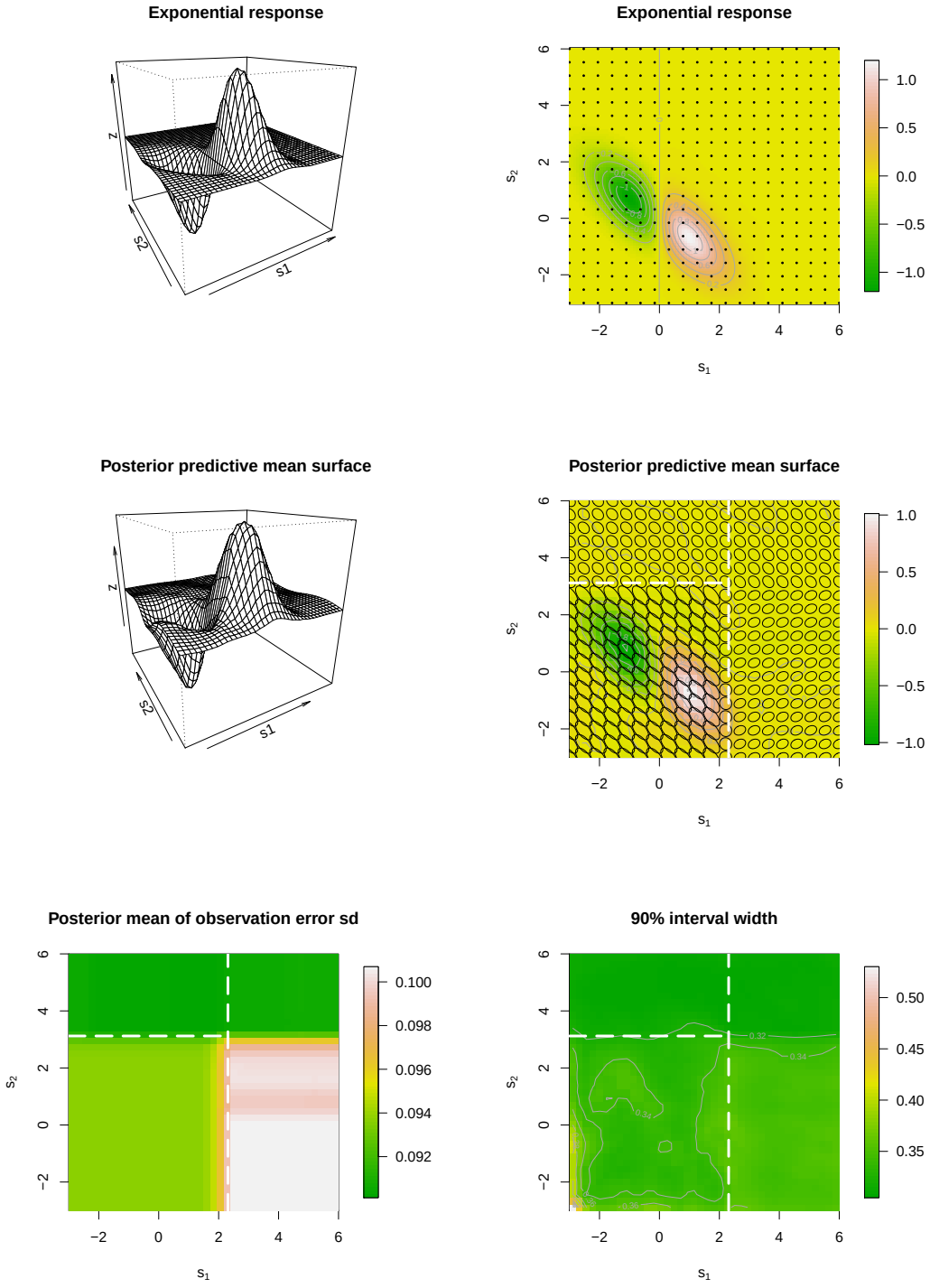


Figure 4.3: 2-d synthetic exponential response (*top*), posterior predictive mean surface (*middle*) and 90% interval width (*bottom right*), and posterior mean of observation error SD (*bottom left*)

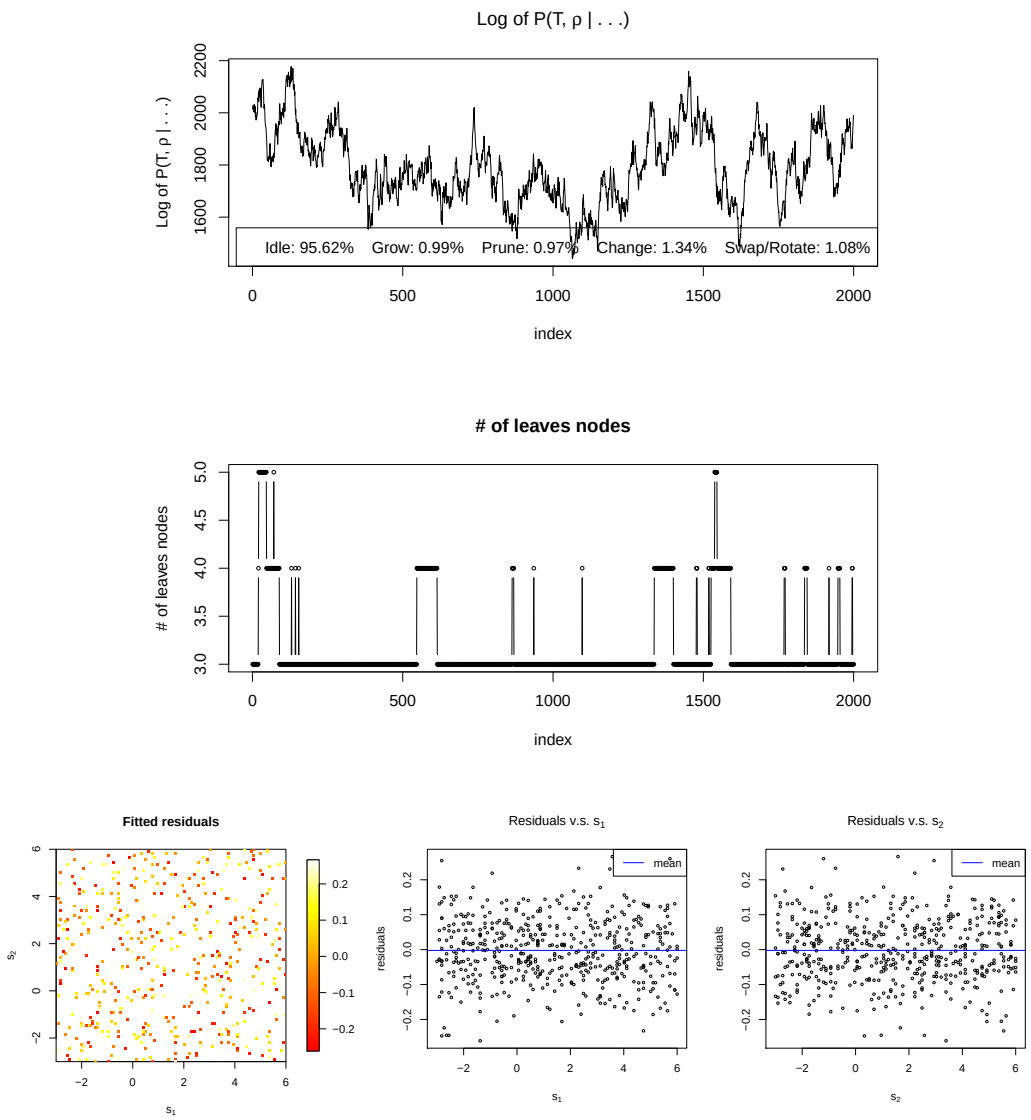


Figure 4.4: Log conditional of treed models (*top*, number of partitions visited *middle*), and fitted residuals (*bottom*).

method described by Lemos and Sansó (2009). Specifically, $\mathbf{Q}_\nu$ is represented as

$$\begin{aligned}
\mathbf{Q}_\nu &= \begin{pmatrix} \Psi_1 + \Psi_2\cos(2\omega_3) & \Psi_2\sin(2\omega_3) \\ \Psi_2\sin(2\omega_3) & \Psi_1 - \Psi_2\cos(2\omega_3) \end{pmatrix}, \\
\boldsymbol{\Psi} &= \frac{1}{2}\left(\frac{1}{\omega_1^2} + \frac{1}{\omega_2^2}, \frac{1}{\omega_1^2} - \frac{1}{\omega_2^2}\right),
\end{aligned}$$

where $\omega_1 > \omega_2 > 0$ and $\omega_2 > 0$ are the semi-major and semi-minior axes, respectively, and $-\pi/2 < \omega_3 < \pi/2$ denotes the angle between $\omega_2$ and the x-axis. A Gaussian random walk is used as a proposal for each of $\omega_1$, $\omega_2$, and $\omega_3$. Proposals that do not satisfy the above constraints are automatically rejected.

The posterior predictive mean surface and the corresponding 90% interval width are shown in Figure 4.5. These surfaces are obtained by averaging the RJ-MCMC samples from 2 independent runs of TPCGP with variable kernrels. It is found that averaging 2 independent runs is sufficient to obtain promising results. Visually, one can hardly see the difference in the posterior predictive mean surface from that of the previous chapter where the kernel is fixed, except that there are more partitions in the MAP treed model under variable kernels. In fact, a few more partitions are being visited as shown in Figure 4.8. The mechanics behind this is that the partitions tend to split the observations into subsets that are more homogeneous, and the kernels in each partition tend to capture the local structure. Since kernels are allowed to vary, they can pick up more signal from the data, thus may cause more partitions to be created. The ellipses superimposed on the posterior predictive mean surface illustrate the variation of kernels across partitions. They represent the support of 300 randomly selected kernels from a

total of $(3,646 \times 2,000)$ samples. The average acceptance rate for the kernel precision matrices is about 20%. As a result, more information are learned from the data which induces better predictive performance. The fitted residuals given in Figure 4.9 shows no significant visual correlation. The computed p-value from Moran's I test is about 0.52, which suggests that there is not enough evidence to reject the null hypothesis that the fitted residuals are uncorrelated. The MSE resulted from prediction at the $1,918$ validation locations is about 0.128, which is smaller than that of 0.138 under a fixed kernel. Posterior results of the background points $\mathbf{x}$ and linear term $\mathbf{F}\boldsymbol{\beta}$ are given in Figure 4.6 and 4.7, which show no significant difference from the model under a fixed kernel.

Three computational efficient models in the literature are considered for model comparison with TPCGP. The first is *Covariance Tapering* (CT) by Furrer et al. (2006), which is essentially a kriging method with compactly supported correlation function, and its computational advantage comes from having a sparse covariance matrix. The second is *Multivariate Adaptive Regression Splines* (MARS) by Friedman (1991), which is a nonparametric method that uses a set of bases (hinge functions) to model the process of interest. The third is the *Predictive Process model* (PP) by Banerjee et al. (2008), where the process of interest is modeled by a GP whose covariance structure is a transformation of that of a discretized stationary standard GP. The model fitting, prediction, and residuals for these three models are given in Figures 4.11, 4.12, 4.13, 4.14, 4.15, 4.16, 4.17, 4.18, and 4.19. A summary of the results is given in table 4.1. All computations are done or approximated on a desktop computer with Intel Core i5 CPU at 2.8 GHz.

Considering the case of CT, although it provides very good model fitting (the fitted residuals are almost zero), it has a prediction MSE of 0.147, whereas the full TPCGP with variable kernels has a better prediction MSE of 0.128. In terms of computational speed, CT takes about 1 miniute, which is faster than TPCGP with variable kernels since it completes in about 24 hours under the current setup. However, CT provides only a point estimate, whereas TPCGP accounts for uncertainty in the model and parameters with posterior distributions. In situations where the uncertainty of the model and parameters are desired, TPCGP would be preferred over CT. In addition, the current implementation of TPCGP is in R, which is known to be not an ideal language for implementing fast computer programs. On the other hand, despite that the computational time of MARS is also about 1 minute, it is not able to pick up much signal from the data. This can be seen from the fitted residuals, which have noticeable patterns. The computed p-value from the Moran's I test on the fitted residuals is nearly 0, and its prediction MSE is 0.472, which is much higher than that of TPCGP. Moreover, MARS also has the limitation of providing only a point estimate. Lastly, PP is the only competing model here that provides distributional estimate of model parameters. The employed function is called *spLM* available in the *spBayes* R package. The number of MCMC iterations is set to be $4,000$ such that the first $2,000$ are disgarded as burn-in while the last $2,000$ are used to form the posterior distributions. In the first attempt, the same set of $3,646$ basis points (with 50 km basis spacing) is used. The program was not able to complete within two weeks despite the fact that it is written in C++. After restarting the program and reducing the size of the basis grid to 979

(with 100 km spacing), it still takes almost a month to complete, and results in a Moran's I test p-value of nearly 0 for the fitted residuals, and a prediction MSE of 0.229. One may argue that using a larger basis grid, such as the same one ($3,646$ bases) used by TPCGP, can allow PP to generate better results. But the significant increase in the computational time (imaginable by considering the time required by using the smaller 979 basis grid) prevents PP from being a computational efficient model for large datasets. In contrast, TPCGP is able to complete in about 24 hours under the current setup and provides better prediction performance than all competing models considered here. Moreover, in situations where the process of interest has region-specific anisotropy, prior knowledge of the specific regions is required by PP, whereas TPCGP can figure out the regions automatically by partitioning. Lastly, TPCGP is able to handle heteroskedaticity by having a separate measurement error term for each partition, whereas none of the competing models here provide such funtionality.

## 4.3 Conclusion

This chapter extends the basic setup of TPCGP given in Chapter 3 by allowing a separate kernel precision matrix $\mathbf{Q}_\nu$ for each partition. Estimation of these kernel precision matrices is carried out via the Metropolis-Hastings algorithm. This extension improves both the model fitting and prediction performance by having variable kernels to better capture information embedded in the data. Model comparisons with CT, MARS, and PP shows that the full TPCGP has the best prediction performace and

provides full account of uncertainty about the model and parameters. This advantage comes at the cost of a slower (but acceptable) speed than CT and MARS, which are not MCMC based methods. A more fair comparison with PP shows that computation for the full TPCGP is significantly faster for problems considered in this dissertation.

Table 4.1: Model comparisons between TPCGP, CT, MARS, and PP

| Model | p-value from Moran'I test on fitted residuals | Mean squared prediction residuals | Execution time |
|---|---|---|---|
| TPCGP (variable kernels) | 0.52 | 0.128 | $\approx$ 24 hours |
| TPCGP (fixed kernel) | 0.09 | 0.138 | $\approx$ 3 hours |
| CT | 0.67 | 0.147 | $\approx$ 1 minute |
| MARS | $< 10^{-9}$ | 0.472 | $\approx$ 1 minute |
| PP (basis spacing = 100 km) | $< 10^{-9}$ | 0.229 | $\approx$ 30 days |

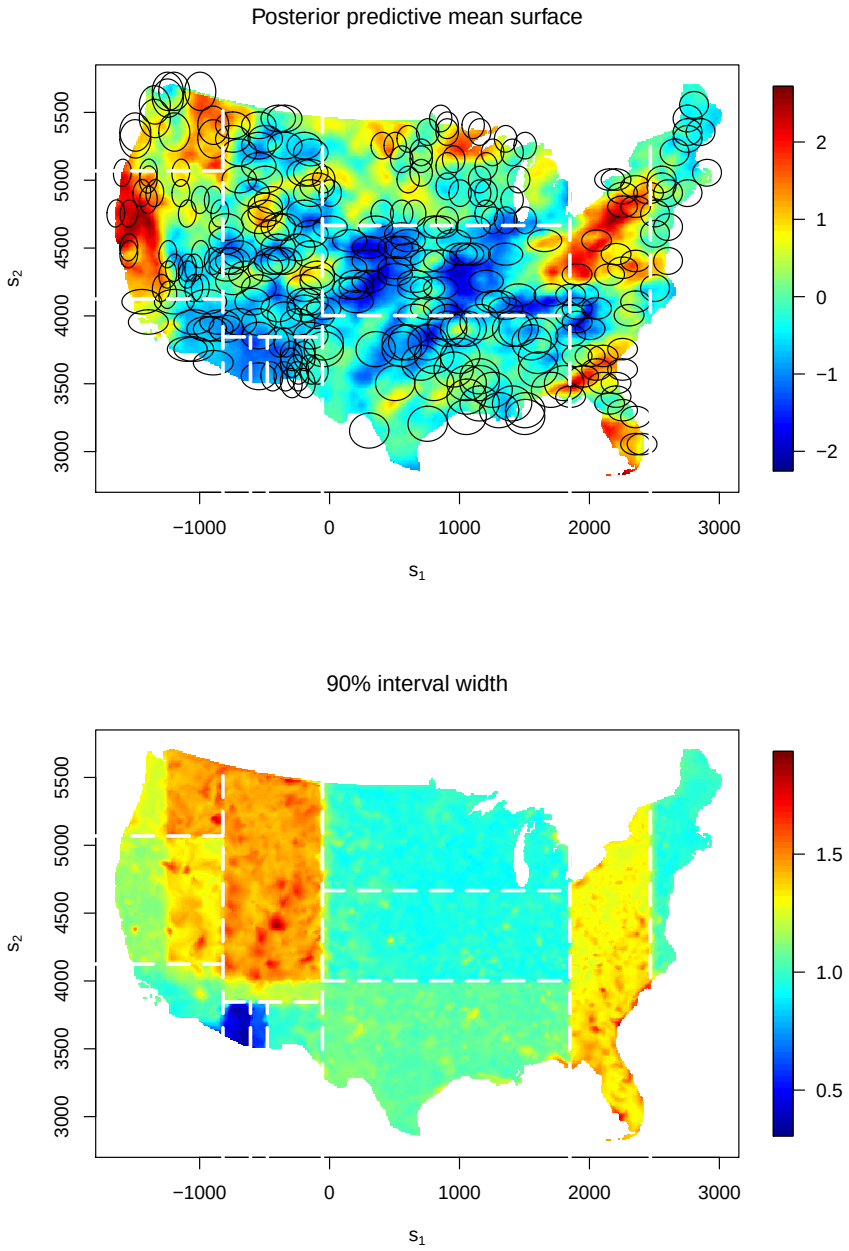


Figure 4.5: Posterior predictive mean surface (*top*) and 90% posterior predictive interval width (*bottom*)

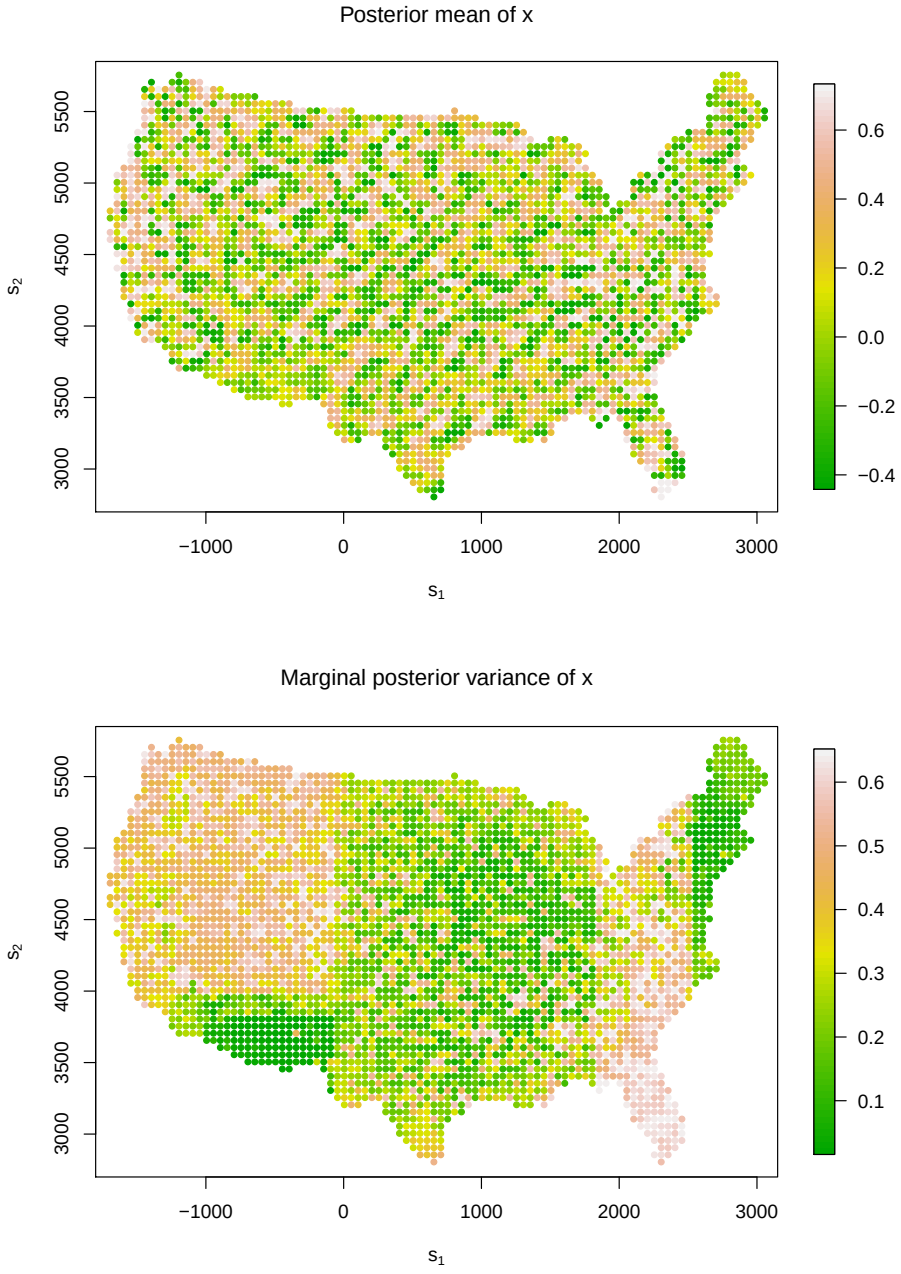


Figure 4.6: Posterior mean (*top*) and variance (*bottom*) of $\mathbf{x}$

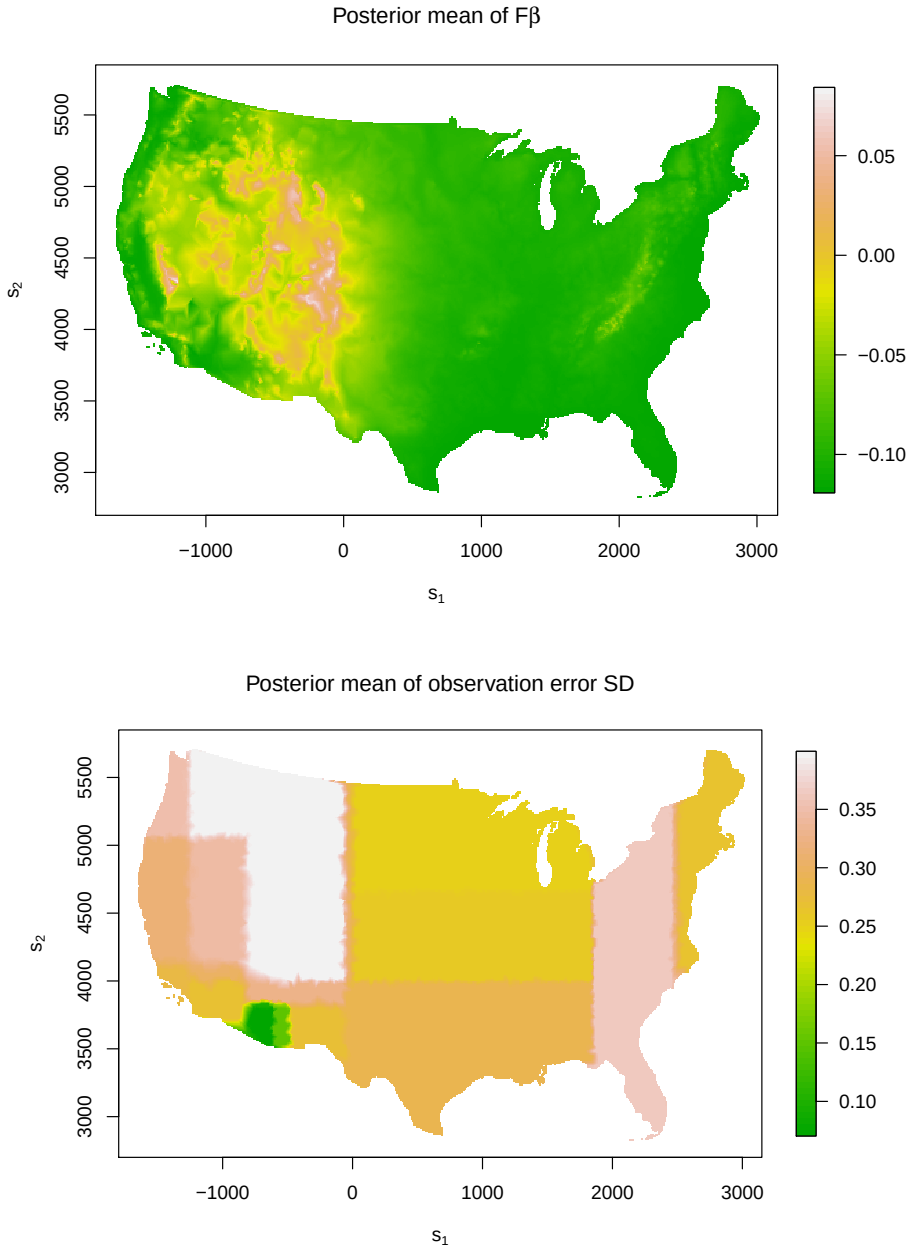


Figure 4.7: Posterior mean of $\mathbf{F}\boldsymbol{\beta}$ (*top*) and observation error standard deviation $\sqrt{1/\phi}$ (*bottom*)

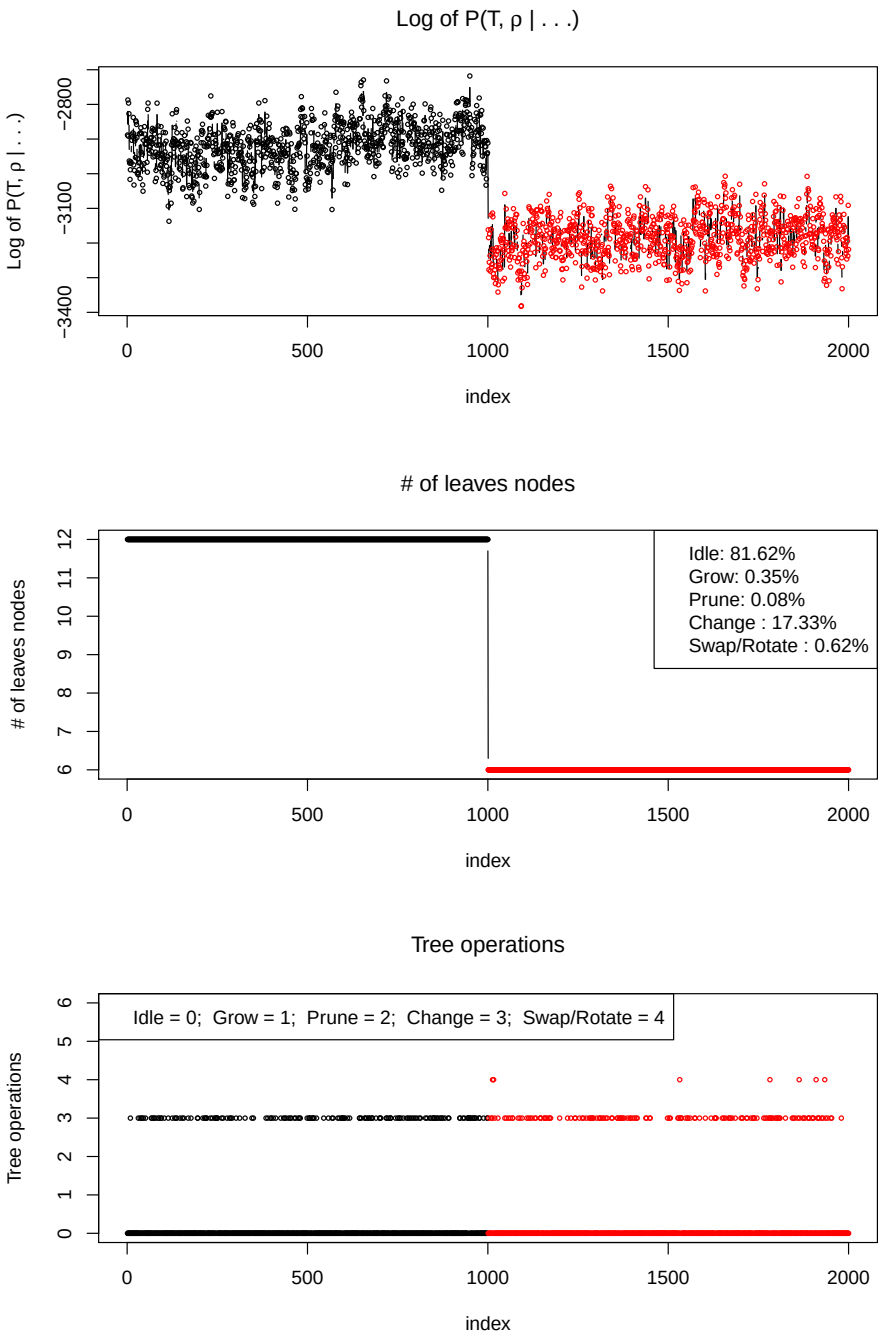


Figure 4.8: Log conditional of treed models (*top*), number of partitions visited (*middle*), and the accepted tree operations (*bottom*)

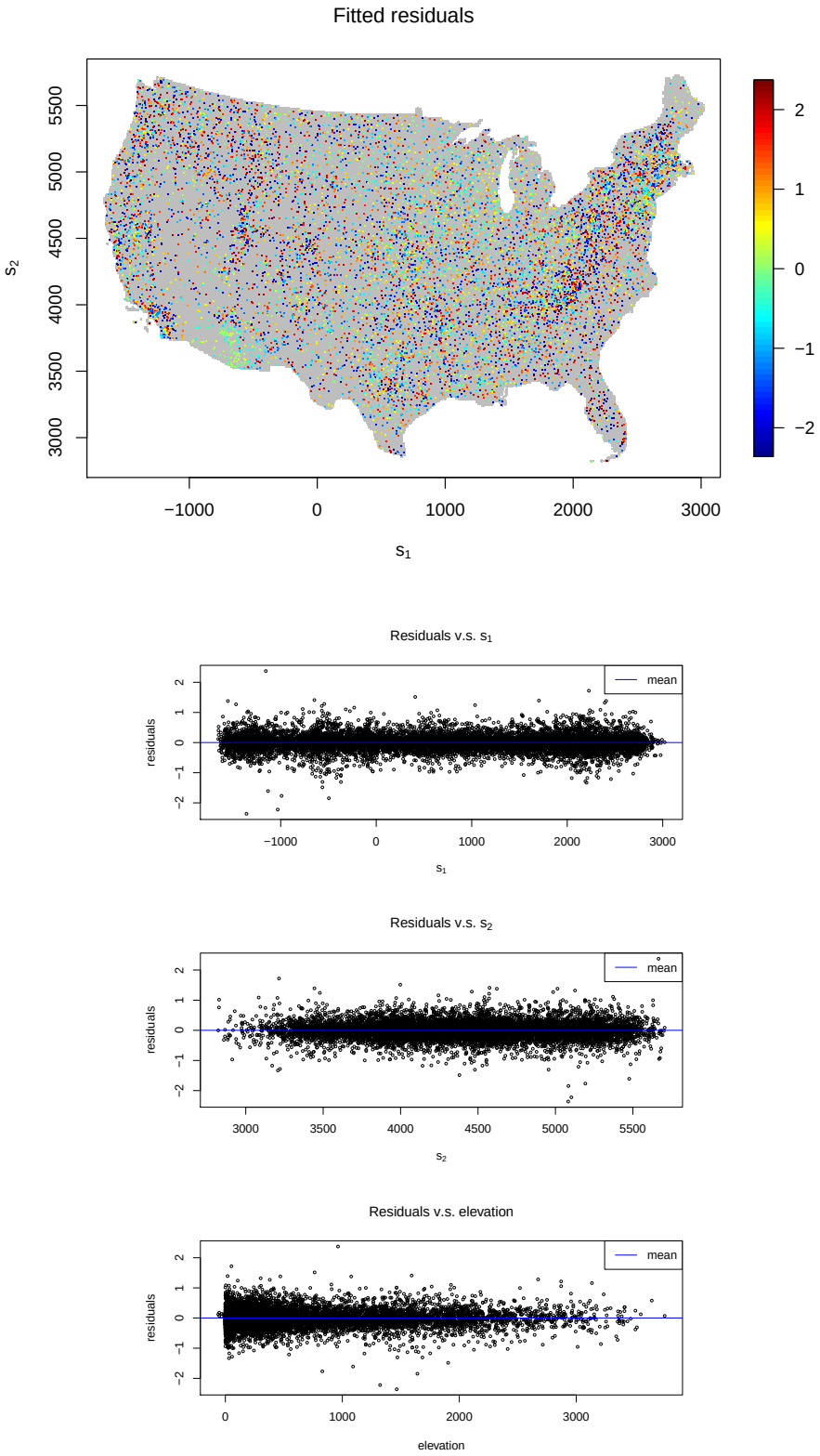


Figure 4.9: Fitted residuals from TPCGP with variable kernels

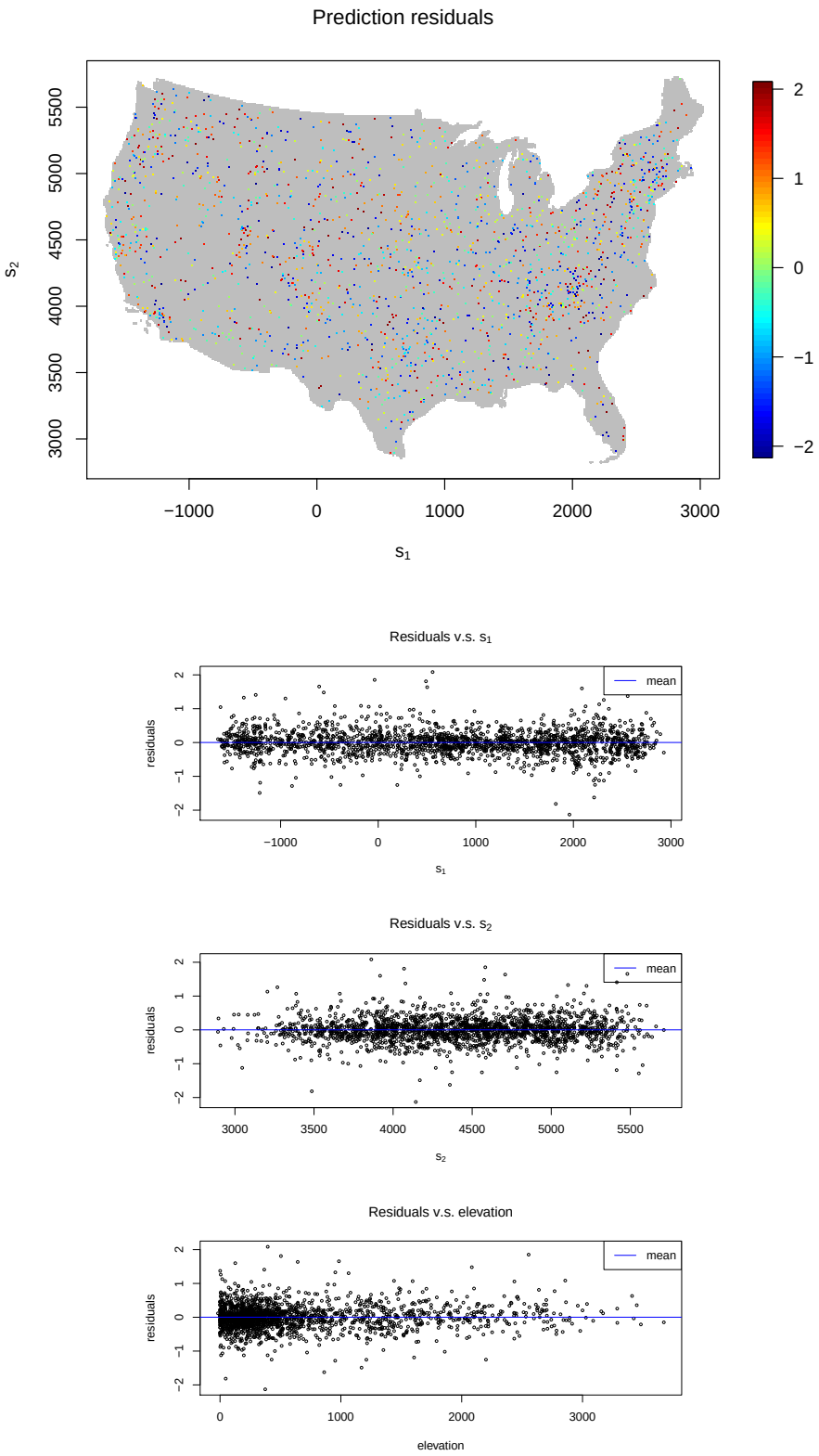


Figure 4.10: Prediction residuals from TPCGP with variable kernels

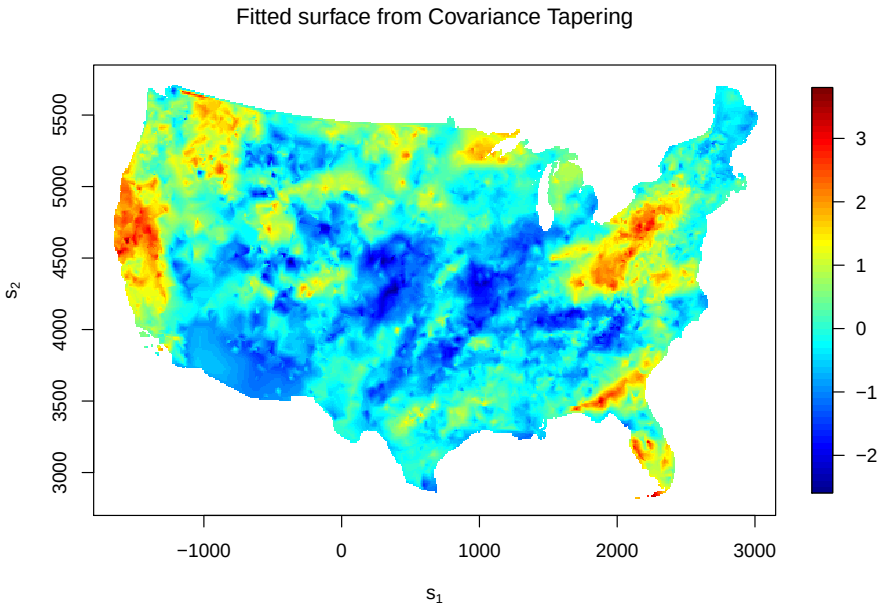


Figure 4.11: Fitted surface from Covariance Tapering

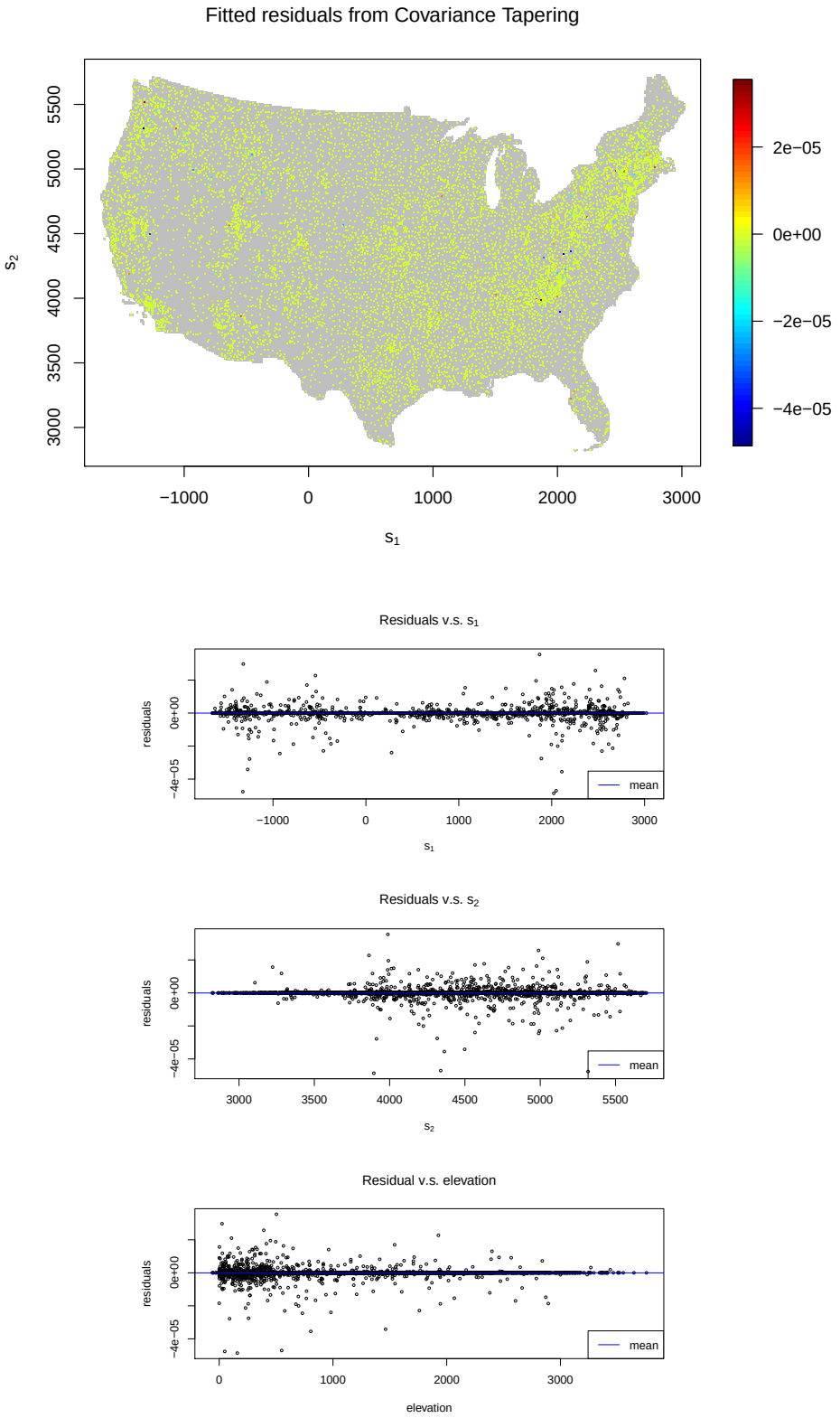


Figure 4.12: Fitted residuals from Covariance Tapering

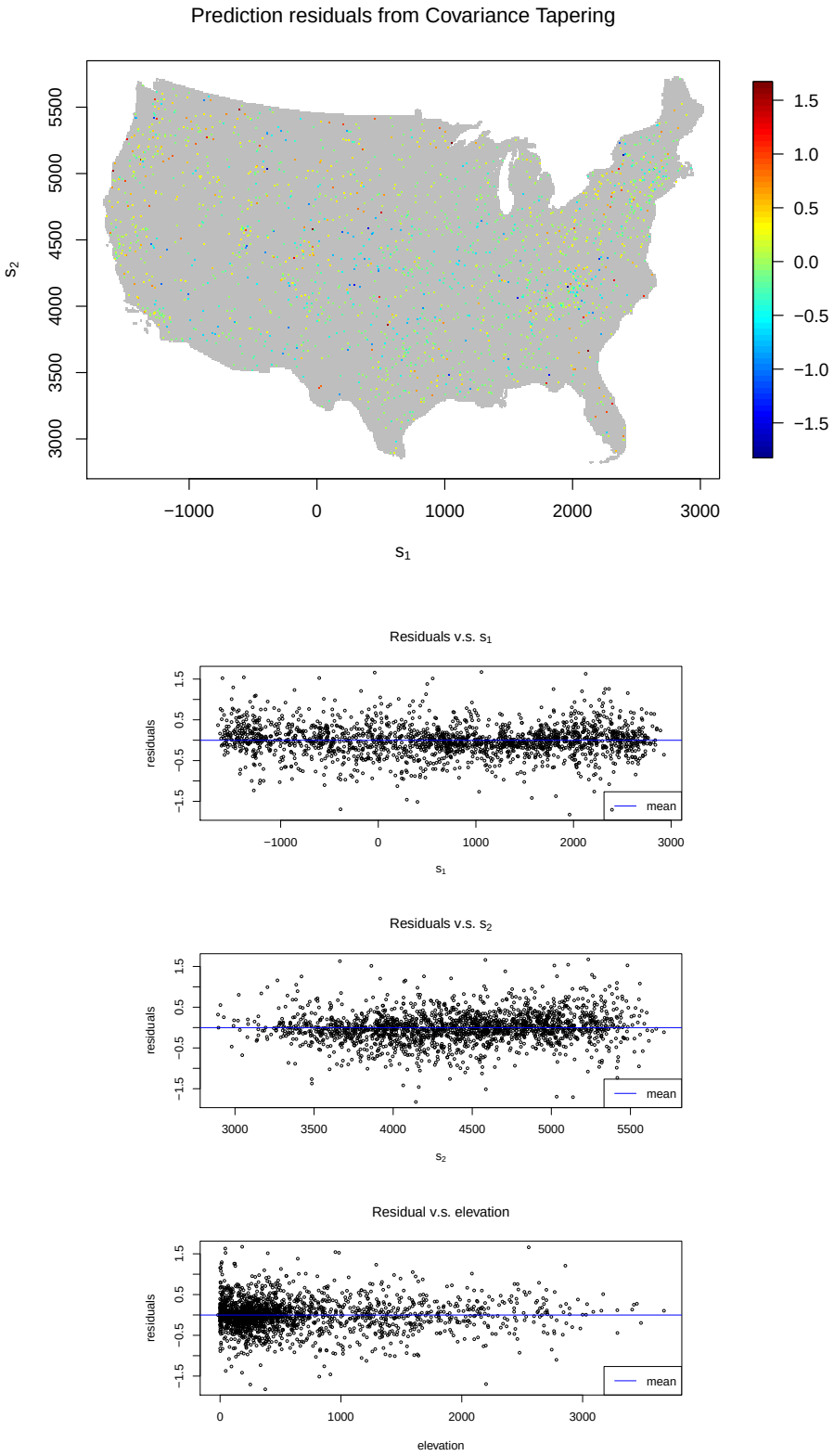


Figure 4.13: Prediction residuals from Covariance Tapering

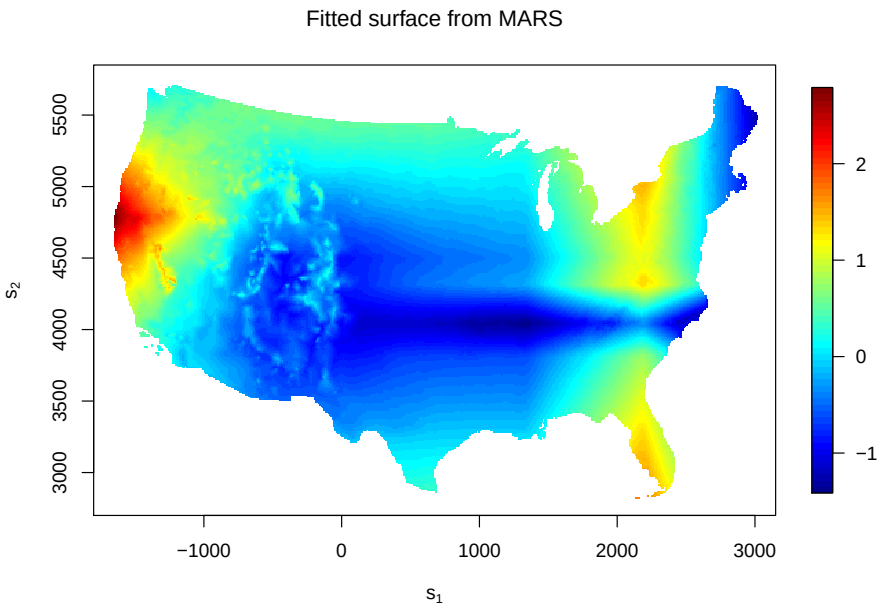


Figure 4.14: Fitted surface from Multivariate Adaptive Regression Splines

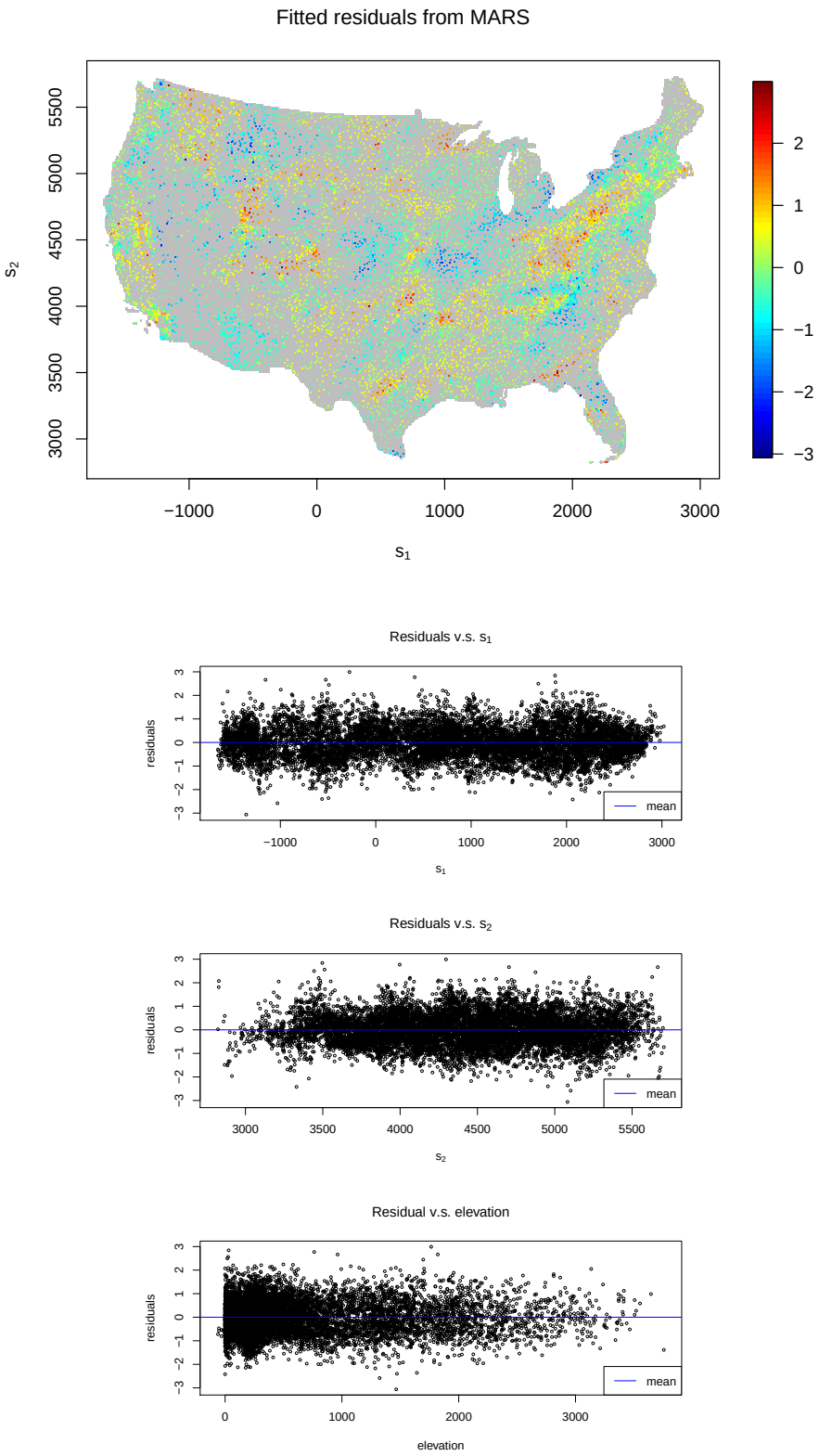


Figure 4.15: Fitted residuals from Multivariate Adaptive Regression Splines

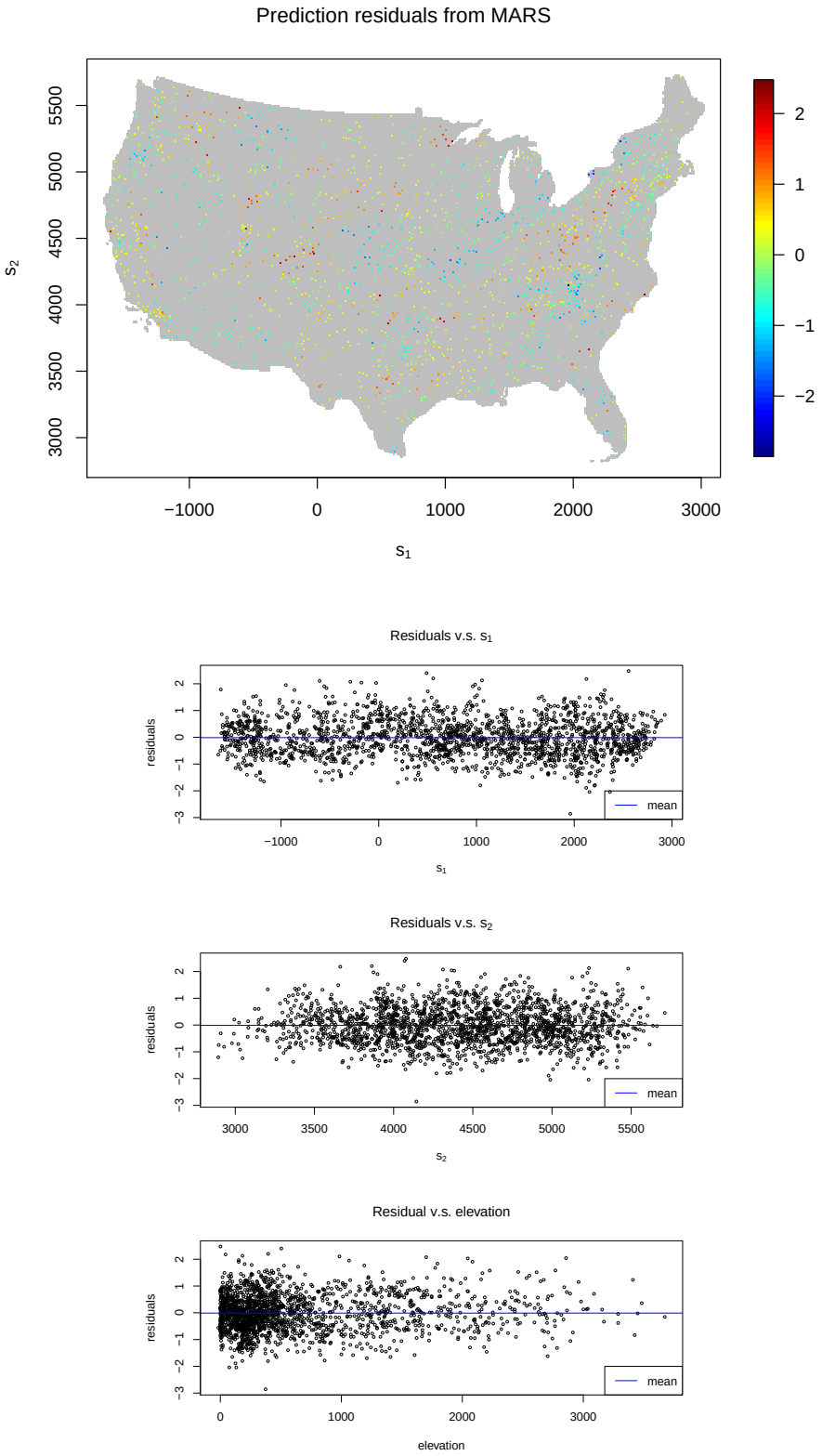


Figure 4.16: Prediction residuals from Multivariate Adaptive Regression Splines

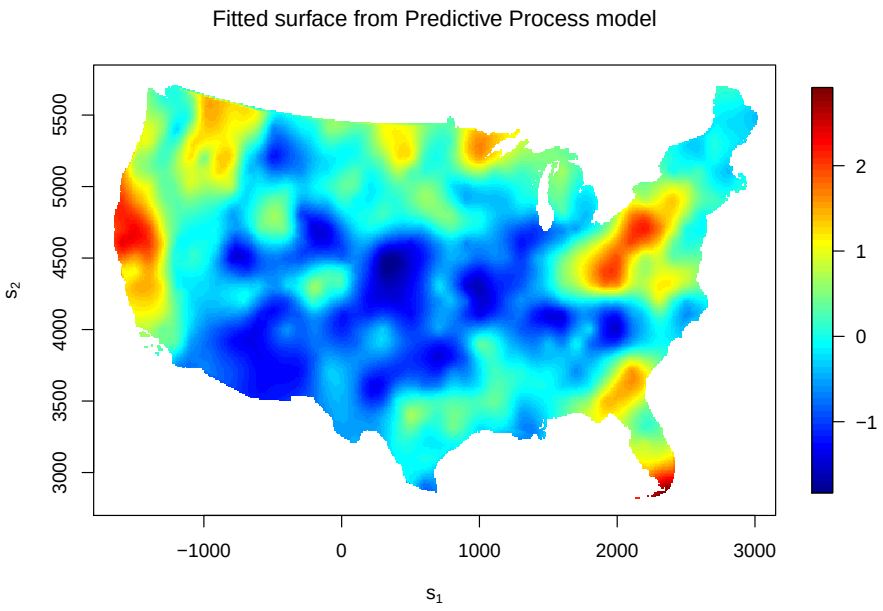


Figure 4.17: Fitted surface from the Predictive Process model

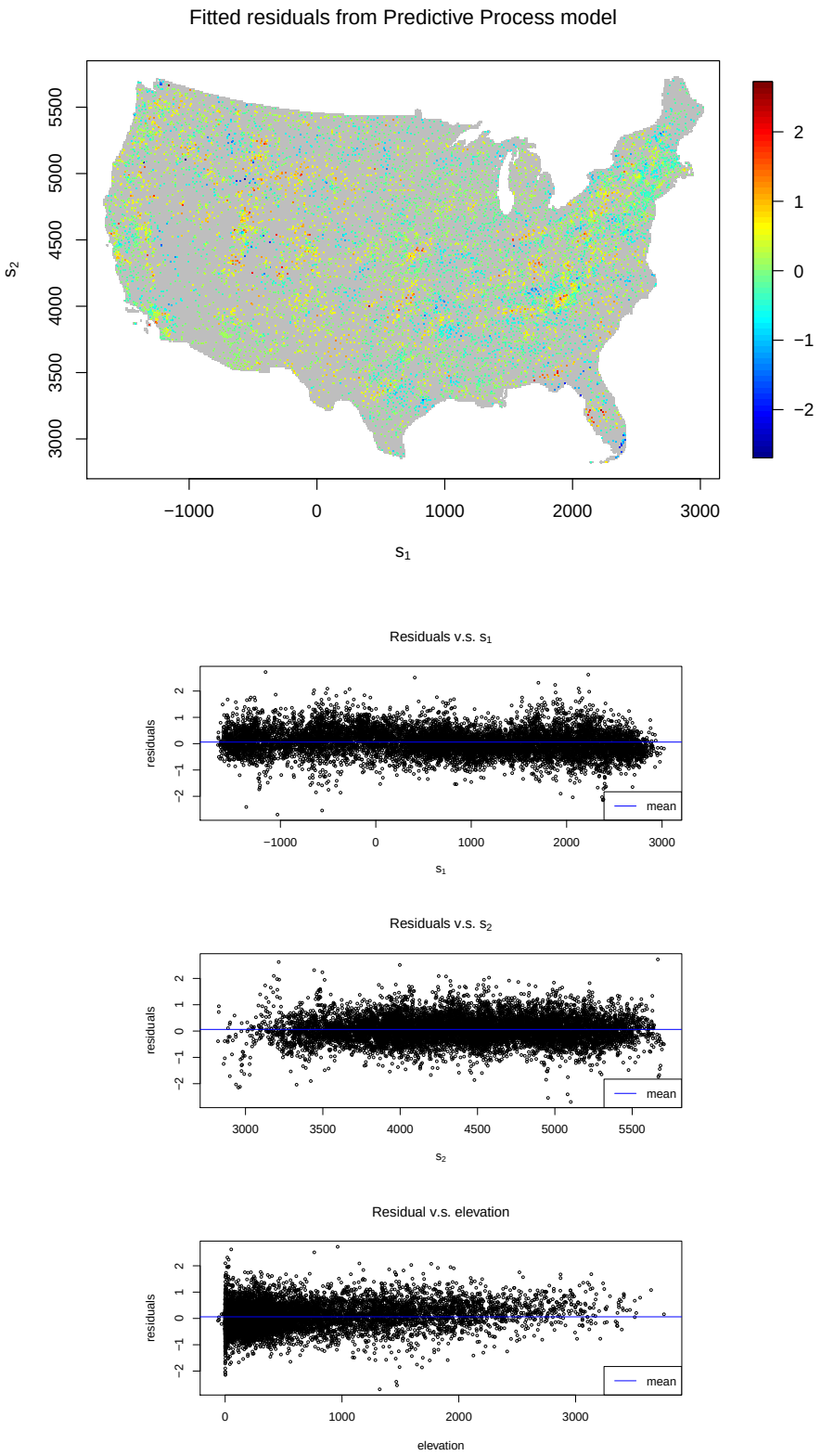


Figure 4.18: Fitted residuals from the Predictive Process model

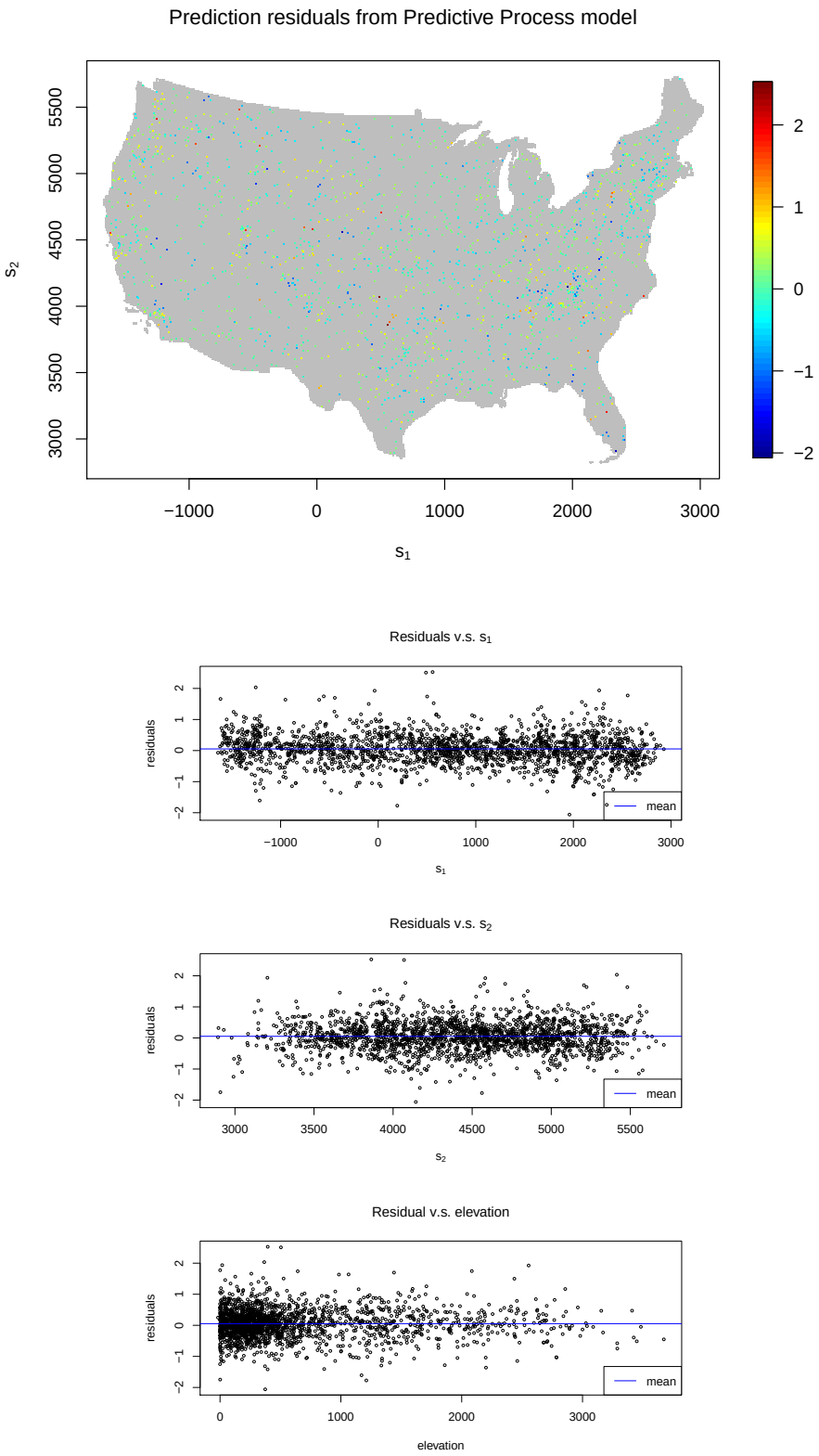


Figure 4.19: Prediction residuals from the Predictive Process model

# Chapter 5

# Sequential Process Convolution GP Models

## 5.1 Introduction

Gaussian processes (GP) have been widely used to model the underlying process of interest in regression and classification models (Neal, 1997, 1998; Rasmussen and Williams, 2006). Some of the major applications include computer experiments (Sacks et al., 1989), and models of spatial and spatio-temporal data (Cressie, 1991; Banerjee et al., 2003). Recent developments in GP models focus on using the Bayesian approach for inference because it provides for full accounting of uncertainty. In this approach, a Gaussian process with a chosen correlation function is specified as the prior for the underlying process of interest. Combining the prior distribution with the likelihood using Bayes' rule forms the posterior distribution which can be sampled using Markov Chain

Monte Carlo (MCMC). One drawback of these standard GP models is that they require a matrix decomposition whose complexity increases at a rate of the cube of the sample size, which makes them impractical for applications with moderately large datasets.

As shown in the previous chapters, an alternative approach which can help alleviate the problem of large sample size is the process convolution approach to constructing a GP (Higdon, 1998, 2002; Calder et al., 2002; Paciorek and Schervish, 2006). This approach generates a GP by convolving a white noise process with a smoothing kernel. Bayesian inference of the model parameters again proceeds using MCMC. Although this approach is computationally efficient for applications with a large sample size, the batch nature of MCMC makes it unsuitable for sequential problems. For example, design points in computer simulation experiments are naturally generated sequentially so that MCMC has to be repeated for each new data arrival, which renders the whole inference process computationally demanding. In this chapter, sequential inference for the process convolution GP model is introduced based on a Sequential Monte Carlo (SMC) method called *Particle Learning* (Carvalho et al., 2010). A similar approach has been developed by Gramacy and Polson (2009) where a standard GP is considered. The updating time of their model is on the order of $Q(t^2)$, where $t$ denotes the time index (or the sample size at time $t$ assuming that one data point arrives at a time). Although the computing time is a significant improvement from $Q(t^3)$ in the setting of MCMC inference, $t$ is not a fixed constant and still has the potential problem of being too large for the model to be computationally efficient. In contrast, the process convolution approach has running time on the order of $Q(m^3)$, where $m$ denotes the

number of background points that is always fixed and usually on the order of hundreds so that the model remains computationally efficient. This advantage would be more obvious as the sample size becomes even moderately large, and this may occur when a batch of data points is considered at each time step. Model results show that the sequential process convolution GP provides comparable model fitting to the standard GP approach at a faster speed.

## 5.2 Sequential Monte Carlo and Particle Learning

In the state space models literature, there are two main statistical inference problems: 1) *sequential state filtering and parameter learning* which is characterized by the joint posterior distribution of states and parameters at each point in time, and 2) *state smoothing* which is characterized by the distribution of the states conditional on all data and marginalizing out all unknown parameters. That is, filtering and learning is about inferring the hidden states and parameters given only the currently available data at each time point, and smoothing is about inferring the states based on the full dataset. In the setting of linear Gaussian models, the Kalman filter (Kalman, 1960) provides analytical recursion equations for both filtering and smoothing assuming knowledge of parameters. For example, in the case of filtering, updating of the model at time $t+1$ is done by treating the model fitting at time $t$ as a prior, which is then combined (using Bayes' rule) with the likelihood of new data arriving at time $t+1$. If the model has unknown static (independent of $t$) parameters, the full sequence of updating equations

with all data up to the current time defines a likelihood which can be combined with a prior for Bayesian inference (West and Harrison, 1997). Depending on the prior, the resulting inferential complexity can go from being analytically tractable to intractable. For more general model specifications, it is common to apply Sequential Monte Carlo (SMC) methods which are also known as particle filters. SMC provides a numerical alternative to the inference problem of non-linear and/or non-Gaussian dynamical process, or when the parameters and their priors do not lead to tractable posteriors. In this chapter, the emphasis is on the filtering/parameter learning problem. SMC uses a set of particles $\{Z_t^{(i)}\}_{i=1}^N$ to approximate the posterior distribution of the state information $Z_t$ about the dynamic process, conditional on the data up to time $t$. The main task is to update the particle approximation from time $t$ to $t+1$. Pure filtering of the state information (assuming knowledge of parameters) can be done using the *bootstrap filter* of Gordon et al. (1993) which upon arrival of new data $\mathbf{y}_{t+1}$ *propagates* the particles via the state evolution equation $P(Z_{t+1}|Z_t)$, then *resamples* the propagated particles with weights proportional to the likelihood $P(\mathbf{y}_{t+1}|Z_{t+1})$. Another method for the same problem would be the Auxiliary Particle Filter (APF) of Pitt and Shephard (1999) which is based on a *resample-propagate* approach. Filtering with learning of unknown static parameters can be done using the filter of Liu and West (2001) which extends the APF by using a kernel approximation to the posterior of the parameters, or the filter of Storvik (2002) which assumes that the posterior of the parameters depends on a low-dimensional set of sufficient statistics that can be recursively updated. Although filtering algorithms can be used for learning of parameters, they are not efficient without

modifications. A new class of SMC algorithms called particle learning (PL) (Carvalho et al., 2010) focuses on the parameter learning part and hence is more suitable for the sequential learning of static models (as opposed to dynamic models). PL is based on a resample-propagate approach as follows

$$\textbf{Resampling:} \quad P(Z_t|\mathbf{y}^{t+1}) \propto P(\mathbf{y}_{t+1}|Z_t)P(Z_t|\mathbf{y}^t),$$

$$\textbf{Propogation:} \quad P(Z_{t+1}|\mathbf{y}^{t+1}) = \int P(Z_{t+1}|Z_t, \mathbf{y}_{t+1})dP(Z_t|\mathbf{y}^{t+1}),$$

where $Z_t$ denotes a particle that contains the *sufficient information* and $\mathbf{y}_t$ denotes the observation vector at time $t$. Sufficient information at time $t$ may include the hidden states, sufficient statistics of the parameters, or even the parameters themselves. The above algorithm starts by resampling the sufficient information $Z_t$ with probability weights proportional to the predictive distribution of the new data $P(\mathbf{y}_{t+1}|Z_t)$. Then, the new set of sufficient information is propagated based on a state transition distribution $P(Z_{t+1}|Z_t, \mathbf{y}_{t+1})$. Let $\{Z_t^{(i)}\}_{i=1}^N$ denote the set of particles that approximates $P(Z_t|\mathbf{y}^t)$, the actual implementation procedure of particle learning can be summarized as follows:

1. Sample indices $\{\zeta(j) : j = 1, \cdots, N\}$ with replacement from a Multinomial distribution with weights proportional to the predictive distribution, i.e., $P(\zeta(j) = i) \propto P(\mathbf{y}_{t+1} \mid Z_t^{(i)})$ for $i = 1, \cdots, N$. Set $\{Z_t^{(j)}\}_{j=1}^N = \{Z_t^{\zeta(j)}\}_{j=1}^N$.

2. Draw $Z_{t+1}^{(j)}$ from $P(Z_{t+1}|Z_t^{(j)}, \mathbf{y}_{t+1})$ to obtain a new particle set $\{Z_{t+1}^{(j)}\}_{j=1}^N$ which approximates $P(Z_{t+1}|\mathbf{y}^{t+1})$.

The following section shows how to perform on-line inference for a process convolution GP model based on the particle learning procedure.

## 5.3 Particle Learning for Process Convolution GP

The default construction of the process convolution GP model is static in the sense that there is no time component involved. To allow sequential inference, the variable $t$ is used to denote the sequential ordering of the data and sufficient information. The data are assumed to be sequentially independent. As shown in the previous sections, definitions of the sufficient information $Z_t$ and predictive distribution $P(\mathbf{y}^{t+1}|Z_t)$ are required for the application of particle learning. The predictive distribution of model (1.20) can be obtained in closed form:

$$\begin{aligned} & P(\mathbf{y}_{t+1}|a_{y,t}, b_{y,t}, \boldsymbol{\beta}_t, \mathbf{C}_t, \mathbf{x}) \\ \equiv \; & \mathscr{T}_{n_{t+1}}\Big(2a_{y,t}\,,\, \mathbf{F}_{t+1}\boldsymbol{\beta}_t + \mathbf{K}_{t+1}\mathbf{x}\,,\, \frac{b_{y,t}}{a_{y,t}}\Big(\mathbf{I}_{n_t} + \mathbf{F}_{t+1}{\mathbf{C}_t}^{-1}\mathbf{F}_{t+1}^{\top}\Big)\Big), \end{aligned} \tag{5.1}$$

where $\mathscr{T}$ denotes the Multivariate Student's t distribution, and $\mathbf{y}_{t+1}$ denotes the $(n_{t+1} \times 1)$ data vector at time $t+1$. Note that all parameters except $\mathbf{x}$ have been integrated out and only the sufficient statistics $\{a_{y,t}, b_{y,t}, \boldsymbol{\beta}_t, \mathbf{C}_t\}$ are needed to compute the predictive probability given a new data point/set $\{\mathbf{y}_{t+1}, \mathbf{F}_{t+1}, \mathbf{K}_{t+1}\}$. These sufficient statistics are based on the complete conditional distributions. Therefore, the sufficient information $Z_t$ would contain $\{a_{y,t}, b_{y,t}, \boldsymbol{\beta}_t, \mathbf{C}_t\}$. Moreover, $\mathbf{x}$ is needed to evaluate the predictive density, thus it is also stored into the sufficient information $Z_t$. If one is interested in $\{\lambda,\ \phi,\ \boldsymbol{\beta}\}$, they can also be kept, namely, $Z_t = \{a_{y,t}, b_{y,t}, \boldsymbol{\beta}_t, \mathbf{C}_t, \mathbf{x}, \lambda, \phi, \boldsymbol{\beta}\}$. Assuming

that the initial priors are given by

$$\begin{aligned}
\boldsymbol{\beta}|\phi &\sim N_{p+1}(\boldsymbol{\beta}_0, (\phi\mathbf{C}_0)^{-1}), && (5.2)\\
\phi &\sim G(a_{y,0}, b_{y,0}), && (5.3)\\
\mathbf{x}|\lambda &\sim N_m(\mathbf{0}, \lambda^{-1}\mathbf{I}_m), && (5.4)\\
\lambda &\sim G(a_{x,0}, b_{x,0}), && (5.5)
\end{aligned}$$

the complete conditionals at time $t$ can be found as

$$\begin{aligned}
\boldsymbol{\beta}|\phi &\sim N_{p+1}(\boldsymbol{\beta}_t\,, (\phi\mathbf{C}_t)^{-1}), && (5.6)\\
\phi &\sim G(a_{y,t}, b_{y,t}), && (5.7)\\
\mathbf{x}|\lambda &\sim N_m(\boldsymbol{\mu}_t, \boldsymbol{\Sigma}_t), && (5.8)\\
\lambda &\sim G(a_{x,t}, b_{x,t}), && (5.9)
\end{aligned}$$

where

$$\boldsymbol{\beta}_t = \mathbf{C}_t^{-1}({\mathbf{F}_t}^\top\mathbf{v}_t + \mathbf{C}_{t-1}\boldsymbol{\beta}_{t-1}), \quad \mathbf{C}_t = ({\mathbf{F}_t}^\top\mathbf{F}_t + \mathbf{C}_{t-1}),$$

$$a_{x,t} = m/2 + a_{x,0}, \quad b_{x,t} = 0.5\mathbf{x}^\top\mathbf{x} + b_{x,0}, \quad a_{y,t} = n_t/2 + a_{y,t-1},$$

$$b_{y,t} = b_{y,t-1} + \frac{1}{2}(s_t^2 + (\boldsymbol{\beta}_{t-1} - \hat{\boldsymbol{\beta}}_t)^\top(\mathbf{C}_{t-1}^{-1} + ({\mathbf{F}_t}^\top\mathbf{F}_t)^{-1})^{-1}(\boldsymbol{\beta}_{t-1} - \hat{\boldsymbol{\beta}}_t)),$$

$$\boldsymbol{\mu}_t = (\phi{\mathbf{K}^t}^\top\mathbf{K}^t + \lambda\mathbf{I}_m)^{-1}\phi{\mathbf{K}^t}^\top(\mathbf{y}^t - \mathbf{F}^t\boldsymbol{\beta}), \quad \boldsymbol{\Sigma}_t = (\phi{\mathbf{K}^t}^\top\mathbf{K}^t + \lambda\mathbf{I}_m)^{-1},$$

$$\hat{\boldsymbol{\beta}}_t = ({\mathbf{F}_t}^\top\mathbf{F}_t)^{-1}{\mathbf{F}_t}^\top\mathbf{v}_t, \quad \mathbf{v}_t = \mathbf{y}_t - \mathbf{K}_t\mathbf{x}, \quad s_t^2 = (\mathbf{v}_t - \mathbf{F}_t\hat{\boldsymbol{\beta}}_t)^\top(\mathbf{v}_t - \mathbf{F}_t\hat{\boldsymbol{\beta}}_t),$$

$n^t$ denotes the total number of observations up to time $t$, $\mathbf{y}^t = ({\mathbf{y}^{t-1}}^\top, \mathbf{y}_t^\top)^\top$, $\mathbf{F}^t = ({\mathbf{F}^{t-1}}^\top, \mathbf{F}_t^\top)^\top$, and $\mathbf{K}^t = ({\mathbf{K}^{t-1}}^\top, \mathbf{K}_t^\top)^\top$ denote the $(n^t \times 1)$ data vector, $(n^t \times (p+1))$ design matrix, and $(n^t \times m)$ kernel matrix, respectively. In the propagate step, the

resampled sufficient information $Z_t$ is updated to account for the new data $\{\mathbf{y}_{t+1}, \mathbf{F}_{t+1}\}$. Specifically, the sufficient statistics are updated deterministically based on the above equations, but for time $t+1$. Once the sufficient statistics are updated, then the parameters are sampled from their conditionals since they are needed to compute the predictive density for the next data arrival. Following is a rough sketch of the resampling and propagation procedures:

1. Upon arrival of $\mathbf{y}_{t+1}$, sample indices $\{\zeta(j) : j = 1, \cdots, N\}$ with replacement from a Multinomial distribution with weights proportional to the predictive distribution (5.1), i.e., $P(\zeta(j) = i) \propto P(\mathbf{y}_{t+1} \,|\, Z_t^{(i)})$ for $i = 1, \cdots, N$. Set $\{Z_t^{(j)}\}_{j=1}^N = \{Z_t^{\zeta(j)}\}_{j=1}^N$.

2. For $j = 1, \cdots, N$, update the sufficient statistics to obtain $a_{x,t+1}^{(j)}, b_{x,t+1}^{(j)}, a_{y,t+1}^{(j)}, b_{y,t+1}^{(j)}$, $\boldsymbol{\beta}_{t+1}^{(j)}$, $\mathbf{C}_{t+1}^{(j)}, \boldsymbol{\mu}_{t+1}^{(j)}$, and $\boldsymbol{\Sigma}_{t+1}^{(j)}$, then sample

$$\begin{aligned}
\lambda^{(j)} &\sim G(a_{x,t+1}^{(j)}, b_{x,t+1}^{(j)}), \\
\phi^{(j)} &\sim G(a_{y,t+1}^{(j)}, b_{y,t+1}^{(j)}), \\
\boldsymbol{\beta}^{(j)} &\sim N_{p+1}(\boldsymbol{\beta}_{t+1}^{(j)}, (\phi \mathbf{C}_{t+1}^{(j)})^{-1}), \\
\mathbf{x}^{(j)} &\sim N_m(\boldsymbol{\mu}_{t+1}^{(j)}, \boldsymbol{\Sigma}_{t+1}^{(j)}).
\end{aligned}$$

The following section applies our methodology to two illustrative examples. For convenience, the sequential process convolution GP model will be abbreviated by SPCGP. Comparisons with the standard process convolution GP model (PCGP), the sequential GP with standard specification (PLGP) by Gramacy and Polson (2009), and the

standard GP with MCMC, are also given. PLGP is provided by the *plgp* R package (Gramacy, 2010), and the standard GP with MCMC is provided by the *tgp* R package (Gramacy, 2007).

## 5.4 Illustration

### 5.4.1 1-d Synthetic Sinusoidal Data

One hundred data points are generated by sampling from the 1-d response below,

$$z(s) = \sin\left(\frac{\pi s}{5}\right) + 0.2\cos\left(\frac{4\pi s}{5}\right), \quad 0 \leq s \leq 9.6, \tag{5.10}$$

and adding $N(0, sd = 0.1^2)$ noise to the sampled points (shown in Figure 5.1). One

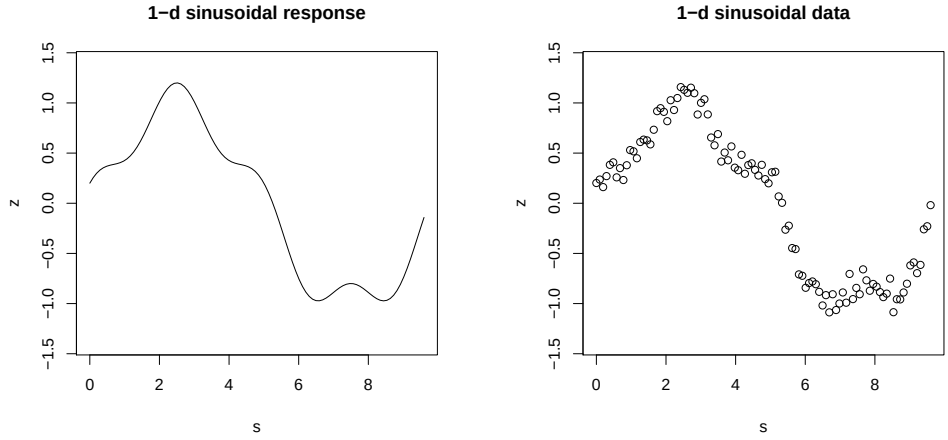


Figure 5.1: 1-d synthetic sinusoidal response (*left*) and data (*right*)

data point is randomly selected without replacement at each time step as an input to

the model. The initial priors are given by

$$\beta \sim N(0,\ (10\phi)^{-1}), \quad \phi \sim G(a_y = 1,\ 0.001),$$

$$\mathbf{x}|\lambda \sim N_m(\mathbf{0},\ (\lambda \mathbf{I}_m)^{-1}), \quad \lambda \sim G(a_x = 1,\ 0.001).$$

Note that $\beta$ is a scalar which corresponds to the intercept, and the linear component is not included. A Gaussian kernel with 30 bases is applied and the standard deviation of the kernel is set equal to the spacing between adjacent basis points. A total of 500 particles are used in the simulation. The posterior predictive mean surfaces with the corresponding 90% interval are shown in Figure 5.2 for $t = \{5, 10, 20, 30, 40, 50\}$.

When only a few data points are available, uncertainty at unobserved locations is reflected through having a larger 90% posterior predictive interval, as opposed to a relatively smaller interval at the observed locations. As more data points arrive and spread over the entire domain, the model quickly improves and is able to obtain a mean surface at $t = 50$ that has most of the features of the true response. The posterior predictive summary for $t = 100$ is shown in the top left panel of Figure 5.3, along with the corresponding 500 particles on the right. The mean surface resembles the true response, and the 90% interval well captures the data variability. Sensitivity of the model against the number of particles is illustrated in the middle and bottom panels, which are based on 100 and 20 particles, respectively. Note that SPCGP is fairly robust in the sense that even with very few particles such as 20, the result is comparable to that of using 500 particles. Figure 5.4 displays results from a SPCGP (*top left*), PLGP (*bottom left*), PCGP (*top right*), and a standard GP with MCMC (*bottom right*).

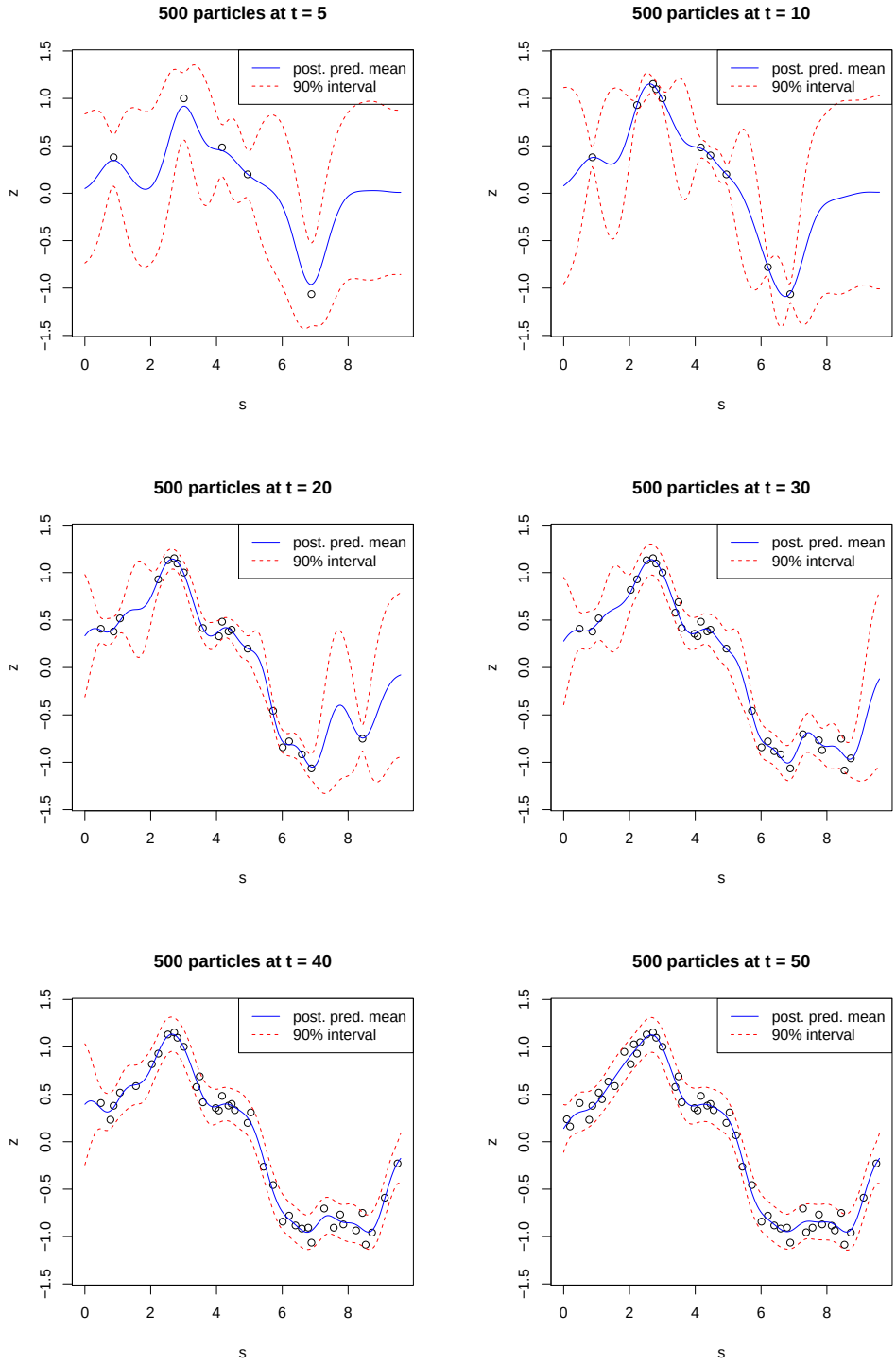


Figure 5.2: Posterior predictive summary from SPCGP for $t = \{5, 10, 20, 30, 40, 50\}$

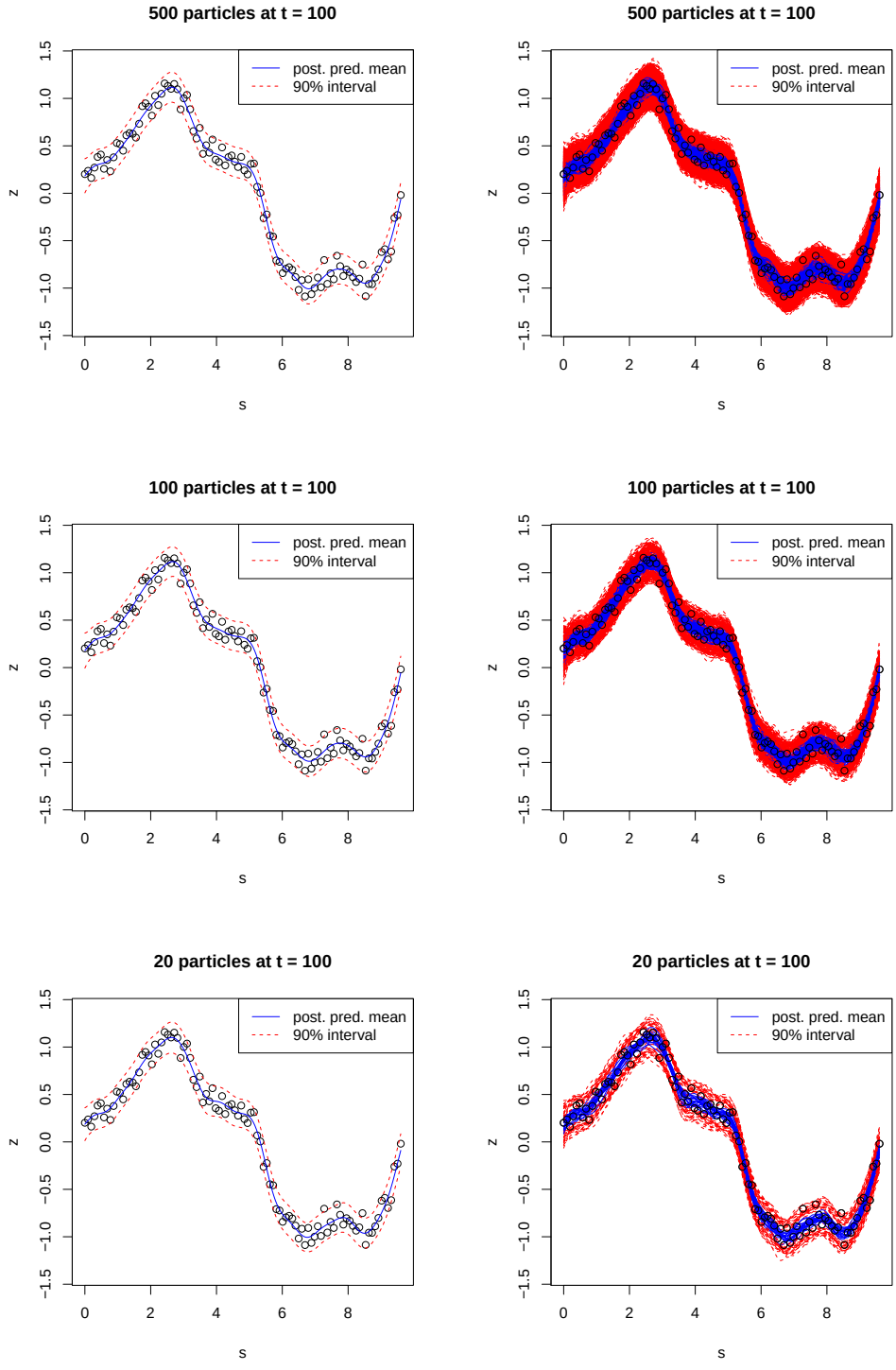


Figure 5.3: Posterior predictive summary from SPCGP for $t = 100$ with $500, 100$, and 20 particles

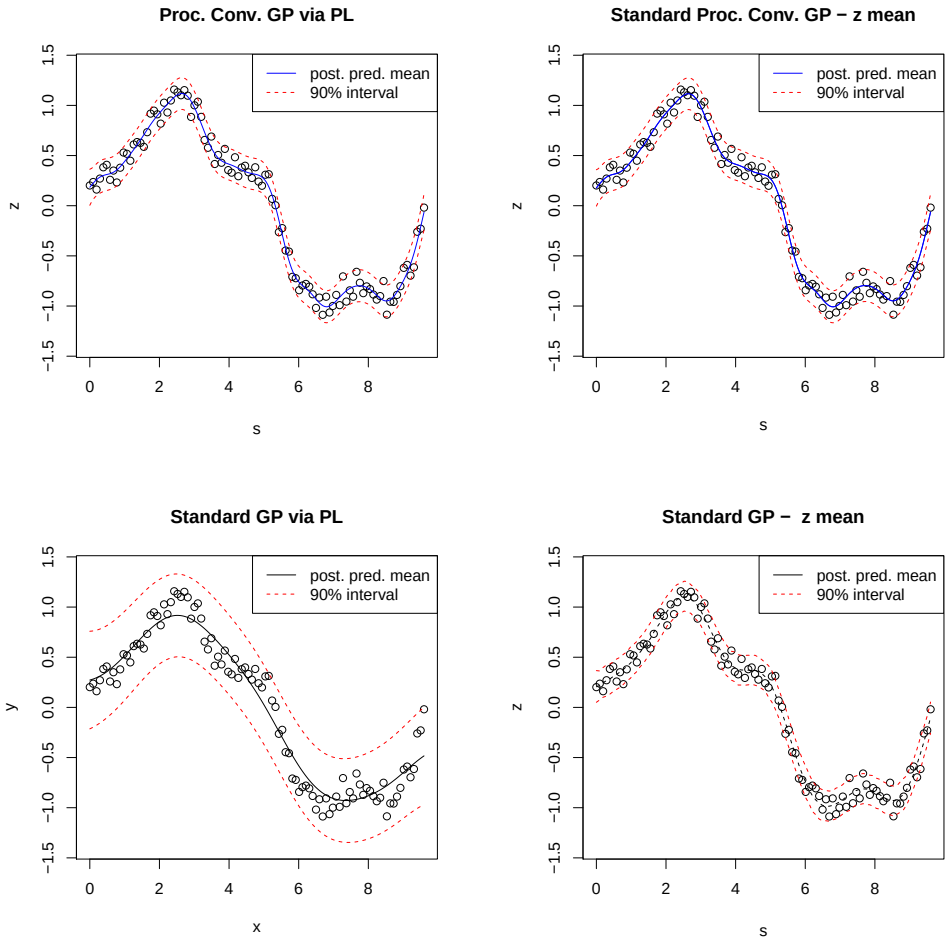


Figure 5.4: Posterior predictive summary based on SPCGP (*top left*), PCGP (*top right*), PLGP (*bottom left*), and standard GP (*bottom right*)

Results from both sequential approaches are displayed for $t = 100$ with 500 particles. The MCMC approaches are based on a total of 600 iterations with a burn-in of 100 so that the number of samples is 500. Note that the posterior predictive summaries from all models are quite similar except for PLGP. It produces a mean surface that over smoothes the data and fails to capture the local features in the true response. The resulting 90% posterior predictive interval is an over estimate of the true data variability. The misfitting might be due to the default prior parameters in the *plgp* R package. In contrast, SPCGP is able to capture the true response, both globally and locally, and also has a more reasonable prediction for the data variability.

### 5.4.2 The Pump-and-Treat Problem

The Pump-and-Treat problem (Matott et al., 2011) involves a groundwater contamination scenario based on the Lockwood Solvent Groundwater Plume Site located near Billings, Montana. Two plumes (A and B) containing chlorinated solvents were developed due to industrial practices near the Yellowstone river as shown in Figure 5.5. Of interest is plume A located in the southern section of the site. The primary concern is to prevent the plume from migrating to and contaminating the Yellowstone river. The proposed remediation involves drilling two pump-and-treat wells. This problem has been modeled using a computer simulator where the inputs are pumping rates for the two pump-and-treat wells, and the output is a cost function which combines the financial cost of running the wells with a large penalty for any contamination of the river (the penalty ensures that any optimal solution will not allow any contamination

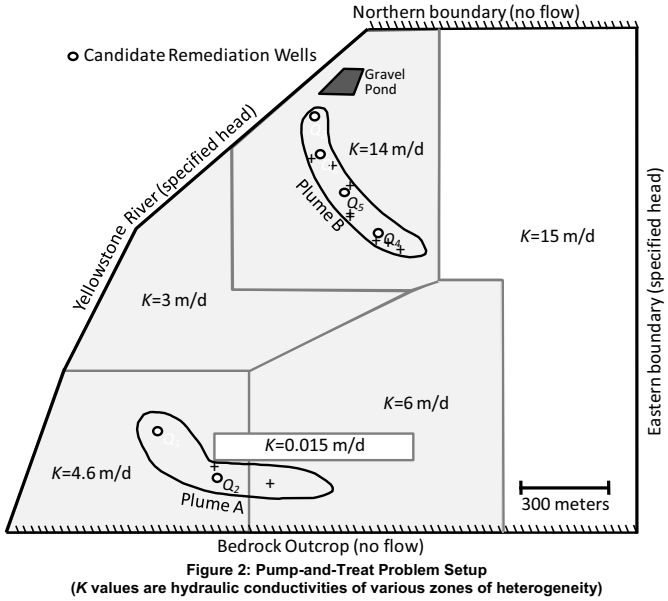


Figure 5.5: Lockwood Solvent Groundwater Plume Site located near Billings, Montana

of the river). The two pumping rates can be set between 0 and 20,000. The objective is to minimize the cost function, i.e., the expense of running the wells. Because of the non-trivial time for each simulator run, it is not possible to run the simulator at every possible combination of inputs and find the one that has the minimum cost. Instead, a computer simulation experiment approach (Sacks et al., 1989) is taken to sequentially build a surrogate model while searching for the minimum of the surface (Jones et al., 1998; Taddy et al., 2009; Gramacy and Lee, 2011). This method proceeds sequentially by adding new design points (a pair of pump rates and the associated cost) one-by-one based on some criterion and update the model fit conditional on the new design point. Updating of the model fit could be done with MCMC, however, it could be computationally demanding since the MCMC has to be repeated for every new design

point. Instead, SPCGP is applied to this problem. The model is setup by specifying a $(25\times25)$ basis grid, and a Bézier kernel whose circular support has a radius of three times the spacing of any two adjacent bases. To choose the new data point (simulator run), the expected improvement (EI) approach (Jones et al., 1998) is employed by choosing the point $\mathbf{s}$ that maximizes

$$E[I^g(\mathbf{s})] = E[max(f_{best} - f(\mathbf{s}))^g, 0], \tag{5.11}$$

where $f_{best}$ denotes the current best point (inputs with minimum response) and $f(\mathbf{s})$ denotes the predicted output response (from the current state of SPCGP) at input $\mathbf{s}$. The power $g$ can be specified to tune the local versus global character of the optimization. For example, $g = 1$ yields the standard expected improvement statistic, while for $g = 2$, $E[I^2(\mathbf{s})] = var[I(\mathbf{s})] + E[I(\mathbf{s})]^2$ explicitly rewards the improvement in variance and thus gives relatively more weight toward global exploration of the response surface. Since finding the maximizing $\mathbf{s}$ exactly would be another difficult problem, optimization is approximated by considering 200 candidate points in the input space generated using Latin hypercube sampling (McKay et al., 1979) and then choosing the candidate point with largest expected improvement (Gramacy and Lee, 2009). The initial priors are specified as

$$\beta \sim N(0,\ \phi^{-1}), \quad \phi \sim G(1,\ 0.0001),$$

$$\mathbf{x}|\lambda \sim N_m(\mathbf{0},\ (\lambda\mathbf{I}_m)^{-1}), \quad \lambda \sim G(1,\ 0.001),$$

where $\beta$ denotes the mean level (intercept). A narrow $G(1,\ 0.0001)$ prior is imposed on $\phi$ because the simulator is approximating a deterministic process, however, it makes

sense to leave some room for any error that might result, and this error is expected to be relatively small. A total of 60 input points are considered, where the first 30 are generated from a Latin hypercube to build an initial model surface, and the remaining 30 are chosen sequentially one-by-one using the EI approach described above. A total of 500 particles are used for the simulation.

Fixing $g = 1$, the posterior predictive mean surfaces from SPCGP are shown for $t = \{6, 9, 15, 30, 40, 50\}$ in Figure 5.6. For $t = \{6, 9, 15\}$, the mean surfaces illustrate the intermediate results during the process of generating the initial model surface for $t = 30$. Starting from $t = 30$, the EI algorithm is deployed and it is apparent that most of the design points considered are near the minimum of the mean surface. This is expected because the goal of this approach is to explore the input space in areas that are likely to provide the minimum response. The posterior predictive summary from SPCGP and PLGP at $t = 60$ are shown in the top and bottom panels of Figure 5.7, respectively. The middle panel displays results from repeatedly applying PCGP with 600 MCMC iterations (100 burn-in and 500 samples) to each newly generated design point. The mean surface from these models are resembling in general, with SPCGP and PCGP predicting more local features than PLGP. The 90% interval width of both SPCGP and PCGP have low uncertainty at the data locations and high uncertainty at the unobserved locations, whereas that of PLGP seem to overestimate/underestimate the uncertainty at the observed/unobserved locations. Nonetheless, the locations of the predicted minimums found by these models are comparable to one another. Figure 5.8 displays the posterior predictive mean surfaces under 500 (*top*), 100 (*middle*), and 20

(*bottom*) particles with $g = 1$ (*left*) and $g = 2$ (*right*). Results that have the same $g$ value are closely resembling. For different $g$ values, the difference comes from the fact that the $g = 2$ case tends to fit a better overall surface than the $g = 1$ case by exploring places with high uncertainty. Nonetheless, the locations of minimums are similar for all, even with very few particles such as 20. Since finding the minimum location is the main purpose of this problem, all models and settings considered here have comparable performance in this regard. All simulations are run on an Intel Core 2 duo CPU at 2.4 Ghz with 4 Gb of RAM. Table 5.1 and 5.2 display the running times of SPCGP, PCGP, and PLGP v.s. the number of LHS candidates (200, 500, and 1000) with 500 particles (or MCMC samples) at $g = 1$. The left panel displays the average updating time (model update and prediction on LHS candidates, but not including simulator time) for each design point, and the right panel shows the total running time for the whole process. For 200 LHS candidates, the average updating time for SPCGP is 15.78 seconds, and that of PCGP is 21.09 seconds. The difference is not too big and mostly due to the extra 100 burn-in iterations in PCGP. However, this difference could be exacerbated in more complicated datasets, where MCMC might need a larger number of burn-in steps to reach equilibrium, and also require thinning in order to obtain less correlated samples. These mechanisms can greatly increase the computational time of MCMC. In contrast, such difficulties generally do not exist in PL. In addition, PL (or generally, SMC) is completely parallelizable since the particles can be updated independently of one another up to having a unique computing node for each particle. In the case of PLGP, the average updating time and total running time are not too much larger than

those of SPCGP for 200 LHS candidates. However, as the number of LHS candidates increases, PLGP significantly slows down, while SPCGP maintains roughly the same speed. This suggests that for more complicated problems where a higher number of LHS candidates are needed, SPCGP is computationally more efficient than PLGP (for low dimensional problems).

## 5.5 Conclusion

On-line inference for a GP model based on MCMC is inefficient because it requires re-running the MCMC for every new data arrival, which can be computationally demanding due to slow convergence. In this chapter, a sequential inference approach for the process convolution GP model is developed, which is based on a method called Particle Learning. It allows parameter inference to be performed on-line for each new batch of data without having to use MCMC. This convolution approach allows for the handling of much larger datasets than would not be computationally feasible under the standard GP approach. This is because the computational expense is tied to the number of background process points instead of the number of datapoints, although the convolution approach is only practical for lower-dimensional problems because of the need to create a grid of background process points. Another advantage of SMC methods is that they are completely parallelizable - the particles can be updated independently of one another. In contrast, MCMC has to be done, to a large extent, in serial. Illustrations of SPCGP on a 1-d synthetic dataset and a 2-d optimization problem show promising

results in terms of model fitting and computational speed, robustness in optimization, as well as robustness with respect to the number of particles required.

Table 5.1: Average updating time (seconds)

| | Number of LHS candidates | | |
|---|---|---|---|
| Model | 200 | 500 | 1000 |
| SPCGP | 15.78 | 16.09 | 16.10 |
| PCGP | 21.16 | 22.15 | 21.72 |
| PLGP | 19.28 | 96.96 | 263.02 |

Table 5.2: Total running time (seconds)

| | Number of LHS candidates | | |
|---|---|---|---|
| Model | 200 | 500 | 1000 |
| SPCGP | 595.73 | 610.96 | 618.69 |
| PCGP | 759.27 | 792.964 | 789.89 |
| PLGP | 700.90 | 3030.97 | 8012.80 |

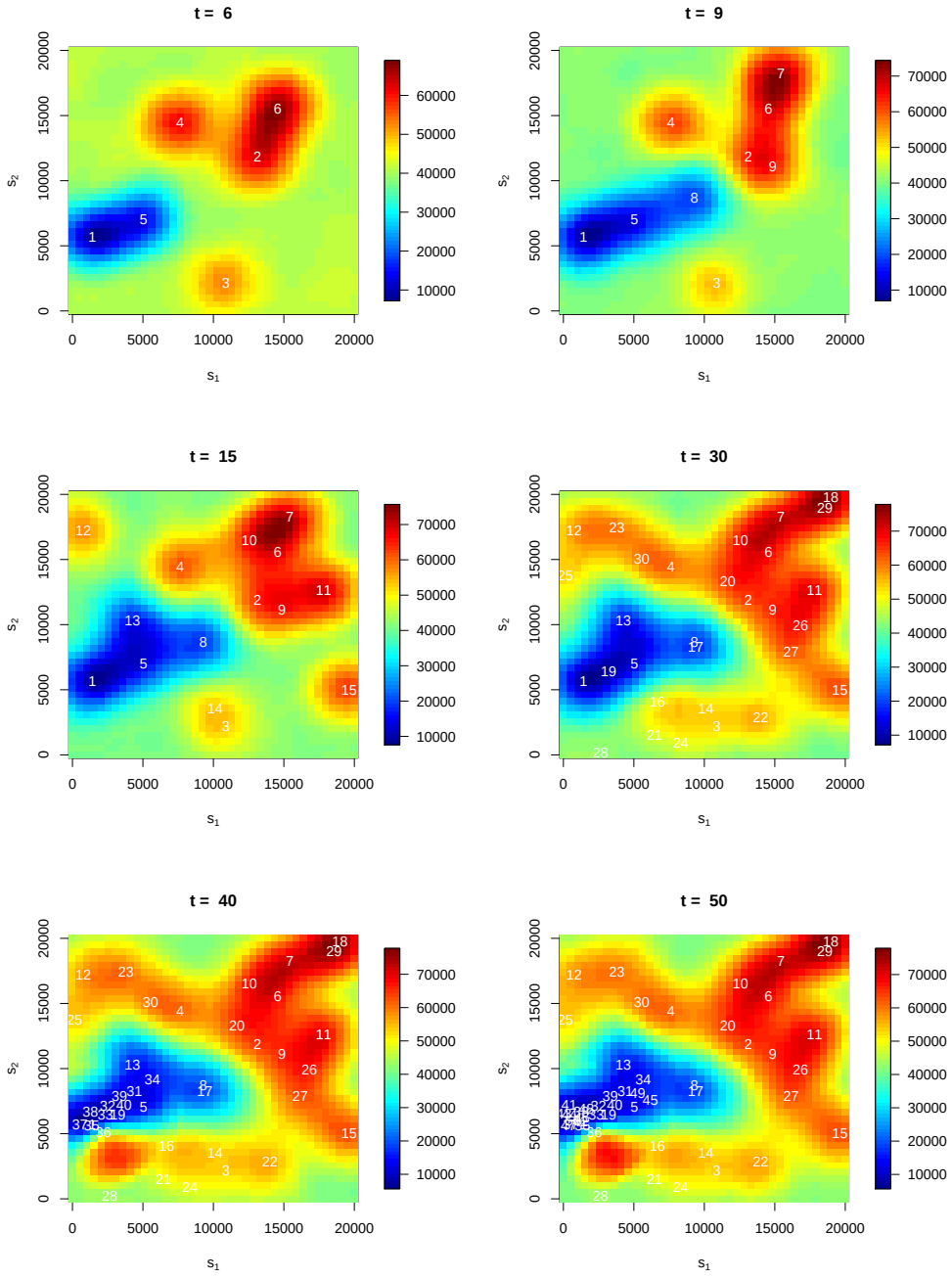


Figure 5.6: Posterior predictive mean from SPCGP with $g = 1$ for $t = \{6, 9, 15, 30, 40, 50\}$ for the Pump-and-Treat problem

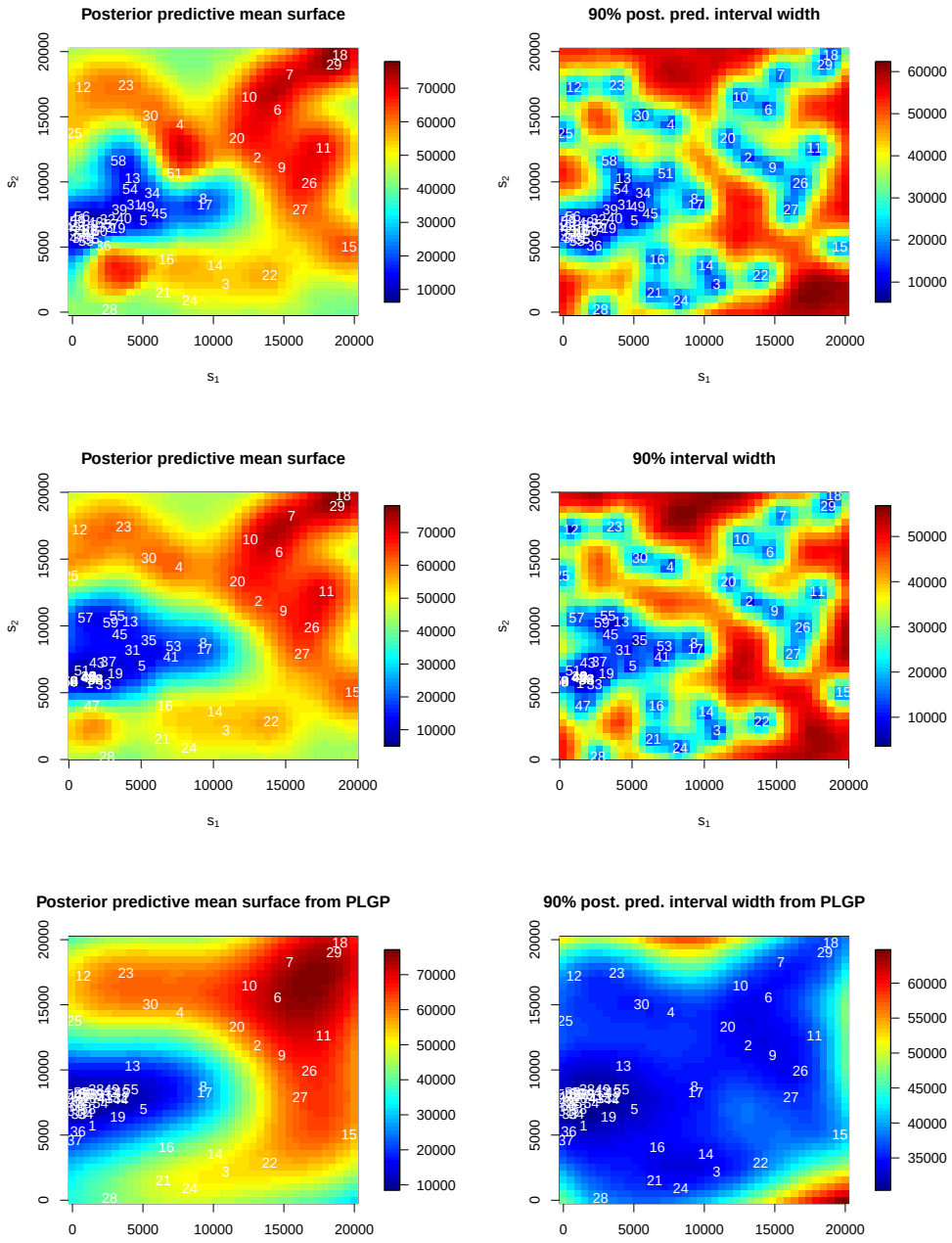


Figure 5.7: Posterior predictive summary from SPCGP (*top*), PCGP (*middle*), and PLGP (*bottom*) with $g = 1$ at $t = 60$ for the Pump-and-Treat problem

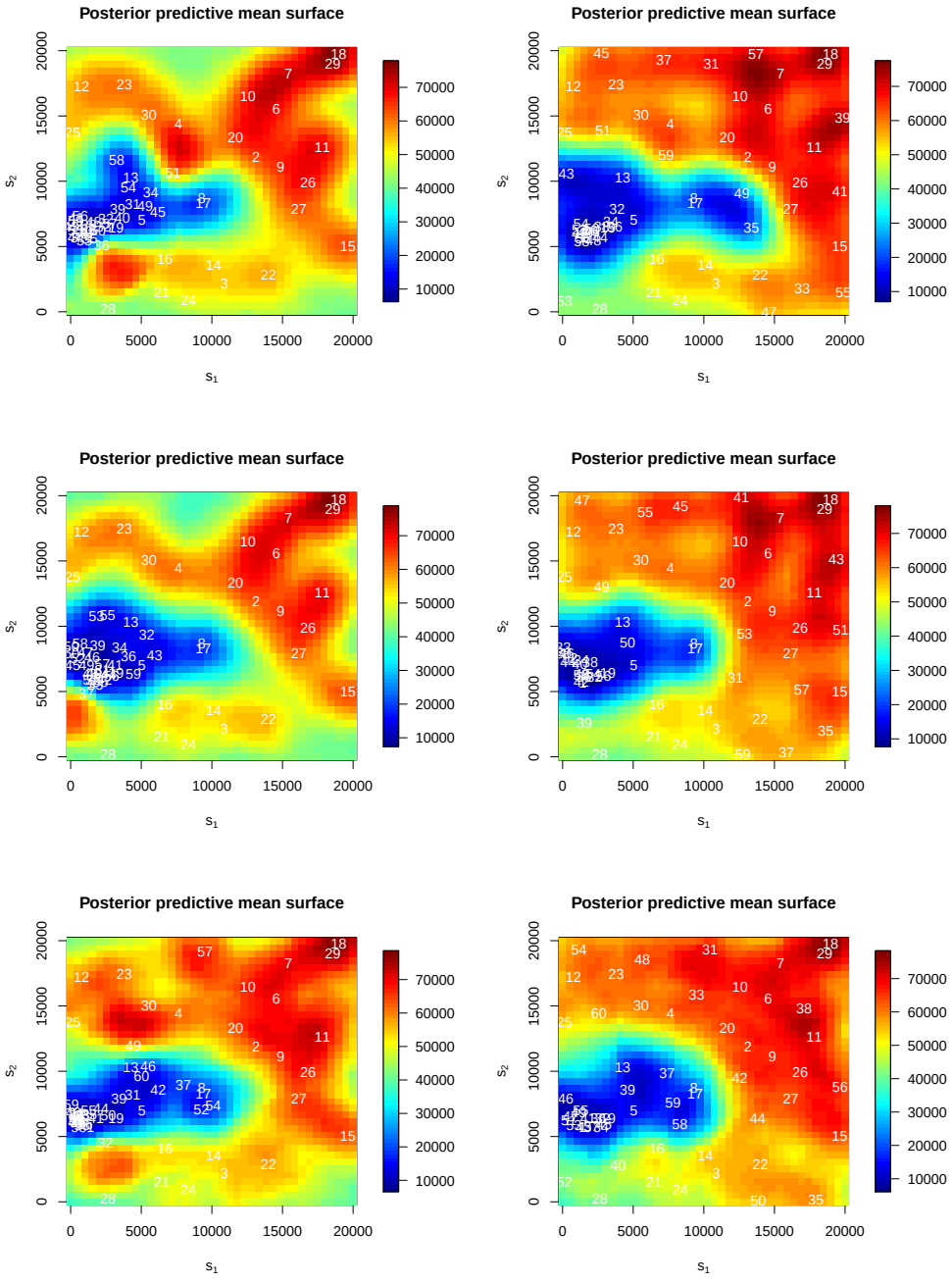


Figure 5.8: Posterior predictive mean from SPCGP based on 500 (*top*), 100 (*middle*), and 20 (*bottom*) particles for $g = 1$ (*left*) and $g = 2$ (*right*) at $t = 60$ for the Pump-and-Treat problem

# Chapter 6

# Conclusion and Future Work

This dissertation centers around the process convolution approach of building a Gaussian process model. In general, a GP can be constructed by convolving a smoothing kernel with a discrete latent process having a Gaussian prior. As an introduction, a simulation study on this approach is given, with a particular interest in finding the sufficient number of bases required for satisfactory model performance. Then, a non-stationary GP model (TPCGP) is developed based on this approach by partitioning the spatial domain and allowing a separate latent process and kernel for each partition. Partitioning is achieved using a binary tree generating process. A Bayesian approach is used to guide partitioning and estimate the parameters simultaneously. Results show that TPCGP has promising performance in terms of model fitting, prediction, and computational speed. On the other hand, a sequential inference approach for the standard process convolution GP model (SPCGP) is developed. This approach is based on a Sequential Monte Carlo method called Particle Learning. Results show that SPCGP

makes on-line inference more efficient as opposed to traditional MCMC inference and other competing approaches. Moreover, this sequential design is fairly robust even with few particles, and is completely parallelizable thus has potential for a more efficient implementation.

The work presented in this dissertation is only a subset of the potential area that can be explored, especially for TPCGP. Possible future work for TPCGP includes having non-axis aligned partitioning methods, e.g., the Voronoi partitioning. One example of such models is given by Kim et al. (2005), where a traditional GP is considered for each Voronoi partition. Another approach is to modify the current partitioning method by allowing the split angle to vary. Currently, the setup is that each binary split is a straight line parallel to one of the axes and perpendicular to all others. Alternatively, a binary split can be made at an angle and different split angles can be estimated for different partitions, along with other parameters. To take one more step forward, a more flexible approach would be to define a parametric curve for each binary split and estimate the curves from data. Another possible extension would be to pre-specify partitions such as using the boundaries of counties or states. This approach can be useful for modeling a process, e.g., housing prices, whose characteristic are closely related to features in the local region. Partitioning in TPCGP is one of the key determinants in the way kernels are vary, which is critical for capturing local structure in the data. Having a more flexible partitioning method is likely to improve the learning ability of the model. On the other hand, TPCGP can be further extended into a spatio-temporal model. A similar attempt has been done by Calder et al. (2002) and Lemos and Sansó (2009)

for the standard process convolution GP, where the background process is treated as a state indexed by time, and a transition equation is defined for the temporal evolution of the background process. It is possible to extend TPCGP in the same manner, which requires also defining a reasonable transition between the tree structures over time. Inference for this extension can be performed using Particle Learning since the temporal evolution of tree structure would make inference not amenable to Kalman filter. All of these ideas will be considered for future research and journal publications.

# Appendix A

# Derivation

This appendix provides derivation of the conditional posterior distributions for the parameters and model structure of the treed process convolution Gaussian process model (TPCGP). Assuming that the spatial domain $\mathcal{S}$ is partitioned into $b$ disjoint regions $\{\mathcal{S}_\nu : \nu = 1, \cdots, b\}$ as described in Section 1.7. The sampling distribution in partition $\nu$ is given by

$$\begin{aligned} & f(\mathbf{y}_\nu|\boldsymbol{\beta}_\nu, \phi_\nu, \mathbf{Q}_\nu, \mathbf{x}, \boldsymbol{\rho}, \mathcal{T}, \mathbf{F}_\nu) \\ = & \Big(\frac{\phi_\nu}{2\pi}\Big)^{n_\nu/2} \exp\Big\{ -\frac{\phi_\nu}{2}(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})^\top(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})\Big\}, \end{aligned}$$

where $\mathbf{K}_\nu$ depends on the kernel precision matrix $\mathbf{Q}_\nu$. The joint likelihood is given by

$$\begin{aligned} & L(\{\boldsymbol{\beta}_\nu, \phi_\nu, \mathbf{Q}_\nu\}_{\nu=1}^b, \mathbf{x}, \boldsymbol{\rho}, \mathcal{T}|\mathbf{y}, \mathbf{F}) \\ = & \prod_{\nu=1}^b f(\mathbf{y}_\nu|\boldsymbol{\beta}_\nu, \phi_\nu, \mathbf{Q}_\nu, \mathbf{x}, \boldsymbol{\rho}, \mathcal{T}, \mathbf{F}_\nu) \\ = & \prod_{\nu=1}^b \Big(\frac{\phi_\nu}{2\pi}\Big)^{n_\nu/2} \exp\Big\{ -\frac{\phi_\nu}{2}(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})^\top(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})\Big\}, \end{aligned}$$

where $\mathbf{x} = (\mathbf{x}_1^\top, \cdots, \mathbf{x}_b^\top)^\top$ such that $\mathbf{x}_\nu$ denotes the latent process vector in $\mathcal{S}_\nu$. Suppose that the following conjugate priors are imposed on the leaf node parameters:

$$\boldsymbol{\beta}_\nu|\phi_\nu, \boldsymbol{\beta}_0, \mathbf{C}, \boldsymbol{\rho}, \mathcal{T} \sim N_{p+1}(\boldsymbol{\beta}_0\,, (\phi_\nu \mathbf{C})^{-1}), \quad \mathbf{x}_\nu|\lambda_\nu, \boldsymbol{\rho}, \mathcal{T} \sim N_{m_\nu}(\mathbf{0}\,, (\lambda_\nu \mathbf{I}_{m_\nu})^{-1}),$$

$$\phi_\nu|b_y, \boldsymbol{\rho}, \mathcal{T} \sim G(a_y\,, b_y), \quad \lambda_\nu|b_x, \boldsymbol{\rho}, \mathcal{T} \sim G(a_x\,, b_x), \quad \mathbf{Q}_\nu|\boldsymbol{\rho}, \mathcal{T} \sim W((\psi\mathbf{H})^{-1}, \psi),$$

$$\boldsymbol{\beta}_0 \sim N_{p+1}(\boldsymbol{\mu}, \mathbf{B}^{-1}), \quad \mathbf{C} \sim W((\varphi\mathbf{V})^{-1}, \varphi), \quad b_x \sim G(\tau_x, \xi_x), \quad b_y \sim G(\tau_y, \xi_y),$$

where $\{a_x, a_y, \boldsymbol{\mu}, \mathbf{B}, \varphi, \mathbf{V}, \tau_x, \xi_x, \tau_y, \xi_y, \psi, \mathbf{H}\}$ are constants.

## A.1 Joint posterior distribution

$$\begin{aligned}
& P(\{\boldsymbol{\beta}_\nu, \phi_\nu, \lambda_\nu, \mathbf{Q}_\nu\}_{\nu=1}^b, \mathbf{x}, \boldsymbol{\beta_0}, \mathbf{C}, b_x, b_y, \boldsymbol{\rho}, \mathcal{T}|\mathbf{y}, \mathbf{F}) \\
\propto\ & \Bigg( \prod_{\nu=1}^b L(\{\boldsymbol{\beta}_\nu, \phi_\nu, \mathbf{Q}_\nu\}_{\nu=1}^b, \mathbf{x}, \boldsymbol{\rho}, \mathcal{T}|\mathbf{y}, \mathbf{F}) P(\boldsymbol{\beta}_\nu|\phi_\nu, \boldsymbol{\beta}_0, \mathbf{C}, \boldsymbol{\rho}, \mathcal{T}) P(\mathbf{x}_\nu|\lambda_\nu, \boldsymbol{\rho}, \mathcal{T}) \times \\
& P(\phi_\nu|b_y, \boldsymbol{\rho}, \mathcal{T}) P(\lambda_\nu|b_x, \boldsymbol{\rho}, \mathcal{T}) P(\mathbf{Q}_\nu|\boldsymbol{\rho}, \mathcal{T}) \Bigg) P(\boldsymbol{\beta}_0) P(\mathbf{C}) P(b_x) P(b_y) P(\boldsymbol{\rho}|\mathcal{T}) P(\mathcal{T}) \\
\propto\ & \Bigg( \prod_{\nu=1}^b \Big(\frac{\phi_\nu}{2\pi}\Big)^{n_\nu/2} \exp\Big\{ -\frac{\phi_\nu}{2} (\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})^\top (\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x}) \Big\} \\
& \Big(\frac{\phi_\nu}{2\pi}\Big)^{(p+1)/2} |\mathbf{C}|^{1/2} \exp\Big\{ -\frac{\phi_\nu}{2} (\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)^\top \mathbf{C} (\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0) \Big\} \times \\
& \Big(\frac{\lambda_\nu}{2\pi}\Big)^{m_\nu/2} \exp\Big\{ -\frac{\lambda_\nu}{2} \mathbf{x}_\nu^\top \mathbf{x}_\nu \Big\} \times \frac{b_y^{a_y}}{\Gamma(a_y)} \phi_\nu^{a_y - 1} \exp\{-b_y\phi_\nu\} \times \frac{b_x^{a_x}}{\Gamma(a_x)} \lambda_\nu^{a_x - 1} \exp\{-b_x\lambda_\nu\} \times \\
& \frac{|\mathbf{Q}_\nu|^{(\psi - r - 1)/2} |\psi\mathbf{H}|^{\psi/2}}{2^{\psi r/2} \Gamma_r(\psi/2)} \exp\Big\{ -\frac{1}{2} \mathrm{tr}(\psi\mathbf{H}\mathbf{Q}_\nu) \Big\} \Bigg) \times \\
& \Big(\frac{1}{2\pi}\Big)^{(p+1)/2} |\mathbf{B}|^{1/2} \exp\Big\{ -\frac{1}{2} (\boldsymbol{\beta}_0 - \boldsymbol{\mu})^\top \mathbf{B} (\boldsymbol{\beta}_0 - \boldsymbol{\mu}) \Big\} \times \\
& \frac{|\mathbf{C}|^{(\varphi - p)/2} |\varphi\mathbf{V}|^{\varphi/2}}{2^{\varphi(p+1)/2} \Gamma_{p+1}(\varphi/2)} \exp\Big\{ -\frac{1}{2} \mathrm{tr}(\varphi\mathbf{V}\mathbf{C}) \Big\} \times \\
& \frac{\xi_x^{\tau_x}}{\Gamma(\tau_x)} b_x^{\tau_x - 1} \exp\{-\xi_x b_x\} \times \frac{\xi_y^{\tau_y}}{\Gamma(\tau_y)} b_y^{\tau_y - 1} \exp\{-\xi_y b_y\} \times P(\boldsymbol{\rho}|\mathcal{T}) P(\mathcal{T}). \qquad \text{(A.1)}
\end{aligned}$$

## A.2 Conditional Posterior Distribution for x

The conditional posterior distribution of $\mathbf{x}$ depends on all other parameters and is given by

$$
\begin{aligned}
& P(\mathbf{x}|\{\boldsymbol{\beta}_\nu, \phi_\nu, \lambda_\nu, \mathbf{Q}_\nu\}_{\nu=1}^b, \boldsymbol{\rho}, \mathcal{T}, \mathbf{y}, \mathbf{F}) \\
\propto & \prod_{\nu=1}^{b} \exp\Big\{ -\frac{\phi_\nu}{2}(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})^\top(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})\Big\} \exp\Big\{ -\frac{\lambda_\nu}{2}\mathbf{x}_\nu^\top\mathbf{x}_\nu\Big\} \\
\propto & \exp\Big\{ -\frac{1}{2}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta} - \mathbf{K}\mathbf{x})^\top\boldsymbol{\Phi}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta} - \mathbf{K}\mathbf{x})\Big\} \exp\Big\{ -\frac{1}{2}\mathbf{x}^\top\boldsymbol{\Lambda}\mathbf{x}\Big\} \\
\propto & \exp\Big\{ -\frac{1}{2}\Big(\mathbf{w}^\top\boldsymbol{\Phi}\mathbf{w} + \mathbf{x}^\top\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{K}\mathbf{x} - 2\mathbf{x}^\top\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{w} + \mathbf{x}^\top\boldsymbol{\Lambda}\mathbf{x}\Big)\Big\} \\
\propto & \exp\Big\{ -\frac{1}{2}\Big(\mathbf{x}^\top(\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{K} + \boldsymbol{\Lambda})\mathbf{x} - 2\mathbf{x}^\top\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{w}\Big)\Big\} \\
\propto & \exp\Big\{ -\frac{1}{2}\Big(\mathbf{x}^\top(\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{K} + \boldsymbol{\Lambda})\mathbf{x} - 2\mathbf{x}^\top(\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{K} + \boldsymbol{\Lambda})(\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{K} + \boldsymbol{\Lambda})^{-1}\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{w}\Big)\Big\} \\
\propto & \exp\Big\{ -\frac{1}{2}(\mathbf{x} - (\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{K} + \boldsymbol{\Lambda})^{-1}\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{w})^\top(\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{K} + \boldsymbol{\Lambda})(\mathbf{x} - (\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{K} + \boldsymbol{\Lambda})^{-1}\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{w})\Big\},
\end{aligned}
$$

which is a multivariate Gaussian distribution:

$$
\begin{aligned}
& \mathbf{x}|\{\boldsymbol{\beta}_\nu, \phi_\nu, \lambda_\nu, \mathbf{Q}_\nu\}_{\nu=1}^b, \boldsymbol{\rho}, \mathcal{T}, \mathbf{y}, \mathbf{F} \\
\sim & N_m\Big((\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{K} + \boldsymbol{\Lambda})^{-1}\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{w}\ ,\ (\mathbf{K}^\top\boldsymbol{\Phi}\mathbf{K} + \boldsymbol{\Lambda})^{-1}\Big), \qquad \text{(A.2)}
\end{aligned}
$$

where

$$
\boldsymbol{\Phi} = \begin{pmatrix} \boldsymbol{\Phi}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Phi}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{\Phi}_b \end{pmatrix}, \quad \boldsymbol{\Lambda} = \begin{pmatrix} \boldsymbol{\Lambda}_1 & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Lambda}_2 & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \boldsymbol{\Lambda}_b \end{pmatrix},
$$

$$
\boldsymbol{\Phi}_\nu = \phi_\nu\mathbf{I}_{n_\nu}, \quad \boldsymbol{\Lambda}_\nu = \lambda_\nu\mathbf{I}_{n_\nu}, \quad \mathbf{w} = \mathbf{y} - \mathbf{F}\boldsymbol{\beta}.
$$

## A.3 Conditional Posterior Distribution for $\lambda_\nu$

The conditional posterior distribution of $\lambda_\nu$ is obtained by integrating out $\boldsymbol{\beta}_\nu$ and $\boldsymbol{\phi}_\nu$ from the joint posterior:

$$
\begin{aligned}
& P(\{\lambda_\nu\}_{\nu=1}^b|\mathbf{x}, b_x, \boldsymbol{\rho}, \mathcal{T}) \\
\propto\ & \Bigg( \prod_{\nu=1}^{b} \int \Big(\frac{\phi_\nu}{2\pi}\Big)^{n_\nu/2} \exp\Big\{ -\frac{\phi_\nu}{2}(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})^\top(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})\Big\} \times \\
& \Big(\frac{\phi_\nu}{2\pi}\Big)^{(p+1)/2} \det(\mathbf{C}) \exp\Big\{ -\frac{\phi_\nu}{2}\boldsymbol{\beta}_\nu^\top\mathbf{C}\boldsymbol{\beta}_\nu\Big\} \frac{b_y^{a_y}}{\Gamma(a_y)}\phi_\nu^{a_y-1}\exp\{-b_y\phi_\nu\} \times \\
& \Big(\frac{\lambda_\nu}{2\pi}\Big)^{m_\nu/2} \exp\Big\{ -\frac{\lambda_\nu}{2}\mathbf{x}_\nu^\top\mathbf{x}_\nu\Big\} \frac{b_x^{a_x}}{\Gamma(a_x)}\lambda_\nu^{a_x-1}\exp\{-b_x\lambda_\nu\} d\boldsymbol{\beta}_\nu d\phi_\nu \Bigg) P(\boldsymbol{\rho}|\mathcal{T})P(\mathcal{T}) \\
\propto\ & \prod_{\nu=1}^{b} \lambda_\nu^{m_\nu/2+a_x-1} \exp\Big\{ -\Big(\frac{1}{2}\mathbf{x}_\nu^\top\mathbf{x}_\nu + b_x\Big)\lambda_\nu\Big\}.
\end{aligned}
$$

It can be shown that $\lambda_\nu$ follows a Gamma distribution:

$$
\lambda_\nu|\mathbf{x}_\nu, b_x, \boldsymbol{\rho}, \mathcal{T} \ \sim\ G\Big(\frac{m_\nu}{2} + a_x\ ,\ \frac{1}{2}\mathbf{x}_\nu^\top\mathbf{x}_\nu + b_x\Big). \tag{A.3}
$$

## A.4 Conditional Posterior Distribution for $\phi_\nu$

The conditional posterior distribution of $\phi_\nu$ is obtained by integrating out $\boldsymbol{\beta}_\nu$ from the joint posterior:

$$
\begin{aligned}
& P(\phi_\nu|\mathbf{x}, \mathbf{Q}, \boldsymbol{\beta}_0, \mathbf{C}, b_y, \boldsymbol{\rho}, \mathcal{T}, \mathbf{y}, \mathbf{F}) \\
\propto\ & \prod_{\nu=1}^{b} \int \Big(\frac{\phi_\nu}{2\pi}\Big)^{n_\nu/2} \exp\Big\{ -\frac{\phi_\nu}{2}(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})^\top(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})\Big\} \times \\
& \Big(\frac{\phi_\nu}{2\pi}\Big)^{(p+1)/2} |\mathbf{C}|^{1/2} \exp\Big\{ -\frac{\phi_\nu}{2}(\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)^\top\mathbf{C}(\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)\Big\} \times \phi_\nu^{a_y-1}\exp\{-b_y\phi_\nu\} d\boldsymbol{\beta}_\nu
\end{aligned}
$$

$$
\begin{aligned}
&\propto \prod_{\nu=1}^{b} \int \phi_\nu^{(n_\nu/2+(p+1)/2+a_y)-1} \exp\Big\{ -\frac{\phi_\nu}{2}\Big(s_\nu^2 + (\boldsymbol{\beta}_\nu - \hat{\boldsymbol{\beta}}_\nu)^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)(\boldsymbol{\beta}_\nu - \hat{\boldsymbol{\beta}}_\nu) + \\
&(\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)^\top \mathbf{C}(\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0) + 2b_y\Big)\Big\} d\boldsymbol{\beta}_\nu \\
&\propto \prod_{\nu=1}^{b} \int \phi_\nu^{(n_\nu/2+(p+1)/2+a_y)-1} \exp\Big\{ -\frac{\phi_\nu}{2}\Big(s_\nu^2 + \boldsymbol{\beta}_\nu^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)\boldsymbol{\beta}_\nu + \hat{\boldsymbol{\beta}}_\nu^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu - \\
&2\boldsymbol{\beta}_\nu^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu + \boldsymbol{\beta}_\nu^\top \mathbf{C}\boldsymbol{\beta}_\nu + \boldsymbol{\beta}_0^\top \mathbf{C}\boldsymbol{\beta}_0 - 2\boldsymbol{\beta}_\nu^\top \mathbf{C}\boldsymbol{\beta}_0 + 2b_y\Big)\Big\} d\boldsymbol{\beta}_\nu \\
&\propto \prod_{\nu=1}^{b} \phi_\nu^{(n_\nu/2+(p+1)/2+a_y)-1} \exp\Big\{ -\frac{\phi_\nu}{2}\Big(s_\nu^2 + 2b_y + \hat{\boldsymbol{\beta}}_\nu^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu + \boldsymbol{\beta}_0^\top \mathbf{C}\boldsymbol{\beta}_0\Big)\Big\} \times \\
&\int \exp\Big\{ -\frac{\phi_\nu}{2}\Big(\boldsymbol{\beta}_\nu^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)\boldsymbol{\beta}_\nu - 2\boldsymbol{\beta}_\nu^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu + \boldsymbol{\beta}_\nu^\top \mathbf{C}\boldsymbol{\beta}_\nu - 2\boldsymbol{\beta}_\nu^\top \mathbf{C}\boldsymbol{\beta}_0\Big)\Big\} d\boldsymbol{\beta}_\nu \\
&\propto \prod_{\nu=1}^{b} \phi_\nu^{(n_\nu/2+(p+1)/2+a_y)-1} \exp\Big\{ -\frac{\phi_\nu}{2}\Big(s_\nu^2 + 2b_y + \hat{\boldsymbol{\beta}}_\nu^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu + \boldsymbol{\beta}_0^\top \mathbf{C}\boldsymbol{\beta}_0\Big)\Big\} \times \\
&\int \exp\Big\{ -\frac{\phi_\nu}{2}\Big(\boldsymbol{\beta}_\nu^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})\boldsymbol{\beta}_\nu - 2\boldsymbol{\beta}_\nu^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})(\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})^{-1} \times \\
&((\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu + \mathbf{C}\boldsymbol{\beta}_0)\Big)\Big\} d\boldsymbol{\beta}_\nu \\
&\propto \prod_{\nu=1}^{b} \phi_\nu^{(n_\nu/2+(p+1)/2+a_y)-1} \exp\Big\{ -\frac{\phi_\nu}{2}\Big(s_\nu^2 + 2b_y + \hat{\boldsymbol{\beta}}_\nu^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu + \boldsymbol{\beta}_0^\top \mathbf{C}\boldsymbol{\beta}_0 - \\
&((\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu + \mathbf{C}\boldsymbol{\beta}_0)^\top ((\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})^{-1})^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})(\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})^{-1} \times \\
&((\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu + \mathbf{C}\boldsymbol{\beta}_0)\Big)\Big\} |(\phi_\nu(\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C}))^{-1}|^{1/2} \\
&\propto \prod_{\nu=1}^{b} \phi_\nu^{(n_\nu/2+a_y)-1} \exp\Big\{ -\frac{\phi_\nu}{2}\Big(s_\nu^2 + 2b_y + \hat{\boldsymbol{\beta}}_\nu^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu + \boldsymbol{\beta}_0^\top \mathbf{C}\boldsymbol{\beta}_0 - \\
&((\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu + \mathbf{C}\boldsymbol{\beta}_0)^\top ((\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})^{-1})^\top ((\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu + \mathbf{C}\boldsymbol{\beta}_0)\Big)\Big\} \\
&\propto \prod_{\nu=1}^{b} \phi_\nu^{(n_\nu/2+a_y)-1} \exp\Big\{ -\frac{\phi_\nu}{2}\Big(s_\nu^2 + 2b_y + \hat{\boldsymbol{\beta}}_\nu^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu + \boldsymbol{\beta}_0^\top \mathbf{C}\boldsymbol{\beta}_0 - \\
&\hat{\boldsymbol{\beta}}_\nu^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)^\top ((\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})^{-1})^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu - \boldsymbol{\beta}_0^\top \mathbf{C}^\top ((\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})^{-1})^\top \mathbf{C}\boldsymbol{\beta}_0 - \\
&2\boldsymbol{\beta}_0^\top \mathbf{C}^\top ((\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})^{-1})^\top (\mathbf{F}_\nu^\top \mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu\Big)\Big\}
\end{aligned}
$$

$$
\begin{aligned}
\propto & \prod_{\nu=1}^{b} \phi_\nu^{(n_\nu/2+a_y)-1} \exp\Big\{ -\frac{\phi_\nu}{2}\Big(s_\nu^2 + 2b_y + \hat{\boldsymbol{\beta}}_\nu^\top((\mathbf{F}_\nu^\top\mathbf{F}_\nu) - \\
& (\mathbf{F}_\nu^\top\mathbf{F}_\nu)^\top((\mathbf{F}_\nu^\top\mathbf{F}_\nu + \mathbf{C})^{-1})^\top(\mathbf{F}_\nu^\top\mathbf{F}_\nu))\hat{\boldsymbol{\beta}}_\nu + \boldsymbol{\beta}_0^\top(\mathbf{C} - \mathbf{C}^\top((\mathbf{F}_\nu^\top\mathbf{F}_\nu + \mathbf{C})^{-1})^\top\mathbf{C})\boldsymbol{\beta}_0 - \\
& 2\boldsymbol{\beta}_0^\top\mathbf{C}^\top((\mathbf{F}_\nu^\top\mathbf{F}_\nu + \mathbf{C})^{-1})^\top(\mathbf{F}_\nu^\top\mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu\Big)\Big\} \\
\propto & \prod_{\nu=1}^{b} \phi_\nu^{(n_\nu/2+a_y)-1} \exp\Big\{ -\frac{\phi_\nu}{2}\Big(s_\nu^2 + 2b_y + \hat{\boldsymbol{\beta}}_\nu^\top((\mathbf{F}_\nu^\top\mathbf{F}_\nu) - \\
& (\mathbf{F}_\nu^\top\mathbf{F}_\nu)(\mathbf{F}_\nu^\top\mathbf{F}_\nu + \mathbf{C})^{-1}(\mathbf{F}_\nu^\top\mathbf{F}_\nu))\hat{\boldsymbol{\beta}}_\nu + \boldsymbol{\beta}_0^\top(\mathbf{C} - \mathbf{C}(\mathbf{F}_\nu^\top\mathbf{F}_\nu + \mathbf{C})^{-1}\mathbf{C})\boldsymbol{\beta}_0 - \\
& 2\boldsymbol{\beta}_0^\top\mathbf{C}(\mathbf{F}_\nu^\top\mathbf{F}_\nu + \mathbf{C})^{-1}(\mathbf{F}_\nu^\top\mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu\Big)\Big\} \\
\propto & \prod_{\nu=1}^{b} \phi_\nu^{(n_\nu/2+a_y)-1} \exp\Big\{ -\frac{\phi_\nu}{2}\Big(s_\nu^2 + 2b_y + \hat{\boldsymbol{\beta}}_\nu^\top((\mathbf{F}_\nu^\top\mathbf{F}_\nu)^{-1} + \mathbf{C}^{-1})^{-1}\hat{\boldsymbol{\beta}}_\nu + \\
& \boldsymbol{\beta}_0^\top((\mathbf{F}_\nu^\top\mathbf{F}_\nu)^{-1} + \mathbf{C}^{-1})^{-1}\boldsymbol{\beta}_0 - 2\boldsymbol{\beta}_0^\top((\mathbf{F}_\nu^\top\mathbf{F}_\nu)^{-1} + \mathbf{C}^{-1})^{-1}\hat{\boldsymbol{\beta}}_\nu\Big)\Big\} \\
\propto & \prod_{\nu=1}^{b} \phi_\nu^{(n_\nu/2+a_y)-1} \exp\Big\{ -\frac{\phi_\nu}{2}\Big(s_\nu^2 + 2b_y + (\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}_\nu)^\top((\mathbf{F}_\nu^\top\mathbf{F}_\nu)^{-1} + \mathbf{C}^{-1})^{-1}(\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}_\nu)\Big)\Big\}.
\end{aligned}
$$

It can be shown that $\phi_\nu$ follows a Gamma distribution:

$$
\begin{aligned}
& \phi_\nu|\mathbf{x}, \mathbf{Q}, \boldsymbol{\beta}_0, \mathbf{C}, b_y, \boldsymbol{\rho}, \mathcal{T}, \mathbf{y}, \mathbf{F} \\
\sim & \; G\Big(\frac{n_\nu}{2} + a_y \;,\; b_y + \frac{1}{2}(s_\nu^2 + (\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}_\nu)^\top\mathbf{R}_\nu^{-1}(\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}_\nu))\Big), \qquad \text{(A.4)}
\end{aligned}
$$

where

$$
\mathbf{R}_\nu = ((\mathbf{F}_\nu^\top\mathbf{F}_\nu)^{-1} + \mathbf{C}^{-1}), \quad \hat{\boldsymbol{\beta}}_\nu = (\mathbf{F}_\nu^\top\mathbf{F}_\nu)^{-1}\mathbf{F}_\nu^\top\mathbf{v}_\nu,
$$

$$
\mathbf{v}_\nu = \mathbf{y}_\nu - \mathbf{K}_\nu\mathbf{x}, \quad s_\nu^2 = (\mathbf{v}_\nu - \mathbf{F}_\nu\hat{\boldsymbol{\beta}}_\nu)^\top(\mathbf{v}_\nu - \mathbf{F}_\nu\hat{\boldsymbol{\beta}}_\nu).
$$

The following matrix identity is used in the last two steps,

$$
(M + X^\top X)^{-1} = (X^\top X)^{-1} - (X^\top X)^{-1}(M^{-1} + (X^\top X)^{-1})^{-1}(X^\top X)^{-1}.
$$

## A.5 Conditional Posterior Distribution for $\boldsymbol{\beta}_\nu$

The conditional posterior distribution for $\boldsymbol{\beta}_\nu$ is obtained by conditioning on all other parameters:

$$
\begin{aligned}
& P(\{\boldsymbol{\beta}_\nu\}_{\nu=1}^b|\{\phi_\nu,\mathbf{Q}_\nu\}_{\nu=1}^b,\mathbf{x},\boldsymbol{\beta}_0,\mathbf{C},\boldsymbol{\rho},\mathcal{T},\mathbf{y},\mathbf{F}) \\
\propto\ & \prod_{\nu=1}^b \Big(\frac{\phi_\nu}{2\pi}\Big)^{n_\nu/2} \exp\Big\{-\frac{\phi_\nu}{2}(\mathbf{y}_\nu-\mathbf{F}_\nu\boldsymbol{\beta}_\nu-\mathbf{K}_\nu\mathbf{x})^\top(\mathbf{y}_\nu-\mathbf{F}_\nu\boldsymbol{\beta}_\nu-\mathbf{K}_\nu\mathbf{x})\Big\}\times \\
& \Big(\frac{\phi_\nu}{2\pi}\Big)^{(p+1)/2}|\mathbf{C}|^{1/2}\exp\Big\{-\frac{\phi_\nu}{2}(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)^\top\mathbf{C}(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)\Big\} \\
\propto\ & \prod_{\nu=1}^b \exp\Big\{-\frac{\phi_\nu}{2}\Big(s_\nu^2+(\boldsymbol{\beta}_\nu-\hat{\boldsymbol{\beta}}_\nu)^\top(\mathbf{F}_\nu^\top\mathbf{F}_\nu)(\boldsymbol{\beta}_\nu-\hat{\boldsymbol{\beta}}_\nu)+(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)^\top\mathbf{C}(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)\Big)\Big\} \\
\propto\ & \prod_{\nu=1}^b \exp\Big\{-\frac{\phi_\nu}{2}\Big(\boldsymbol{\beta}_\nu^\top(\mathbf{F}_\nu^\top\mathbf{F}_\nu)\boldsymbol{\beta}_\nu+\hat{\boldsymbol{\beta}}_\nu^\top(\mathbf{F}_\nu^\top\mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu-2\boldsymbol{\beta}_\nu^\top(\mathbf{F}_\nu^\top\mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu+ \\
& \boldsymbol{\beta}_\nu^\top\mathbf{C}\boldsymbol{\beta}_\nu+\boldsymbol{\beta}_0^\top\mathbf{C}\boldsymbol{\beta}_0-2\boldsymbol{\beta}_\nu^\top\mathbf{C}\boldsymbol{\beta}_0\Big)\Big\} \\
\propto\ & \prod_{\nu=1}^b \exp\Big\{-\frac{\phi_\nu}{2}\Big(\boldsymbol{\beta}_\nu^\top(\mathbf{F}_\nu^\top\mathbf{F}_\nu+\mathbf{C})\boldsymbol{\beta}_\nu-2\boldsymbol{\beta}_\nu^\top(\mathbf{F}_\nu^\top\mathbf{F}_\nu+\mathbf{C})(\mathbf{F}_\nu^\top\mathbf{F}_\nu+\mathbf{C})^{-1}\times \\
& ((\mathbf{F}_\nu^\top\mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu+\mathbf{C}\boldsymbol{\beta}_0)\Big)\Big\} \\
\propto\ & \prod_{\nu=1}^b \exp\Big\{-\frac{1}{2}\Big(\boldsymbol{\beta}_\nu-(\mathbf{F}_\nu^\top\mathbf{F}_\nu+\mathbf{C})^{-1}((\mathbf{F}_\nu^\top\mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu+\mathbf{C}\boldsymbol{\beta}_0)\Big)^\top\Big(\phi_\nu(\mathbf{F}_\nu^\top\mathbf{F}_\nu+\mathbf{C})\Big)\times \\
& \Big(\boldsymbol{\beta}_\nu-(\mathbf{F}_\nu^\top\mathbf{F}_\nu+\mathbf{C})^{-1}((\mathbf{F}_\nu^\top\mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu+\mathbf{C}\boldsymbol{\beta}_0)\Big)\Big\}.
\end{aligned}
$$

It can be shown that $\boldsymbol{\beta}_\nu$ follows a multivariate Gaussian distribution:

$$
\begin{aligned}
& \boldsymbol{\beta}_\nu|\phi_\nu,\mathbf{Q}_\nu,\mathbf{x},\boldsymbol{\beta}_0,\mathbf{C},\boldsymbol{\rho},\mathcal{T},\mathbf{y},\mathbf{F} \\
\sim\ & N_{p+1}\Big((\mathbf{F}_\nu^\top\mathbf{F}_\nu+\mathbf{C})^{-1}((\mathbf{F}_\nu^\top\mathbf{F}_\nu)\hat{\boldsymbol{\beta}}_\nu+\mathbf{C}\boldsymbol{\beta}_0)\ ,\ (\phi_\nu(\mathbf{F}_\nu^\top\mathbf{F}_\nu+\mathbf{C}))^{-1}\Big), \\
\sim\ & N_{p+1}\Big((\mathbf{F}_\nu^\top\mathbf{F}_\nu+\mathbf{C})^{-1}(\mathbf{F}_\nu^\top\mathbf{v}_\nu+\mathbf{C}\boldsymbol{\beta}_0)\ ,\ (\phi_\nu(\mathbf{F}_\nu^\top\mathbf{F}_\nu+\mathbf{C}))^{-1}\Big). \qquad \text{(A.5)}
\end{aligned}
$$

When all partitions have a single $\boldsymbol{\beta}_\nu$, i.e., $\boldsymbol{\beta}_\nu = \boldsymbol{\beta}$ for $\nu = 1, \cdots, b$, and given the prior $\boldsymbol{\beta}|\boldsymbol{\beta}_0, \mathbf{C} \sim N_{p+1}(\boldsymbol{\beta}_0\,, \mathbf{C}^{-1})$, the conditional posterior distribution for $\boldsymbol{\beta}$ is given by

$$
\begin{aligned}
& P(\boldsymbol{\beta}|\{\phi_\nu, \mathbf{Q}_\nu\}_{\nu=1}^b, \mathbf{x}, \boldsymbol{\beta}_0, \mathbf{C}, \boldsymbol{\rho}, \mathcal{T}, \mathbf{y}, \mathbf{F}) \\
\propto\ & \prod_{\nu=1}^{b} \Big(\frac{\phi_\nu}{2\pi}\Big)^{n_\nu/2} \exp\Big\{ -\frac{\phi_\nu}{2}(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta} - \mathbf{K}_\nu\mathbf{x})^\top(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta} - \mathbf{K}_\nu\mathbf{x})\Big\} \times \\
& \Big(\frac{1}{2\pi}\Big)^{(p+1)/2} |\mathbf{C}|^{1/2} \exp\Big\{ -\frac{1}{2}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top \mathbf{C}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\Big\} \\
\propto\ & \exp\Big\{ -\frac{1}{2}\Big((\mathbf{y} - \mathbf{F}\boldsymbol{\beta} - \mathbf{K}\mathbf{x})^\top \boldsymbol{\Phi}(\mathbf{y} - \mathbf{F}\boldsymbol{\beta} - \mathbf{K}\mathbf{x}) + (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top \mathbf{C}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\Big)\Big\} \\
\propto\ & \exp\Big\{ -\frac{1}{2}\Big((\mathbf{v} - \mathbf{F}\boldsymbol{\beta})^\top \boldsymbol{\Phi}(\mathbf{v} - \mathbf{F}\boldsymbol{\beta}) + (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top \mathbf{C}(\boldsymbol{\beta} - \boldsymbol{\beta}_0)\Big)\Big\}.
\end{aligned}
$$

It can be shown that $\boldsymbol{\beta}$ follows a multivariate Gaussian distribution:

$$
\begin{aligned}
& \boldsymbol{\beta}|\{\phi_\nu, \mathbf{Q}_\nu\}_{\nu=1}^b, \mathbf{x}, \boldsymbol{\beta}_0, \mathbf{C}, \boldsymbol{\rho}, \mathcal{T}, \mathbf{y}, \mathbf{F} \\
\sim\ & N_{p+1}\Big((\mathbf{F}^\top\boldsymbol{\Phi}\mathbf{F} + \mathbf{C})^{-1}(\mathbf{F}^\top\boldsymbol{\Phi}\mathbf{v} + \mathbf{C}\boldsymbol{\beta}_0)\ ,\ (\mathbf{F}^\top\boldsymbol{\Phi}\mathbf{F} + \mathbf{C})^{-1}\Big).
\end{aligned}
$$

## A.6 Conditional Posterior Distribution for $\mathbf{Q}_\nu$

$$
\begin{aligned}
& P(\mathbf{Q}_\nu|\{\boldsymbol{\beta}_\nu, \phi_\nu\}_{\nu=1}^b, \mathbf{x}, \boldsymbol{\rho}, \mathcal{T}, \mathbf{y}, \mathbf{F}) \\
\propto\ & \prod_{\nu=1}^{b} \exp\Big\{ -\frac{\phi_\nu}{2}(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})^\top(\mathbf{y}_\nu - \mathbf{F}_\nu\boldsymbol{\beta}_\nu - \mathbf{K}_\nu\mathbf{x})\Big\} \times \\
& |\mathbf{Q}_\nu|^{(\psi-r-1)/2} \exp\Big\{ -\frac{1}{2}\mathrm{tr}(\psi\mathbf{H}\mathbf{Q}_\nu)\Big\} \\
\propto\ & \prod_{\nu=1}^{b} \exp\Big\{ -\frac{\phi_\nu}{2}\Big(\mathbf{x}^\top\mathbf{K}_\nu(\mathbf{Q})^\top\mathbf{K}_\nu(\mathbf{Q})\mathbf{x} - 2\mathbf{w}_\nu^\top\mathbf{K}_\nu(\mathbf{Q})\mathbf{x}\Big)\Big\} \times \\
& |\mathbf{Q}_\nu|^{(\psi-r-1)/2} \exp\Big\{ -\frac{1}{2}\mathrm{tr}(\psi\mathbf{H}\mathbf{Q}_\nu)\Big\} \\
\propto\ & \exp\Big\{ -\frac{1}{2}\Big(\mathbf{x}^\top\mathbf{K}(\mathbf{Q})^\top\boldsymbol{\Phi}\mathbf{K}(\mathbf{Q})\mathbf{x} - 2\mathbf{w}^\top\boldsymbol{\Phi}\mathbf{K}(\mathbf{Q})\mathbf{x}\Big)\Big\} \times |\mathbf{Q}_\nu|^{(\psi-r-1)/2} \exp\Big\{ -\frac{1}{2}\mathrm{tr}(\psi\mathbf{H}\mathbf{Q}_\nu)\Big\} \\
\propto\ & |\mathbf{Q}_\nu|^{(\psi-r-1)/2} \exp\Big\{ -\frac{1}{2}\Big(\mathbf{x}^\top\mathbf{K}(\mathbf{Q})^\top\boldsymbol{\Phi}\mathbf{K}(\mathbf{Q})\mathbf{x} - 2\mathbf{w}^\top\boldsymbol{\Phi}\mathbf{K}(\mathbf{Q})\mathbf{x} + \mathrm{tr}(\psi\mathbf{H}\mathbf{Q}_\nu)\Big)\Big\}.
\end{aligned}
$$

## A.7 Conditional Posterior Distribution for $\boldsymbol{\beta}_0$

$$
\begin{aligned}
& P(\boldsymbol{\beta}_0|\{\boldsymbol{\beta}_\nu,\phi_\nu\}_{\nu=1}^b,\mathbf{C},\boldsymbol{\rho},\mathcal{T}) \\
\propto\ & \prod_{\nu=1}^{b}\left(\exp\Big\{-\frac{\phi_\nu}{2}(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)^\top\mathbf{C}(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)\Big\}\right)\times\exp\Big\{-\frac{1}{2}(\boldsymbol{\beta}_0-\boldsymbol{\mu})^\top\mathbf{B}(\boldsymbol{\beta}_0-\boldsymbol{\mu})\Big\} \\
\propto\ & \prod_{\nu=1}^{b}\left(\exp\Big\{-\frac{1}{2}\Big(\boldsymbol{\beta}_0^\top(\phi_\nu\mathbf{C})\boldsymbol{\beta}_0-2\boldsymbol{\beta}_0^\top(\phi_\nu\mathbf{C})\boldsymbol{\beta}_\nu\Big)\Big\}\right)\times\exp\Big\{-\frac{1}{2}(\boldsymbol{\beta}_0^\top\mathbf{B}\boldsymbol{\beta}_0-2\boldsymbol{\beta}_0^\top\mathbf{B}\boldsymbol{\mu})\Big\} \\
\propto\ & \exp\Big\{-\frac{1}{2}\Big(\boldsymbol{\beta}_0^\top\Big(\mathbf{C}\sum_{\nu=1}^{b}\phi_\nu\Big)\boldsymbol{\beta}_0-2\boldsymbol{\beta}_0^\top\Big(\mathbf{C}\sum_{\nu=1}^{b}\phi_\nu\boldsymbol{\beta}_\nu\Big)+\boldsymbol{\beta}_0^\top\mathbf{B}\boldsymbol{\beta}_0-2\boldsymbol{\beta}_0^\top\mathbf{B}\boldsymbol{\mu}\Big)\Big\} \\
\propto\ & \exp\Big\{-\frac{1}{2}\Big(\boldsymbol{\beta}_0^\top\Big(\mathbf{C}\sum_{\nu=1}^{b}\phi_\nu+\mathbf{B}\Big)\boldsymbol{\beta}_0-2\boldsymbol{\beta}_0^\top\Big(\mathbf{C}\sum_{\nu=1}^{b}\phi_\nu\boldsymbol{\beta}_\nu+\mathbf{B}\boldsymbol{\mu}\Big)\Big\}.
\end{aligned}
$$

It can be shown that $\boldsymbol{\beta}_0$ follows a multivariate Gaussian distribution:

$$
\begin{aligned}
& \boldsymbol{\beta}_0|\{\boldsymbol{\beta}_\nu,\phi_\nu\}_{\nu=1}^b,\mathbf{C},\boldsymbol{\rho},\mathcal{T} \\
\sim\ & N_{p+1}\Big(\Big(\mathbf{C}\sum_{\nu=1}^{b}\phi_\nu+\mathbf{B}\Big)^{-1}\Big(\mathbf{C}\sum_{\nu=1}^{b}\phi_\nu\boldsymbol{\beta}_\nu+\mathbf{B}\boldsymbol{\mu}\Big)\ ,\ \Big(\mathbf{C}\sum_{\nu=1}^{b}\phi_\nu+\mathbf{B}\Big)^{-1}\Big).
\end{aligned}
$$

When $\boldsymbol{\beta}_\nu=\boldsymbol{\beta}$ for $\nu=1,\cdots,b$, it can be shown that

$$
\boldsymbol{\beta}_0|\boldsymbol{\beta},\mathbf{C},\boldsymbol{\rho},\mathcal{T}\sim N_{p+1}\Big((\mathbf{C}+\mathbf{B})^{-1}(\mathbf{C}\boldsymbol{\beta}+\mathbf{B}\boldsymbol{\mu})\ ,\ (\mathbf{C}+\mathbf{B})^{-1}\Big),
$$

which is based on this prior: $\boldsymbol{\beta}_c|\boldsymbol{\beta}_0,\mathbf{C}\sim N_{p+1}(\boldsymbol{\beta}_0\,,\mathbf{C}^{-1})$.

## A.8 Conditional Posterior Distribution for C

$$
\begin{aligned}
& P(\mathbf{C}|\{\boldsymbol{\beta}_\nu,\phi_\nu\}_{\nu=1}^b,\boldsymbol{\rho},\mathcal{T}) \\
\propto\ & \Big(\prod_{\nu=1}^b |\mathbf{C}|^{1/2}\exp\Big\{-\frac{\phi_\nu}{2}(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)^\top\mathbf{C}(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)\Big\}\Big)|\mathbf{C}|^{(\varphi-p)/2}\exp\Big\{-\frac{1}{2}\mathrm{tr}(\varphi\mathbf{V}\mathbf{C})\Big\} \\
\propto\ & |\mathbf{C}|^{(\varphi+b-p)/2}\exp\Big\{-\frac{1}{2}\sum_{\nu=1}^b\phi_\nu(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)^\top\mathbf{C}(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)\Big\}\exp\Big\{-\frac{1}{2}\mathrm{tr}(\varphi\mathbf{V}\mathbf{C})\Big\} \\
\propto\ & |\mathbf{C}|^{(\varphi+b-p)/2}\exp\Big\{-\frac{1}{2}\mathrm{tr}\Big(\sum_{\nu=1}^b\phi_\nu(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)^\top\mathbf{C}\Big)\Big\}\exp\Big\{-\frac{1}{2}\mathrm{tr}(\varphi\mathbf{V}\mathbf{C})\Big\} \\
\propto\ & |\mathbf{C}|^{(\varphi+b-p)/2}\exp\Big\{-\frac{1}{2}\mathrm{tr}\Big(\sum_{\nu=1}^b\phi_\nu(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)^\top\mathbf{C}+\varphi\mathbf{V}\mathbf{C}\Big)\Big\} \\
\propto\ & |\mathbf{C}|^{(\varphi+b-p)/2}\exp\Big\{-\frac{1}{2}\mathrm{tr}\Big(\Big(\sum_{\nu=1}^b\phi_\nu(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)^\top+\varphi\mathbf{V}\Big)\mathbf{C}\Big)\Big\}.
\end{aligned}
$$

It can be shown that $\mathbf{C}$ follows a Wishart distribution:

$$
\mathbf{C}|\{\boldsymbol{\beta}_\nu,\phi_\nu\}_{\nu=1}^b,\boldsymbol{\rho},\mathcal{T}\sim W\Big(\Big(\sum_{\nu=1}^b\phi_\nu(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)(\boldsymbol{\beta}_\nu-\boldsymbol{\beta}_0)^\top+\varphi\mathbf{V}\Big)^{-1}\ ,\ \varphi+b\Big).
$$

When $\boldsymbol{\beta}_\nu=\boldsymbol{\beta}$ for $\nu=1,\cdots,b$,

$$
\mathbf{C}|\boldsymbol{\beta},\boldsymbol{\rho},\mathcal{T}\sim W\Big(\Big((\boldsymbol{\beta}-\boldsymbol{\beta}_0)(\boldsymbol{\beta}-\boldsymbol{\beta}_0)^\top+\varphi\mathbf{V}\Big)^{-1}\ ,\ \varphi+1\Big).
$$

which is based on this prior: $\boldsymbol{\beta}|\boldsymbol{\beta}_0,\mathbf{C}\sim N_{p+1}(\boldsymbol{\beta}_0\,,\mathbf{C}^{-1})$.

## A.9 Conditional Posterior Distribution for $b_x$ and $b_y$

$$
b_x|\{\lambda_\nu\}_{\nu=1}^b,\boldsymbol{\rho},\mathcal{T}\sim G\Big(a_xb+\tau_x\ ,\ \sum_{\nu=1}^b\lambda_\nu+\xi_x\Big),
$$

$$
b_y|\{\phi_\nu\}_{\nu=1}^b,\boldsymbol{\rho},\mathcal{T}\sim G\Big(a_yb+\tau_y\ ,\ \sum_{\nu=1}^b\phi_\nu+\xi_y\Big).
$$

## A.10 Conditional posterior distribution for $(\mathcal{T}, \boldsymbol{\rho})$

$$
\begin{aligned}
& P(\mathcal{T}, \boldsymbol{\rho} | \{\mathbf{Q}_\nu\}_{\nu=1}^b, \mathbf{x}, \boldsymbol{\beta}_0, \mathbf{C}, b_x, b_y, \mathbf{y}, \mathbf{F}) \\
\propto\ & \prod_{\nu=1}^{b} \Bigg( \int \Big(\frac{\phi_\nu}{2\pi}\Big)^{n_\nu/2} \exp\Big\{ -\frac{\phi_\nu}{2} (\mathbf{y}_\nu - \mathbf{F}_\nu \boldsymbol{\beta}_\nu - \mathbf{K}_\nu \mathbf{x})^\top (\mathbf{y}_\nu - \mathbf{F}_\nu \boldsymbol{\beta}_\nu - \mathbf{K}_\nu \mathbf{x}) \Big\} \times \\
& \Big(\frac{\phi_\nu}{2\pi}\Big)^{(p+1)/2} |\mathbf{C}|^{1/2} \exp\Big\{ -\frac{\phi_\nu}{2} (\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0)^\top \mathbf{C} (\boldsymbol{\beta}_\nu - \boldsymbol{\beta}_0) \Big\} \times \\
& \Big(\frac{\lambda_\nu}{2\pi}\Big)^{m_\nu/2} \exp\Big\{ -\frac{\lambda_\nu}{2} \mathbf{x}_\nu^\top \mathbf{x}_\nu \Big\} \times \frac{b_y^{a_y}}{\Gamma(a_y)} \phi_\nu^{a_y - 1} \exp\{-b_y \phi_\nu\} \times \frac{b_x^{a_x}}{\Gamma(a_x)} \lambda_\nu^{a_x - 1} \exp\{-b_x \lambda_\nu\} \times \\
& \frac{|\mathbf{Q}_\nu|^{(\psi - r - 1)/2} |\psi \mathbf{H}|^{\psi/2}}{2^{\psi r/2} \Gamma_r(\psi/2)} \exp\Big\{ -\frac{1}{2} \mathrm{tr}(\psi \mathbf{H} \mathbf{Q}_\nu) \Big\} d\boldsymbol{\beta}_\nu d\phi_\nu d\lambda_\nu \Bigg) \times \\
& \Big(\frac{1}{2\pi}\Big)^{(p+1)/2} |\mathbf{B}|^{1/2} \exp\Big\{ -\frac{1}{2} (\boldsymbol{\beta}_0 - \boldsymbol{\mu})^\top \mathbf{B} (\boldsymbol{\beta}_0 - \boldsymbol{\mu}) \Big\} \times \\
& \frac{|\mathbf{C}|^{(\varphi - p)/2} |\varphi \mathbf{V}|^{\varphi/2}}{2^{\varphi(p+1)/2} \Gamma_{p+1}(\varphi/2)} \exp\Big\{ -\frac{1}{2} \mathrm{tr}(\varphi \mathbf{V} \mathbf{C}) \Big\} \times \\
& \frac{\xi_x^{\tau_x}}{\Gamma(\tau_x)} b_x^{\tau_x - 1} \exp\{-\xi_x b_x\} \times \frac{\xi_y^{\tau_y}}{\Gamma(\tau_y)} b_y^{\tau_y - 1} \exp\{-\xi_y b_y\} \times P(\boldsymbol{\rho} | \mathcal{T}) P(\mathcal{T}) \\
\propto\ & \prod_{\nu=1}^{b} \Bigg( \int \Big(\frac{1}{2\pi}\Big)^{(n_\nu + m_\nu + p + 1)/2} (2\pi)^{(p+1)/2} |(\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})^{-1}|^{1/2} \frac{b_y^{a_y}}{\Gamma(a_y)} \frac{b_x^{a_x}}{\Gamma(a_x)} \times \\
& \phi_\nu^{(n_\nu/2 + a_y) - 1} \exp\Big\{ -\frac{\phi_\nu}{2} \Big( s_\nu^2 + 2 b_y + (\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}_\nu)^\top ((\mathbf{F}_\nu^\top \mathbf{F}_\nu)^{-1} + \mathbf{C}^{-1})^{-1} (\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}_\nu) \Big) \Big\} \times \\
& \lambda_\nu^{(m_\nu/2 + a_x) - 1} \exp\Big\{ -\frac{\lambda_\nu}{2} \Big( \mathbf{x}_\nu^\top \mathbf{x}_\nu + 2 b_x \Big) \Big\} \times \\
& \frac{|\mathbf{Q}_\nu|^{(\psi - r - 1)/2} |\psi \mathbf{H}|^{\psi/2}}{2^{\psi r/2} \Gamma_r(\psi/2)} \exp\Big\{ -\frac{1}{2} \mathrm{tr}(\psi \mathbf{H} \mathbf{Q}_\nu) \Big\} d\phi_\nu d\lambda_\nu \Bigg) \times P(\boldsymbol{\rho} | \mathcal{T}) P(\mathcal{T}) \\
\propto\ & \Bigg( \prod_{\nu=1}^{b} \Big(\frac{1}{2\pi}\Big)^{(n_\nu + m_\nu)/2} |(\mathbf{F}_\nu^\top \mathbf{F}_\nu + \mathbf{C})|^{-1/2} \frac{b_y^{a_y}}{\Gamma(a_y)} \frac{b_x^{a_x}}{\Gamma(a_x)} \times \\
& \Gamma(n_\nu/2 + a_y) \Big( b_y + \frac{1}{2} \Big( s_\nu^2 + (\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}_\nu)^\top ((\mathbf{F}_\nu^\top \mathbf{F}_\nu)^{-1} + \mathbf{C}^{-1})^{-1} (\boldsymbol{\beta}_0 - \hat{\boldsymbol{\beta}}_\nu) \Big) \Big)^{-(n_\nu/2 + a_y)} \times \\
& \Gamma(m_\nu/2 + a_x) \Big( \frac{1}{2} \mathbf{x}_\nu^\top \mathbf{x}_\nu + b_x \Big)^{-(m_\nu/2 + a_x)} \times \\
& \frac{|\mathbf{Q}_\nu|^{(\psi - r - 1)/2} |\psi \mathbf{H}|^{\psi/2}}{2^{\psi r/2} \Gamma_r(\psi/2)} \exp\Big\{ -\frac{1}{2} \mathrm{tr}(\psi \mathbf{H} \mathbf{Q}_\nu) \Big\} \Bigg) P(\boldsymbol{\rho} | \mathcal{T}) P(\mathcal{T}).
\end{aligned}
$$

When $\boldsymbol{\beta}_\nu = \boldsymbol{\beta}$ for $\nu = 1, \cdots, b$, only $\phi_\nu$ and $\lambda_\nu$ are integrated out from the joint posterior:

$$
\begin{aligned}
& P(\mathcal{T}, \boldsymbol{\rho} | \{\mathbf{Q}_\nu\}_{\nu=1}^b, \boldsymbol{\beta}, \mathbf{x}, \boldsymbol{\beta}_0, \mathbf{C}, b_y, b_x, \mathbf{y}, \mathbf{F}) \\
\propto \; & \prod_{\nu=1}^{b} \Bigg( \int \Big(\frac{\phi_\nu}{2\pi}\Big)^{n_\nu/2} \exp\Big\{ -\frac{\phi_\nu}{2} (\mathbf{y}_\nu - \mathbf{F}_\nu \boldsymbol{\beta} - \mathbf{K}_\nu \mathbf{x})^\top (\mathbf{y}_\nu - \mathbf{F}_\nu \boldsymbol{\beta} - \mathbf{K}_\nu \mathbf{x}) \Big\} \times \\
& \Big(\frac{\lambda_\nu}{2\pi}\Big)^{m_\nu/2} \exp\Big\{ -\frac{\lambda_\nu}{2} \mathbf{x}_\nu^\top \mathbf{x}_\nu \Big\} \times \frac{b_y^{a_y}}{\Gamma(a_y)} \phi_\nu^{a_y - 1} \exp\{-b_y \phi_\nu\} \times \frac{b_x^{a_x}}{\Gamma(a_x)} \lambda_\nu^{a_x - 1} \exp\{-b_x \lambda_\nu\} \times \\
& \frac{|\mathbf{Q}_\nu|^{(\psi - r - 1)/2} |\psi \mathbf{H}|^{\psi/2}}{2^{\psi r/2} \Gamma_r(\psi/2)} \exp\Big\{ -\frac{1}{2} \mathrm{tr}(\psi \mathbf{H} \mathbf{Q}_\nu) \Big\} d\phi_\nu d\lambda_\nu \Bigg) \times \\
& \Big(\frac{1}{2\pi}\Big)^{(p+1)/2} |\mathbf{C}|^{1/2} \exp\Big\{ -\frac{1}{2} (\boldsymbol{\beta} - \boldsymbol{\beta}_0)^\top \mathbf{C} (\boldsymbol{\beta} - \boldsymbol{\beta}_0) \Big\} \times \\
& \Big(\frac{1}{2\pi}\Big)^{(p+1)/2} |\mathbf{B}|^{1/2} \exp\Big\{ -\frac{1}{2} (\boldsymbol{\beta}_0 - \boldsymbol{\mu})^\top \mathbf{B} (\boldsymbol{\beta}_0 - \boldsymbol{\mu}) \Big\} \times \\
& \frac{|\mathbf{C}|^{(\varphi - p)/2} |\varphi \mathbf{V}|^{\varphi/2}}{2^{\varphi(p+1)/2} \Gamma_{p+1}(\varphi/2)} \exp\Big\{ -\frac{1}{2} \mathrm{tr}(\varphi \mathbf{V} \mathbf{C}) \Big\} \times \\
& \frac{\xi_x^{\tau_x}}{\Gamma(\tau_x)} b_x^{\tau_x - 1} \exp\{-\xi_x b_x\} \times \frac{\xi_y^{\tau_y}}{\Gamma(\tau_y)} b_y^{\tau_y - 1} \exp\{-\xi_y b_y\} \times P(\boldsymbol{\rho} | \mathcal{T}) P(\mathcal{T}) \\
\propto \; & \prod_{\nu=1}^{b} \Bigg( \Big(\frac{1}{2\pi}\Big)^{(n_\nu + m_\nu)/2} \frac{b_y^{a_y}}{\Gamma(a_y)} \frac{b_x^{a_x}}{\Gamma(a_x)} \times \\
& \phi_\nu^{(n_\nu/2 + a_y) - 1} \exp\Big\{ -\frac{\phi_\nu}{2} \Big( (\mathbf{y}_\nu - \mathbf{F}_\nu \boldsymbol{\beta} - \mathbf{K}_\nu \mathbf{x})^\top (\mathbf{y}_\nu - \mathbf{F}_\nu \boldsymbol{\beta} - \mathbf{K}_\nu \mathbf{x}) + 2 b_y \Big) \Big\} \times \\
& \lambda_\nu^{(m_\nu/2 + a_x) - 1} \exp\Big\{ -\frac{\lambda_\nu}{2} \Big( \mathbf{x}_\nu^\top \mathbf{x}_\nu + 2 b_x \Big) \Big\} \times \\
& \frac{|\mathbf{Q}_\nu|^{(\psi - r - 1)/2} |\psi \mathbf{H}|^{\psi/2}}{2^{\psi r/2} \Gamma_r(\psi/2)} \exp\Big\{ -\frac{1}{2} \mathrm{tr}(\psi \mathbf{H} \mathbf{Q}_\nu) \Big\} d\phi_\nu d\lambda_\nu \Bigg) P(\boldsymbol{\rho} | \mathcal{T}) P(\mathcal{T}) \\
\propto \; & \Bigg( \prod_{\nu=1}^{b} \Big(\frac{1}{2\pi}\Big)^{(n_\nu + m_\nu)/2} \frac{b_y^{a_y}}{\Gamma(a_y)} \frac{b_x^{a_x}}{\Gamma(a_x)} \times \\
& \Gamma(n_\nu/2 + a_y) \Big( \frac{1}{2} (\mathbf{y}_\nu - \mathbf{F}_\nu \boldsymbol{\beta} - \mathbf{K}_\nu \mathbf{x})^\top (\mathbf{y}_\nu - \mathbf{F}_\nu \boldsymbol{\beta} - \mathbf{K}_\nu \mathbf{x}) + b_y \Big)^{-(n_\nu/2 + a_y)} \times \\
& \Gamma(m_\nu/2 + a_x) \Big( \frac{1}{2} \mathbf{x}_\nu^\top \mathbf{x}_\nu + b_x \Big)^{-(m_\nu/2 + a_x)} \times \\
& \frac{|\mathbf{Q}_\nu|^{(\psi - r - 1)/2} |\psi \mathbf{H}|^{\psi/2}}{2^{\psi r/2} \Gamma_r(\psi/2)} \exp\Big\{ -\frac{1}{2} \mathrm{tr}(\psi \mathbf{H} \mathbf{Q}_\nu) \Big\} \Bigg) P(\boldsymbol{\rho} | \mathcal{T}) P(\mathcal{T}).
\end{aligned}
$$

# Bibliography

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2003), *Hierarchical Modeling and Analysis for Spatial Data*, Chapman & Hall, Boca Raton, FL.

Banerjee, S., Gelfand, A. E., Finley, A. O., and Sang, H. (2008), "Gaussian predictive process models for large spatial data sets," *Journal of the Royal Statistical Society: Series B*, 70(4), 825–848.

Barry, R. P. and Ver Hoef, J. M. (1996), "Blackbox Kriging: Spatial prediction without specifying variogram models," *Journal of Agricultural, Biological, and Environmental Statistics*, 1, 297–322.

Breiman, L., Friedman, J. H., Olshen, R., and Stone, C. (1984), "Classification and Regression Trees," *Belmont CA: Wadsworth.*

Brenning, A. (2001), "Geostatistics without stationarity assumptions within geographical information systems," *Freiberg Online Geoscience*, 6, 1–108.

Calder, C. A., Holloman, C., and Higdon, D. (2002), "Exploring Space-Time Structure

in Ozone Concentration Using a Dynamic Process Convolution Model," *Case Studies in Bayesian Statistics*, 6, 165–176.

Carvalho, C. M., Johannes, M., Lopes, H. F., and Polson, N. G. (2010), "Particle Learning and Smoothing," *Statistical Science*, 25(1), 88–106.

Chipman, H. A., George, E. I., and McCulloch, R. E. (1998), "Bayesian CART Model Search," *Journal of the American Statistical Association*, 93, 935–948.

Chipman, H. A., George, E. I., and McCulloch, R. E. (2002), "Bayesian Treed Models," *Machine Learning*, 48, 303–324.

Cressie, N. (1991), *Statistics for Spatial Data*, John Wiley and Sons, Inc.

Damian, D., Sampson, P., and Guttorp, P. (2001), "Bayesian estimation of semi-parametric non-stationary spatial covariance structure," *Environmetrics*, 12, 161–178.

Denison, D., Mallick, B., and Smith, A. F. M. (1998), "A Bayesian CART algorithm," *Biometrika*, 85, 363–377.

Duan, A., Guindani, M., and Gelfand, A. E. (2007), "Generalized Spatial Dirichlet Process Models," *Biometrika*, 94 (4), 809–825.

Friedman, J. H. (1991), "Multivariate Adaptive Regression Splines," *Annals of Statistics*, 19 (1), 1–67.

Fuentes, M. and Smith, R. L. (2001), "A new class of nonstationary spatial models," Tech. rep., North Carolina State University, Department of Statistics, Raleigh, NC.

Furrer, R. (2006), "An R Package for Covariance Tapered Kriging of Large Datasets Using Sparse Matrix Techniques," .

Furrer, R., Genton, M. G., and Nychka, D. (2006), "Covariance Tapering for Interpolation of Large Spatial Datasets," *Journal of Computational and Graphical Statistics*, 15 (3), 502–523.

Gelfand, A. E., Kottas, A., and MacEachern, S. N. (2005), "Bayesian Nonparametric Spatial Modeling With Dirichlet Process Mixing," *Journal of the American Statistical Association*, 100 (471), 1021–1035.

Gelman, A., Carlin, J. B., Stern, H. S., and Rubin, D. B. (2004), *Bayesian Data Analysis*, Chapman & Hall.

Geman, S. and Geman, D. (1984), "Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12, 609–628.

Gordon, N., Salmond, D., and Smith, A. (1993), "Novel approach to nonlinear/non-Gaussian Bayesian state estimation," *Radar and Signal Processing, IEEE Proceedings*, F140, 107–113.

Gramacy, R. B. (2005), "Bayesian Treed Gaussian Process Models," Ph.D. thesis, Department of AMS, UCSC, Santa Cruz, 95060.

Gramacy, R. B. (2007), "tgp: An R Package for Bayesian Nonstationary, Semiparamet-

ric Nonlinear Regression and Design by Treed Gaussian Process Models," *Journal of Statistical Software*, 19, 1–46.

Gramacy, R. B. (2010), "R Package plgp: Particle Learning of Gaussian Processes," .

Gramacy, R. B. and Lee, H. K. (2008), "Bayesian Treed Gaussian Process Models with an Application to Computer Modeling," *Journal of the American Statistical Association*, 103(483), 1119–1130.

Gramacy, R. B. and Lee, H. K. (2009), "Adaptive Design and Analyis of Supercomputer Experiments," *Technometrics*, 51, 130–145.

Gramacy, R. B. and Lee, H. K. H. (2011), "Optimization under unknown constraints," in *Bayesian Statistics 9*, eds. J. Bernardo, S. Bayarri, J. Berger, A. Dawid, D. Heckerman, A. Smith, and M. West, pp. 229–256, Oxford University Press.

Gramacy, R. B. and Polson, N. G. (2009), "Particle Learning of Gaussian Process Models for Sequential Design and Optimization," *Journal of Computational and Graphical Statistics*, 20, 102–118.

Green, P. J. (1995), "Reversible jump Markov chain Monte Carlo computation and Bayesian model determination," *Biometrika*, 82, 711–32.

Hastings, W. (1970), "Monte Carlo Sampling Methods Using Markov Chains and Their Applications," *Biometrika*, 57 (1), 97109.

Higdon, D. (1998), "A process-convolution approach to modeling temperatures in the

North Atlantic Ocean," *Journal of Environmental and Ecological Statistics*, 5(2), 173–190.

Higdon, D. (2002), "Space and space-time modeling using process convolutions," in *Quantitative Methods for Current Environmental Issues*, eds. C. Anderson, V. Barnett, P. Chatwin, and A. El-Shaarawi, pp. 37–54, London, Springer.

Higdon, D. (2005), "A Primer on Space-time Modeling from a Bayesian Perspective," Tech. Rep. LA-UR-05-3097, Statistical Sciences Group, Los Alamos National Laboratory.

Higdon, D., Swall, J., and Kern, J. (1999), "Non-stationary Spatial Modeling," in *Bayesian statistics 6*, p. 761768, Proceedings of the Sixth Valencia International Meeting.

Hoef, J. M. V., Cressie, N. A. C., and P., R. P. B. R. (2004), "Flexible spatial models based on the fast Fourier transform (FFT) for cokriging," *Computnl Graph. Statist.*, 13, 265–282.

Hoeting, J., Madigan, D., Raftery, A., and Volinsky, C. (1999), "Bayesian model averaging: A tutorial (with discussion)," *Statistical Science*, 14, 382417.

Johns, C. J., Nychka, D., Kittel, T. G., and Daly, C. (2003), "Infilling Sparse Records of Spatial Fields," *Journal of the American Statistical Association*, 98, 796–806.

Jones, D., Schonlau, M., and Welch, W. (1998), "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, 13, 455–492.

Kalman, R. E. (1960), "A new approach to linear filtering and prediction problems," *Transactions of the ASME-Journal of Basic Engineering*, 82, 35–45.

Kammann, E. E. and Wand, M. P. (2003), "Geoadditive models," *Appl. Statist.*, 52, 1–18.

Kass, R. and Raftery, A. (1995), "Bayes factors," *Journal of the American Statistical Association*, 90, 773795.

Kern, J. C. (2000), "Bayesian Process-convolution Approaches to Specifying Spatial Dependence Structure," Ph.D. thesis, Duke University, Durham, NC 27708.

Kim, H.-M., Mallick, B. K., and Holmes, C. C. (2005), "Analyzing nonstationary spatial data using piecewise Gaussian processes," *Journal of the American Statistical Association*, 100, 653–668.

Lee, H. K. H., Higdon, D., Calder, C. A., and Holloman, C. H. (2005), "Efficient Models for Correlated Data via Convolutions of Intrinsic Processes," *Statistical Modelling*, 5, 53–74.

Lemos, R. T. and Sansó, B. (2009), "Spatio-Temporal Model for Mean, Anomaly and Trend Fields of North Atlantic Sea Surface Temperature," *Journal of the American Statistical Association*, 104, 5–18.

Lin, X., Wahba, G., Xiang, D., Gao, F., Klein, R., and Klein, B. (2000), "Smoothing spline ANOVA models for large data sets with Bernoulli observations and the randomized GACV," *Ann. Statist.*, 28, 1570–1600.

Liu, J. and West, M. (2001), "Combined parameters and state estimation in simulation-based filtering," in *Sequential Monte Carlo Methods in Practice (Eds. A. Doucet, N. de Freitas and N. Gordon)*, pp. 197–223, Springer-Verlag, New York.

Matheron, G. (1963), "Principles of geostatistics," *Economic Geology*, 58, 1246–1266.

Matott, L. S., Leung, K., and Sim, J. (2011), "Application of Matlab and Python Optimizers to Two Case-Studies Involving Groundwater Flow and Contaminant Transport Modeling," *Geospatial Cyberinfrastructure for Polar Research*, 37 (11), 1894–1899.

McKay, M. D., Conover, W. J., and Beckman, R. J. (1979), "A Comparison of Three Methods for Selecting Values of Input Variables in the Analysis of Output from a Computer Code," *Technometrics*, 21, 239–245.

Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953), "Equations of State Calculations by Fast Computing Machines," *Journal of Chemical Physics*, 21 (6), 10871092.

Moran, P. A. P. (1950), "Notes on Continuous Stochastic Phenomena," *Biometrika*, 37, 17–23.

Neal, R. M. (1997), "Monte Carlo implementation of Gaussian process for Bayesian regression and classification," Tech. rep., Dept. of statistics, University of Toronto.

Neal, R. M. (1998), "Regression and classification using Gaussian process priors (with discussion)," *Bayesian statistics*, 6, 476–501.

Paciorek, C. and Schervish, M. J. (2006), "Spatial Modelling Using a New Class of Nonstationary Covariance Functions," *Environmetrics*, 17, 483–506.

Paciorek, C. J. (2003), "Nonstationary Gaussian Processes for Regression and Spatial Modelling," Ph.D. thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania.

Paciorek, C. J. (2007), "Computational techniques for spatial logistic regression with large datasets," *Computnl Statist. Data Anal.*, 51, 36313653.

Pitt, M. and Shephard, N. (1999), "Filtering via simulation: auxiliary particle lters," *Journal of the American Statistical Association*, 94, 590–599.

R Development Core Team (2011), *R: A language and environment for statistical computing*, R Foundation for Statistical Computing, Vienna, Austria, ISBN 3-900051-07-0.

Rasmussen, C. E. and Williams, C. K. I. (2006), *Gaussian Processes for Machine Learning*, The MIT Press.

Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P. (1989), "Design and Analysis of Computer Experiments," *Statistical Science*, 4, 409–435.

Sampson, P. and Guttorp, P. (1992), "Nonparametric Estimation of Nonstationary Spatial Covariance Structure," *Am. Stat. Assoc.*, 87, 108–119.

Schmidt, A. and O'Hagan, A. (2003), "Bayesian inference for non-stationary spatial covariance structure via spatial deformations," *Journal of the Royal Statistical Society: Series B*, 65, 743–758.

Stein, M. L. (1999), *Interpolation of Spatial Data*, Springer, New York, NY.

Storvik, G. (2002), "Particle lters in state space models with the presence of unknown static parameters," *IEEE Transactions of Signal Processing*, 50, 281–289.

Swall, J. L. (1999), "Nonstationary Spatial Modeling Using a Process Convolution Approach," Ph.D. thesis, Duke University, Durham, NC 27708.

Taddy, M. A., Lee, H. K. H., Gray, G. A., and Griffin, J. D. (2009), "Bayesian Guided Pattern Search for Robust Local Optimization," *Technometrics*, 51 (4), 389–401.

West, M. and Harrison, J. (1997), *Bayesian Forecasting and Dynamic Models*, Springer.

Wikle, C. and Cressie, N. (1999), "A dimension-reduced approach to space-time Kalman filtering," *Biometrika*, 86, 815–829.

Xia, G. and Gelfand, A. E. (2006), "Stationary process approximation for the analysis of large spatial datasets," Tech. rep., Institute of Statistics and Decision Sciences, Duke University, Durham.