

# UC Santa Cruz

## UC Santa Cruz Electronic Theses and Dissertations

### Title

Nonstationary Models for Large Spatial Datasets Using Multi-resolution Process Convolutions

### Permalink

<https://escholarship.org/uc/item/8x932794>

### Author

Kirsner, Daniel

### Publication Date

2020

### Copyright Information

This work is made available under the terms of a Creative Commons Attribution-NonCommercial-ShareAlike License, available at <https://creativecommons.org/licenses/by-nc-sa/4.0/>

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA  
SANTA CRUZ

**NONSTATIONARY MODELS FOR LARGE SPATIAL DATASETS  
USING MULTI-RESOLUTION PROCESS CONVOLUTIONS**

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

DOCTOR OF PHILOSOPHY

in

STATISTICAL SCIENCE

by

**Daniel Kirsner**

June 2020

The Dissertation of Daniel Kirsner  
is approved:

---

Bruno Sansó, Chair

---

Abel Rodriguez

---

Rajarshi Guhaniyogi

---

Quentin Williams  
Acting Vice Provost and Dean of Graduate Studies

Copyright © by

Daniel Kirsner

2020

# Table of Contents

List of Figures	v
List of Tables	viii
Abstract	x
Dedication	xii
Acknowledgments	xiii
<b>1 Introduction</b>	<b>1</b>
1.1 Background	1
1.1.1 Computational Efficiency and Spatial Models	2
1.1.2 Non-stationarity and Spatial Models	4
1.2 Discrete Process Convolutions	7
1.2.1 Domain Partitioning	8
1.2.2 Spatially Varying Shrinkage	9
1.2.3 Research Objectives	11
<b>2 Multi-scale Shotgun Stochastic Search</b>	<b>14</b>
2.1 Bayesian Model Selection	14
2.2 A Bayesian multi-resolution model	18
2.2.1 A prior that induces spatially varying resolution	19
2.2.2 Prior for the nonzero $\beta_j^r$ , $\alpha$ , and $\sigma^2$	22
2.2.3 Extending shotgun stochastic search	26
2.2.4 Computational details	28
2.2.5 Prediction and interval estimation	29
2.3 Assessing the proposed model	31
2.3.1 The datasets	32
2.3.2 Parameter settings and competitor details	34
2.3.3 Results	36
2.3.4 Default parameters	43



2.4	Discussion . . . . .	44
<b>3</b>	<b>Multi-Scale Spatial Optimization</b>	<b>45</b>
3.1	Introduction . . . . .	45
3.1.1	LASSO . . . . .	47
3.1.2	Group Lasso . . . . .	50
3.1.3	Composite Absolute Penalties . . . . .	51
3.1.4	Hierarchical Multiple Kernel Learning . . . . .	53
3.2	Multi-resolution Spatial Models . . . . .	55
3.2.1	Maximum A Posteriori Estimation . . . . .	57
3.2.2	Simulation Study . . . . .	60
3.3	Conclusion . . . . .	69
<b>4</b>	<b>From Optimization to Bayesian Model Averaging</b>	<b>70</b>
4.1	Introduction . . . . .	70
4.2	Model Averaging Along Optimization Paths . . . . .	72
4.3	Mediterranean Data . . . . .	75
4.4	Issues With Large Data . . . . .	76
4.5	Summary . . . . .	78
<b>5</b>	<b>MSSS: An R package for Fitting Surfaces With Spatially Varying Resolution</b>	<b>80</b>
5.1	Introduction . . . . .	80
5.2	Spatially Varying Resolution . . . . .	82
5.2.1	Kernel Functions . . . . .	83
5.2.2	Priors for $\alpha$ , the nonzero $\beta_j^r$ , and $\sigma^2$ . . . . .	84
5.3	Fitting the Models . . . . .	88
5.3.1	Multi-scale Shotgun Stochastic Search . . . . .	88
5.3.2	Multi-scale Spatial Optimization . . . . .	90
5.4	Synthetic Data Example . . . . .	91
5.4.1	Implementation Details . . . . .	98
5.5	Summary . . . . .	100
<b>6</b>	<b>Case Study: Prediction and Feature Identification via Spatially Varying Resolution</b>	<b>101</b>
6.1	Introduction . . . . .	101
6.2	Ozark Data . . . . .	103
6.3	Competitor Models . . . . .	103
6.4	MSSS Applied to the Ozark Data . . . . .	105
6.5	California Sea Surface Temperature Data . . . . .	111
6.6	Conclusion . . . . .	116
<b>7</b>	<b>Conclusion</b>	<b>118</b>

# List of Figures

1.1	On the left, a plot of knot locations for the first 3 resolutions in 1 dimension with $J(1) = 3$ . On the right, a plot of knot locations for the first 3 resolutions in 2 dimensions with $J(1) = 9$ . . . . .	9
2.1	Average shrinkage under the G prior at different resolutions with equally spaced data . . . . .	24
2.2	Sample locations for the 3 simulated 2D datasets . . . . .	33
2.3	December 2003 Mediterranean sea surface temperature observations in Celsius. . . . .	34
2.4	Red: predicted mean function; Black: true mean function; Blue dots: kernel locations for MSSS; Gray dots: observed data. All models fit the true mean function pretty well over most of the domain. However only the MSSS with smaller kernel width fit the data well at the discontinuities ( $x=4$ and $x=6$ ). . . . .	37
2.5	Two-dimensional simulated and predicted surfaces on the unit square, one row per dataset. First column is the true surface, and each additional column corresponds to the predicted mean surface from the considered models. . . . .	41
2.6	Plots of the maximum resolution (Res) active at each point on the unit square for the best MSSS model by marginal model probability for each of the three simulated two-dimensional datasets. For the two stationary examples, datasets A and B, the pattern in the multi-resolution structure does not change greatly across the domain.	41

2.7	Predicted SST in Celsius with $\nu = 3$ , a kernel width of 2.5, and the additional restrictions discussed above . . . . .	42
2.8	Plot of the maximum resolution active at each point on the surface for the MSSS model on SST in the Mediterranean. . . . .	43
3.1	J(1)=5 1d Optimization Path Summary . . . . .	62
3.2	J(1)=10 1d Optimization Path Summary . . . . .	62
3.3	J(1)=20 1d Optimization Path Summary . . . . .	63
3.4	J(1)=25 2d Smooth GP Optimization Path Summary . . . . .	63
3.5	J(1)=100 2d Smooth GP Optimization Path Summary . . . . .	64
3.6	J(1)=400 2d Smooth GP Optimization Path Summary . . . . .	64
3.7	J(1)=25 2d Rough GP Optimization Path Summary . . . . .	65
3.8	J(1)=100 2d Rough GP Optimization Path Summary . . . . .	65
3.9	J(1)=400 2d Rough GP Optimization Path Summary . . . . .	66
3.10	Predictions over the Mediterranean for the model with the best BIC	67
3.11	Resolutions active over the Mediterranean for the model with the best BIC . . . . .	67
3.12	J(1)=40 SST Optimization Path Summary . . . . .	68
3.13	J(1)=160 SST Optimization Path Summary . . . . .	68
5.1	Predicted vs Actual. . . . .	94
5.2	Length of a 90% prediction interval. . . . .	95
5.3	Predicted surfaces obtained by varying smoothness parameters. . . . .	96
5.4	Predicted surfaces obtained by varying Markov random field parameters. . . . .	96
5.5	Predicted surfaces obtained by varying $J(1)$ . . . . .	97
5.6	Multi-scale spatial optimization fit to the data. . . . .	97
6.1	Full vs. training satellite measurements from August 4th, 2016. . . . .	104
6.2	Predicted values for this model. . . . .	107
6.3	Spatial residual plot for this model. . . . .	107
6.4	Resolutions used for this model. . . . .	108

6.5	Elevation and percent forest coverage over this region. . . . .	109
6.6	Resolution used for the model with elevation as a predictor. . . .	110
6.7	July 2003 sea surface temperatures off the coast of California. . .	112
6.8	Bathymetry off the coast of California. . . . .	112
6.9	Fitted values on a grid off the coast of California after 100 iterations of MSSS. . . . .	113
6.10	Posterior expected number of resolutions active off the coast of California after 100 iterations of MSSS. . . . .	114
6.11	Missing pattern off the cost of California . . . . .	115
6.12	Predicted temperatures off the cost of California with missing data	116
6.13	Estimated number of resolutions off the cost of California with miss- ing data . . . . .	117
7.1	Left panel: randomly selected realizations from our prior, one for each sparsity level, plotted on the same axes. Smaller values of $\pi$ correspond to stronger spatial variability of the roughness of the sample paths. For $\pi = 1$ we observe homogeneous local variability across space. Right panel: distribution of Holder exponents as a function of $\pi$ . An evident increasing pattern is present, except for $\pi = 1$ , where Holder exponents typically have a very small range, indicating that the fields with dense multi-resolution grids do not exhibit multi-fractal behavior. . . . .	120

# List of Tables

2.1	Equation for and plot of the mean function for the nonstationary 1D example data. . . . .	32
2.2	Different parameter settings for the prior used in the fully crossed design for the simulation study. $\mu = \frac{a_\pi}{a_\pi + b_\pi}$ and $\theta = a_\pi + b_\pi$ . . . .	35
2.3	Numerical summaries for the different models in the one-dimensional example. MSSS always provides excellent predictive interval coverage, and gives excellent out-of-sample fit when either enough kernels at the first resolution are used or the kernels are of the appropriate shape. . . . .	38
2.4	Numerical summaries for the models on the two-dimensional GP dataset with a Matern correlation kernel $\psi = 2$ . Notice that runtimes for MRA do not include parameter estimation. . . . .	38
2.5	Numerical summaries for the models on the 2D Gaussian Process dataset with a Matern correlation kernel, $\psi = .5$ . . . . .	39
2.6	Numerical summaries for the models on the 2D nonstationary kernel convolution dataset. . . . .	39
2.7	Numerical summaries for the models on the SST dataset. . . . .	42
4.1	Results for the Mediterranean data with $J(1) = 40$ . . . . .	75
4.2	Results for the Mediterranean data with $J(1) = 40$ . . . . .	78
5.1	Partial list of R package implementations of discussed methods. . .	81
5.2	Parameters for <code>msss_fit</code> . . . . .	90

5.3	Parameters for <code>msss_pred</code> . . . . .	91
5.4	Parameters for <code>mr_optim_fit</code> . . . . .	92
5.5	Parameters for <code>mr_optim_pred</code> . . . . .	93
6.1	Metrics for model comparison in Heaton et al. (2019). . . . .	105
6.2	Summary table of results from Heaton et al. (2019) with MSSS at the top . . . . .	106

## Abstract

Nonstationary Models for Large Spatial Datasets Using Multi-resolution Process  
Convolutions

by

Daniel Kirsner

Large spatial datasets often exhibit fine scale features that only occur in sub-domains of the space, coupled with large scale features at much larger ranges. The most commonly used model used for spatial datasets is the Gaussian Process, but evaluation of likelihood is computationally expensive. Additionally, traditional Gaussian Processes models make very strong assumptions regarding the symmetry of the Gaussian field. In particular they assume stationarity, namely that covariance functions depend only on the displacement vector between two points, not their locations. This assumption prevents stationary Gaussian Processes from accounting for multi-scale features that only exist in parts of the spatial domain. In this work, we develop multi-resolution kernel convolution methods that explicitly account for local multi-scale features through spatially varying resolution. These methods define an increasingly refined set of nested kernels, and induce sparsity on these grids.

We first introduce modifications to existing multi-resolution kernel convolution models that result in spatially varying resolution through a sparsity inducing prior. We cast spatially varying resolution as a model selection problem, and develop a Shotgun Stochastic Search algorithm that considers an infinite number of resolutions, and permits uncertainty quantification without resorting to MCMC.

We propose a LASSO like prior that achieves spatially varying resolution at its maximum a posteriori, and develop a proximal gradient descent algorithm to find

this optimum considering an infinite number of resolutions. We then develop a Bayesian model averaging approach to perform uncertainty quantification in this setting.

We implement these methods in an efficient and reproducible manner via the R package `MSSS`. We discuss in detail the computational efficiency achieved by leveraging parallel computation, compactly supported kernels, add one column regression updates, and modern optimization methods in `MSSS`. We demonstrate the local feature identification properties of spatially varying resolution and demonstrate the computational performance by considering a land surface temperature dataset from the Ozarks, and a large sea surface temperature dataset collected by a satellite off the coast of California



For Emma.

## Acknowledgments

The text of Chapter 2 is an adaptation of the previously published article:

Daniel Kirsner and Bruno Sansó. Multiscale shotgun stochastic search for large spatial datasets. *Computational Statistics & Data Analysis*, page 106931, 2020

The co-author listed in this publication directed and supervised the research which forms the basis for this dissertation. I also acknowledge the editors of *Computational Statistics & Data Analysis* and the two anonymous reviewers, whose suggestions made this work stronger. This research was funded in part by National Science Foundation award DMS-1513076. I have been very lucky to work with my advisor, Bruno Sansó. His knowledge and intuition in the field of spatial statistics has been crucial to my success, and his advocacy for his students is exceptional. He is patient and attentive as an advisor, and has been supportive of me at every step of my studies.

I have repeatedly drawn on the work of Rajarshi Guhaniyogi, and I am grateful to Raj for going above and beyond in discussing that work with me. I also am grateful to the rest of the Statistics faculty for their excellent teaching and mentorship, without which I would have been lost.

The other students have contributed as much to my learning as the faculty. My cohort mates Matt Heiner, Kurtis Shuler, and Daniel Spencer have provided advice, camaraderie, and made the office an enjoyable place to be my entire time in Santa Cruz. Arthur Lui provided C++ help that was essential to the development of the algorithms in this paper, and Raquel Barata was a perfect neighbor.

# Chapter 1

## Introduction

### 1.1 Background

The traditional problem of model-based spatial statistics is to use a collection of spatially referenced observations to produce an estimate of the mean function of the data generating process, together with uncertainty intervals, across the entire domain. It is usually the case that observations are irregularly scattered over a large domain, and increasingly often, there is a need to handle very large amounts of data. Furthermore, it is desirable that models for this kind of data are able to capture behavior that varies due to differences in scales and in locations. For example, to model sea surface temperature in the Mediterranean, a model must be able to account for large scale features like the fact that the sea is warmer near Turkey than near Spain, and small scale features like how tiny islands in Greece can affect the temperature near the island. Gaussian processes provide a flexible framework for modeling this kind of data.

A well established literature has been developed on the idea of using Gaussian processes as the main tool for model-based geostatistics (see, for example, Gelfand et al., 2010, for a comprehensive review). However, for  $n$  data points,

the computation of the likelihood for a Gaussian process requires inversion of an  $n$  by  $n$  covariance matrix, which is computationally expensive ( $O(n^3)$ ). There are numerous approaches to resolving this issue in a big spatial data context, see Heaton et al. (2019) for a comparative review, and Banerjee (2017) for a review of Bayesian methods.

### 1.1.1 Computational Efficiency and Spatial Models

One class of sparsity inducing techniques seek to reduce the number of non-zero elements in the covariance matrix of the Gaussian process through compactly supported covariance functions. Furrer et al. (2006) show that if a covariance function with known parameters is tapered by multiplying it with a compactly supported covariance function, then the resulting kriging estimates are still asymptotically optimal, but the computational advantages can be significant. Kaufman et al. (2008) show how to estimate the parameters of the underlying covariance function in the tapering context. Sparse covariance matrices can also be built by spatial partitioning, which partitions the domain and assumes independence between subregions. These methods allow for parallel computation as well, and result in nonstationary spatial models, which is often a desirable feature. Partitioning can either be fixed a priori via a number of deterministic rules, such as equal areas (Sang et al., 2011) or clustering (Anderson et al. (2014), Heaton et al. (2017)). Alternatively, the partitions can be learned via computationally intensive trans-dimensional MCMC approaches, such as Kim et al. (2005) and Gramacy and Lee (2008), which will be discussed further in the review of nonstationary methods.

Another class of sparsity inducing techniques build sparsity in the precision matrix of the Gaussian process. Gaussian Markov random fields (GMRF) (Rue and Held, 2005) enforce sparsity by restricting dependence to a neighborhood

defined by a respecified undirected graph. However, this method does not take distance into account, so is most appropriate for data collected on a grid, or when the domain is partitioned in a manner similar to zip codes, or states. Stochastic partial differential equation (PDE) approaches (Lindgren et al., 2011) rely on the equivalence of Matern covariance fields and stochastic PDEs, and then use a basis expansion of the spatial process. Nearest neighbor Gaussian processes (Datta et al., 2016) build a spatial process where each point depends only on the  $k$  nearest neighbors to it, but still uses spatial covariance functions. This induces a sparse precision matrix but unlike Rue and Held (2005), this process still accounts for distance. The computational advantage of sparsity inducing techniques comes from the use of sparse matrix routines. However, a preprocessing step that reorders the data so that the covariance or precision matrix is approximately banded is often required to optimize these methods.

If the data is collected on a regular grid and the model chosen is stationary, the fast Fourier transformation can be used to rapidly evaluate the Gaussian process likelihood. This is sometimes referred to as circulant embedding (Chan and Wood, 1999). An approach that uses circulant embedding to fit models with missing data, which is necessary for out of sample prediction, and that is able to manage edge effects, was developed by Stroud et al. (2017). An approach that reduces the the computational complexity through an approximate covariance model was developed in Guinness (2019).

Dimension reduction is another common approach. These techniques express the underlying spatial process as a sum of  $J$  basis functions, where  $J \ll n$ . The fixed rank kriging approach of Cressie and Johannesson (2008) approximates a spatial process with a linear combination of  $K$  basis functions, but does not attempt to approximate a particular Gaussian process. Predictive processes (Banerjee

et al., 2008) approximate a Gaussian process with a specific covariance function through a set of knots, but this approximation results in biased estimates of the non spatial error. The modified predictive process (Finley et al., 2009) resolves this bias. Discrete process convolutions (Higdon (1998), Stein (2007) Lemos and Sansó (2009) among many others) approximate a Gaussian Process with a linear combination of basis functions that are generated by kernels or radial basis functions usually centered on a grid. Conditional on the data and the parameters, the model reduces to a linear regression with  $J$  coefficients, which entails a reduction of the computational complexity to  $O(J^2n + J^3)$ . A further computational advantage can be gained if the basis functions have compact support. Then, sparse matrix routines can be used to reduce the  $O(J^3)$  portion of the computational complexity. For all low rank methods, selection of  $J$  is a difficulty. If  $J$  is too small, then the model can miss the small scale features, but increasing  $J$  can make the parameter space unfeasibly large and cause numerical issues.

### 1.1.2 Non-stationarity and Spatial Models

Traditional Gaussian process geostatistical models assume stationarity, namely that covariance functions depend only on the displacement vector between two points, not their locations. This assumption can limit the performance of a model. In the context of sea surface temperature, Karspeck et al. (2012) have argued against the appropriateness of stationary Gaussian processes, and Lasinio et al. (2013) discuss how large datasets often display non-stationarity. Many approaches have been developed to account for non-stationarity.

Some approaches map the nonstationary field onto a stationary one. Sampson and Guttorp (1992) use the idea of deforming the space through a function that maps locations in the original, nonstationary field to a latent, stationary field.

They represent this function through splines. Schmidt and O’Hagan (2003) extend this model through a Gaussian process prior for the mapping function, and estimate all parameters via MCMC. These methods are sometimes referred to as image warping. A related approach was developed in Bornn et al. (2012), where a nonstationary field is embedded into a higher dimensional space where the field will exhibit stationarity. The authors use a spline based method similar to Sampson and Guttorp (1992) that estimates the embedding function, but preserves the lower dimensional locations.

Some approaches create classes of nonstationary covariance functions. The approach of Fuentes and Smith (2001) allows the parameters of stationary covariance function to vary in space by convolving them with a fixed kernel. This hierarchy creates a new, nonstationary covariance function that is nonstationary, but locally stationary. Paciorek and Schervish (2006) derive a class of nonstationary covariance functions through convolving stationary covariance functions with spatially varying kernels. The resulting class of covariance functions are closed form. Inference on both of these models requires MCMC on the resulting full Gaussian Process.

By construction, finite basis function representations of Gaussian processes, like discrete process convolutions, are non-stationary, but most models in the literature using such formulations do not attempt to explicitly describe the characteristics of the non-stationarity. Lemos and Sansó (2009); Lemos and Sansó (2012) extend the process convolution approach to reflect non-stationarity explicitly by considering kernels with spatially varying elliptical shapes. This is coupled with a GMRF prior on the knot coefficients, which encourages some spatial sharing of information and improves computation. This model is sensitive to the choice of the number of basis functions, but preserves the computational advantages of low

rank approaches while explicitly characterizing non-stationarity.

Another set of approaches rely on partitioning the domain, using a stationary process on each element of the partition, and assuming independence between the partitions. These approaches also have computational advantages. Rather than inversion of a single  $n$  by  $n$  covariance matrix, evaluation of the likelihood requires only inverting the covariance matrix for each partition. This also permits parallel computation. The piecewise Gaussian process of Kim et al. (2005) accomplishes this by dividing the domain into non-overlapping regions through Voronoi partitioning. To perform inference on the surface and average over different potential partitions, reversible jump MCMC is required, which is computationally expensive. The approach of Gramacy and Lee (2008) replaces the Voronoi partitioning with partitioning through a tree, which is related to CART (Chipman et al., 1998) and tends to result in a smaller number of partitions. The MCMC in this case is still trans-dimensional, but is less of a computational burden than standard reversible jump MCMC. However, realizations from both of these models result in covariance functions that have jagged transitions at the borders of the partitions, which is not always desirable. An approach that combines the treed partitioning approach with process convolutions was proposed by Liang and Lee (2011). In it, a reduced rank discrete process convolution is fit to each of the partitions. This allows for different kernel parameters to be used in different partitions, which makes the model extremely flexible.

Multi resolution models layer multiple processes on top of each other at different resolutions to accomplish dimension reduction while accounting for both fine and large scale features in the data. The approach of Nychka et al. (2015) approximates stationary Gaussian processes through a Gaussian Markov random field prior on coefficients of basis functions at each resolution. This approach also



enforces prior independence between coefficients at different resolutions. The resulting fields are nonstationary, but don't attempt to approximate nonstationary processes directly. However, it is possible for the parameters of the basis functions to vary in space as well, which could more directly model nonstationary. The multi-resolution predictive process in Katzfuss (2017) recursively fits a predictive process (Banerjee et al., 2008) at increasing resolutions by refining an original set of knots. This approach allows for nonstationary covariance functions, but still enforces the same multi-resolution structure across the entire field.

A Bayesian approach that partially relaxes this rigid structure was proposed in Guhaniyogi and Sansó (2017). They propose discrete process convolution with a nested set of knots and isotropic, compactly supported basis functions at differing resolutions. The range of these basis functions is decreasing in resolution, to encourage higher resolutions to reflect high frequency behavior. A prior on the knot coefficients enforces spatially varying increasing shrinkage in resolution, so different parts of the domain adaptively receive different amounts of shrinkage. MCMC is required, but can be performed efficiently due to the compact support of these basis functions. A related model, that provides an extensions to of Katzfuss (2017) approach, was proposed by Benedetti et al. (2018). A spike and slab prior on basis function weights is linked hierarchically between resolutions in a manner that results in spatially varying shrinkage.

## 1.2 Discrete Process Convolutions

We will begin by more precisely discussing discrete process convolutions. Let  $\{w(s) : s \in D\}$  be the spatial process of interest on the domain  $D \in \mathbb{R}^d$ , where  $d \in \{1, 2\}$ . We can construct this Gaussian process in the manner of Higdon (1998). Let  $K(s)$  be a kernel function, and  $\beta_j, j = 1, \dots, J$  a set of Gaussian

random variables corresponding to a set of points in  $D$ ,  $s_1, \dots, s_J$ , usually defined over a regular grid. We focus on the finite dimensional representation of the process,

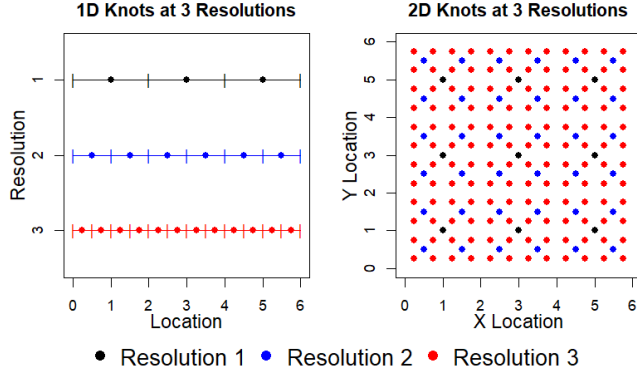
$$w(s) = \sum_{j=1}^J K(s, s_j | \zeta) \beta_j.$$

Note that in general, the parameters for the kernel  $\zeta$  can be a vector. Following Lemos and Sansó (2012) we term this a discrete process convolution (DPC).

These models are subject to the choice of the knot locations  $s_j$ , their total number  $J$ , the kernel functions  $K$  and their associated parameters  $\phi$ . Even with a small number of knots, DPCs are able to capture the long range behavior of a spatial field. But, unless  $J$  is taken as a very large number, a DPC can miss short range features. And clearly, using a very large number of knots defeats the dimension reduction purpose of the DPC representation. In addition, it is often the case that some areas of the domain will show substantially more variability than others. To approach this issue, we will later introduce the multi-resolution DPC. This embeds multiple DPCs at different resolutions into the same model.

### 1.2.1 Domain Partitioning

To define the structure of our multiple resolutions, we will follow the notation of Guhaniyogi and Sansó (2017). Start by partitioning the spatial domain  $D$  into  $J(1)$  square subregions  $D_1, \dots, D_{J(1)}$ . The centers of these regions define the first resolution of knots. To define resolution 2, each of the square subregions  $D_i$  will be partitioned into  $2^d$  square subregions, giving us  $J(2) = J(1) \times 2^d$  subregions on the second resolution. The  $2^d$  partitions of domain  $D_i$  are labeled as  $D_{i,i_2}$  where  $i_2 \in \{1, \dots, 2^d\}$ . We can now iteratively define resolution  $r$  by partitioning the subregions at resolution  $r - 1$  into  $2^d$  square regions, and can index a domain in this region as  $D_{i_1, \dots, i_r}$  where  $i_1 \in \{1, \dots, J(1)\}$  and  $i_2, \dots, i_r \in$



**Figure 1.1:** On the left, a plot of knot locations for the first 3 resolutions in 1 dimension with  $J(1) = 3$ . On the right, a plot of knot locations for the first 3 resolutions in 2 dimensions with  $J(1) = 9$ .

$\{1, \dots, 2^d\}$ . We will refer to the center of domain  $D_{i_1, \dots, i_r}$  as a knot  $s_j^r$  where  $j = \sum_{l=1}^{r-1} ((i_l - 1)(2^d)^{r-l}) + i_r$ . Figure 1.1 displays both one and two dimensional examples of the knot placements.

From this definition, we can see that  $J(r) = J(1) \times 2^{d(r-1)}$ . We can view this partitioning as forming a tree, with the highest nodes at the lowest resolution, and lower nodes representing higher resolutions.  $2^d$  branches come from each node to the nodes at the next level. Motivated by this tree structure, we will define  $parent(D_{i_1, \dots, i_{r-1}, i_r}) = D_{i_1, \dots, i_{r-1}}$  and  $children(D_{i_1, \dots, i_{r-1}}) = \{D_{i_1, \dots, i_{r-1}, i_r} : i_r \in 1, \dots, 2^d\}$ . These definitions are also useful to apply to the knots. We define  $parent(s_j^r) = s_{\lfloor \frac{j-1}{2^d} \rfloor + 1}^{r-1}$  and  $children(s_j^{r-1}) = \{s_k^r : k \in 2^p(j-1) + 1, 2^p(j-1) + 2, \dots, 2^p(j-1) + 2^p\}$ . Lastly, we define the subtree, which is the set of all domains that are ancestors of a particular domain. Formally,  $subtree(D_{i_1, \dots, i_r}) = \{D_{i_1, \dots, i_r, \dots}\}$ .

## 1.2.2 Spatially Varying Shrinkage

We can now introduce the multi-resolution discrete process convolution, and discuss spatially varying shrinkage in more detail. Following the notation of 1.2.1,

we define the spatial process as multi-resolution process convolution, where

$$w(s) = \sum_{r=1}^R \sum_{j=1}^{J(r)} K(s, s_j^r | \zeta_r) \beta_j^r.$$

Note that the parameters of the kernel change with the resolution. In Guhaniyogi and Sansó (2017), the kernel function is taken to be a compactly supported Wendland kernel. If  $l = \lfloor d/2 \rfloor + 2$ , then

$$K(s, s_j^r, \phi_r) = \left(1 - \frac{\|s - s_j^r\|}{\phi_r}\right)_+^{l+1} \left[1 + (l+1) \frac{\|s - s_j^r\|}{\phi_r}\right].$$

The range of this kernel,  $\phi_r$ , is defined as  $\eta \|s_j^r - s_{j-1}^r\|$ , which enforces smaller kernel widths at higher resolutions. The authors also place the following prior on the knot coefficients:

$$\beta_j^r \sim N(0, \alpha_j^r),$$

$$\alpha_j^1 = \delta^{-1}, \alpha_j^2 = \delta_{j,2}^{-1}, \alpha_j^r = \alpha_{\lfloor \frac{j-1}{2} \rfloor}^{r-1} \delta_{j,r}^{-1},$$

$$\delta^1 \sim \text{Gamma}(2, 1), \delta_{j,r} \sim \text{Gamma}(c, 1), c > 2.$$

This prior enforces shrinkage that is increasing in resolution, with  $\mathbb{E}[\beta_j^r] = 0$  and  $\text{Var}[\beta_j^r] = \frac{1}{(c-1)^{r-1}} \rightarrow 0$  as  $r \rightarrow \infty$ . This shrinkage in resolution is paired with spatially varying shrinkage, where

$$\text{Var}(\beta_j^r) = \delta_{j,r}^{-1} \text{Var}(\text{parent}(\beta_j^r)).$$

This means that the shrinkage applied to a parent is also applied to all of its ancestors. Subdomains with a large amount of shrinkage will be encouraged to have knot coefficients that are very close to zero, while other subdomains can have very little shrinkage applied to them. This spatially varying shrinkage can

explicitly characterize non-stationarity in a spatial surface.

### 1.2.3 Research Objectives

In this work, multi-resolution kernel convolution methods that use kernel convolutions in increasingly refined sets of nested grids are developed. The objective of this research is to study to methods that achieve spatially varying resolution, which forces parts of the nested grid to be empty. This sparsity can allow for an infinite number of resolutions to be considered a priori, which makes these methods extremely robust when compared to standard process convolutions, which are sensitive to the choice of the number of basis functions. When comparing methods that induce varying resolution to spatially varying shrinkage, an analogy can be made to model selection in the regression context. Although a full review is omitted here, the review by Hahn and Carvalho (2015) covers many of the trade-offs between shrinkage and sparsity.

For the purposes of this work, a multi-resolution set of knots  $\mathbf{T}$  is termed to have spatially varying resolution if it meets the following criteria:

1. For all  $j \in J(1), s_j^1 \in \mathbf{T}$ .
2. For all  $r > 1, s_j^r \in \mathbf{T} \rightarrow \text{parent}(s_j^r) \in \mathbf{T}$ .
3.  $|\mathbf{T}| < \infty$

The first criterion ensures that the resulting spatial surface has no gaps. The second criterion coupled with the third allows us to, given a set of knots  $\mathbf{T}$ , assign each point in the spatial domain  $D$  an integer number of resolutions.

In Chapter 2, we develop an inferential method for inference on spatial multi-resolution spatial models using shotgun stochastic search. This method casts spatially varying resolution as a Bayesian model selection problem. We propose a

stochastic process prior on knot inclusion that induces spatially varying resolution in the resulting models. This knot selection is performed without a maximum resolution enforced in advance, so an infinite number of models are considered. We develop a prior on the coefficients corresponding to the included knots that results in a closed form marginal distribution, allowing for easy computation of the posterior model probabilities. We also develop a model search algorithm that allows for exploration of the infinite dimensional space of potential knot configurations.

In Chapter 3, we develop Multi-Scale Spatial Optimization. This method casts spatially varying resolution as an optimization problem. We assume a prior on the coefficients associated with the knots that achieves spatially varying resolution at its maximum a posteriori, similar to the LASSO (Tibshirani, 1996). We develop a proximal gradient descent algorithm for finding the MAP efficiently for a set of penalty parameters.

In Chapter 4, we develop a Bayesian model averaging approach to gain some uncertainty quantification on the set of maxima found using the optimization method from the previous chapter. We also discuss how this model and other with similar characteristics struggle with extremely large datasets, and propose modifications to the prior that can help alleviate this issue.

To achieve the objectives in the previous chapters, the development of the R package `MSSS` to fit models with spatially varying resolution was required. This software is discussed in detail in Chapter 5.

In Chapter 6, we discuss in depth the manner in which spatially varying resolution is able to identify features in geostatistical data, and how this differs from other spatial models for large datasets. We first perform a comparison to other spatial models by reviewing in detail the land surface temperature case study of

Heaton et al. (2019), and showing how models with spatially varying resolution are able to explicitly identify a mountain range in this dataset. We then analyze a large sea surface temperature dataset collected off the coast of California during a period with an upwelling.

# Chapter 2

## Multi-scale Shotgun Stochastic Search

### 2.1 Bayesian Model Selection

In this chapter, we appeal to the Bayesian model selection literature to select promising sets of knots that display spatially varying resolution. We will first review this literature in the linear model context. The traditional setup for such models is

$$p(\mathbf{y}|M_\gamma, \boldsymbol{\beta}_\gamma, \sigma^2) \sim N_n(\mathbf{X}_\gamma \boldsymbol{\beta}_\gamma, \sigma^2) \quad (2.1)$$

$$p(\boldsymbol{\beta}_\gamma, \sigma^2) = p(\boldsymbol{\beta}_\gamma|\sigma^2)p(\sigma^2) \quad (2.2)$$

$$Pr(\beta_j = 0|\gamma_j = 0) = 1 \quad (2.3)$$

$$Pr(\gamma_j = 1) = \pi^{\gamma_j}(1 - \pi)^{1-\gamma_j} \quad (2.4)$$

This is sometimes referred to spike and slab variable selection (Mitchell and Beauchamp, 1988). Some authors have argued for shrinkage priors (examples



include Park and Casella (2008) and Armagan et al. (2013)) as opposed to selection. These models include priors with large amounts of mass near zero to induce irrelevant variables to take values close to zero, while also having fat tails that allow important variables to remain large. However, these models do not achieve exact sparsity, so cannot attain spatially varying resolution. The spike and slab model is subject to choice of the slab equation (2.2) and the choice of the prior on the model space equation (2.4). There are  $2^p$  potential models, so if  $p$  is large, a strategy for enumerating potential models must be considered.

An early default choice for the slab is the null based g-prior of Zellner (1986), where

$$p(\boldsymbol{\beta}_\gamma | \sigma^2) = N_p(0, \sigma^2 g(\mathbf{X}_\gamma^T \mathbf{X}_\gamma)^{-1}).$$

This prior results in a closed form posterior model probability, which is computationally convenient. However, the resulting resulting Bayes factors and posterior model probabilities display suboptimal behaviors when the hyperparameter  $g$  is fixed (Liang et al., 2008). These issues can be resolved by placing a prior on  $g$  that resolves the issue while preserving some of the computational advantages. Some examples are Liang et al. (2008) which will be discussed in more detail in this chapter and Bayarri et al. (2012), which will be discussed in chapter 5. Other priors, such as the non-local prior (Johnson and Rossell, 2012), which has no mass in the neighborhood of zero, have been proposed. The non-local prior does not result in a closed form Bayes factor, but the resulting posterior model probabilities have very strong asymptotic optimality, and a Laplace approximation is provided.

The prior on the model space controls how large the model is expected to be a priori. Assuming that all potential models have equal probability is equivalent to setting  $\pi = 1/2$ , but this is often much larger of a model than is expected a priori. A common choice is to set  $\pi$  via Empirical Bayes, (George and Foster,

2000). Scott and Berger (2010) argue that setting a prior on  $\pi$  results in better calibrated models, specifically because this prior controls for multiplicity. The closely related complexity prior (Castillo et al., 2015) is a  $Beta(1, p^u)$  with  $u > 1$ . The authors show that this prior has asymptotic optimality properties and is useful for large  $p$ .

In this setting, there are  $2^p$  potential  $\gamma$ , which is too many to enumerate exhaustively for even moderate  $p$ . However, to make predictions that account for model uncertainty using Bayesian model averaging (Raftery et al., 1997), we require  $p(\gamma|\mathbf{y})$ . A number of strategies have been proposed to compute  $p(\gamma|\mathbf{y})$  without explicitly evaluating every potential  $\gamma$ . If equation (2.2) results in conjugacy, then the marginal model probabilities are available in closed form, allowing for Gibbs sampling. In a non-conjugate setting, the posterior can be simulated from using reversible jump MCMC (Green, 1995). Stochastic search variable selection (George and McCulloch, 1993) replaces equation (2.3) with a very concentrated normal prior, which allows for Gibbs sampling even in more complex cases. All of these MCMC approaches suffer from mixing issues, especially with large  $p$ , when the  $\gamma_j$  are sampled one at a time, but strategies for updating in blocks are extremely difficult to generalize. Bottolo et al. (2010) develop a parallel tempering approach that alleviates some of the mixing issues inherent to the Markov chain based sampling approaches. In all previously discussed searches, each iteration requires the evaluation of a number of models, but results in only a single sample, and requires revisiting the same model multiple times in order to compute posterior model probabilities. The non MCMC approach developed by Hans et al. (2007) evaluates entire neighborhoods in parallel at each iteration, which can have computational advantages. Another non MCMC based approach was developed by Clyde et al. (2011). This method enumerates the model space

by sampling from the space of potential models without replacement, so iterations are not wasted.

We develop a multi-resolution model that achieves spatially varying resolution through the tools of Bayesian variable selection. We adopt a form of the hyper-g prior (Liang et al., 2008) as the slab distribution and discuss its spatial properties. For a prior on the model space, we develop a stochastic process prior in an infinite number of dimensions that forces the models considered to demonstrate spatially varying resolution. And to explore the space of possible sparse knot configurations by using the tools of model selection, we extend shotgun stochastic search (Hans et al., 2007) to the this setting, which allows our method to take advantage of parallel computing environments and to evaluate many models at each iteration. Due to the large number of potential models, these computational advantages are essential. We demonstrate how to use this method to perform prediction, uncertainty quantification, and demonstrate competitive computational performance when compared with other approaches on a variety of spatial fields. We also demonstrate how the resulting spatially varying resolution allows us to graphically summarize the multi-resolution structure in the field.

Though to our knowledge, the use of knot selection has not been used to fit spatial models, some similar ideas have been used to perform nonlinear regression. In Smith and Kohn (1996), the authors use a g-slab, a fixed  $\pi$  in the model space prior, and Gibbs sampling to perform knot selection on an additive model with a cubic spline component and a fixed maximum number of knots.

## 2.2 A Bayesian multi-resolution model

We start with a standard spatial regression model,

$$y(s)_i = \mathbf{x}(s)_i^T \boldsymbol{\alpha} + w(s) + \epsilon(s)_i, \quad \epsilon(s)_i \stackrel{i.i.d.}{\sim} N(0, \sigma^2),$$

where  $\mathbf{x}(s)_i$  is a  $q \times 1$  vector of individual level predictors,  $\boldsymbol{\alpha}$  is the  $q \times 1$  vector of fixed effect regression coefficients associated with the predictors,  $w(s)$  is the spatial effect,  $i$  is the index for replicates at a particular point  $s$ , and  $\epsilon(s)_i$  is random noise, corresponding to observational error or micro-scale variability. Note that the predictors occur on the individual level, not the level of the spatial process. The spatial process is defined by a multi-resolution DPC,

$$w(s) = \sum_{r=1}^{\infty} \sum_{j=1}^{J(r)} K(s, s_j^r | \phi_r, \nu) \beta_j^r.$$

For computational purposes, we require that  $K$  is compactly supported, with range  $\phi_r$ . To satisfy the desideratum that higher resolution kernels reflect small scale behavior, we let the kernel width decrease linearly as the resolution increases, i.e.  $\phi_r = \tau \|s_j^r - s_{j-1}^r\|$  for some  $\tau > 1$ . We propose to use a Bezier kernel (Brenning, 2001), which is compactly supported, and has a parameter  $\nu$  that controls the differentiability. This kernel function is defined as

$$K(s, s_j^r, \phi_r, \nu) = \begin{cases} \left(1 - \left(\frac{\|s - s_j^r\|}{\phi_r}\right)^2\right)^\nu & \|s - s_j^r\| < \phi_r \\ 0 & \text{otherwise.} \end{cases}$$

The compact support allows for the use of sparse matrix libraries, which speeds up the computation and reduces the memory overhead. In section 2.3 we will discuss the sensitivity of this method to the parameters  $\nu$  and  $\tau$ . We will now

turn our attention to the coefficients  $\beta_j^r$ . To achieve spatially varying *resolution*, we require sparsity, i.e.  $\beta_j^r = 0$  for some  $r$  and  $j$ , that is structured in a manner such that the number of resolutions varies in space.

### 2.2.1 A prior that induces spatially varying resolution

Motivated by this analogy, we will adapt a standard variable selection prior on the coefficients of our model (Hans et al., 2007) to this setting in order to induce spatially varying resolution. First, some notation must be introduced. Let  $\boldsymbol{\gamma} = [\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \dots]$  be a vector of infinite length with  $length(\boldsymbol{\gamma}_r) = J(r)$ . Let the  $j$ th entry of  $\boldsymbol{\gamma}_r$  be called  $\gamma_{r,j}$ ,  $j \in \{1, 2, \dots, J(r)\}$ . We will set  $\gamma_{r,j} = 1$  if  $\beta_j^r \neq 0$ , and will put a prior on this vector. This is vector of infinite length, so a prior  $p(\boldsymbol{\gamma})$  will be better understood as a stochastic process.

For our prior to induce spatially varying resolution, we would like to satisfy three properties. First, every resolution one knot must be associated with a nonzero coefficient. Without the entire resolution one grid, it is conceivable that parts of our spatial field would be modeled as constant in space, which does not make sense. Second, to allow the resolution to vary spatially, with a different number of resolutions possible at different locations, we only consider configurations that satisfy

$$\beta_j^r \neq 0 \implies \beta_{\lfloor \frac{j-1}{2^d} \rfloor + 1}^{r-1} \neq 0, \text{ which is identical to } \gamma_{r,j} = 1 \implies \gamma_{\lfloor r-1, \frac{j-1}{2^d} \rfloor + 1} = 1.$$

In other words, if a coefficient associated with a knot is nonzero, then the coefficient associated with the parent of the knot must also be nonzero. Throughout this paper, we will interchangeably refer to the set of nonzero coefficients in the model as the knot configuration. Models with these restrictions will produce a field that has locally varying resolution. This feature is analogous to the manner

in which shrinkage works in Guhaniyogi and Sansó (2017), where the shrinkage applied to the coefficient of a parent is also applied to its children. Lastly, we would like our prior to not result in infinitely many nonzero coefficients, as these models will not be computationally feasible. Motivated by this, we set  $Pr(\gamma_{1,j} = 1) = 1$ , and

$$Pr(\gamma_{r,j} = 1 | \gamma_{r-1}) = \pi \times \gamma_{r-1, \lfloor \frac{j-1}{2^d} \rfloor + 1}. \quad (2.5)$$

The prior described in equation (2.5) follows the three properties discussed above. Every resolution one knot is in the model, and if a knot at resolution  $r > 1$  is in the model, then its parent must be as well. To understand some of the other features of this prior, we can consider the random variable  $X_r = \sum_{j=1}^{J(r)} \gamma_{r,j}$ , the number of nonzero  $\beta_j^r$  at resolution  $r$ .  $X_r$  can be thought of as a branching process (Chung, 2012). The initial state of the process is  $X_1 = J(1)$ , and the offspring distribution be  $Binomial(2^d, \pi)$ . The extinction probability of this process is analogous to the probability of having a finite number of nonzero  $\beta_j^r$ . By the properties of a branching process, the extinction probability is 1 as long as the expected value of the offspring distribution is less than 1. Therefore, if we set  $\pi$  such that  $\pi 2^d < 1$ , then the extinction probability of this process is 1, and the prior favors a finite number of nonzero coefficients.

To complete the specification of our prior, we must either fix  $\pi$  at some constant less than 1, or assume  $\pi$  to be a random variable and choose a prior for it. Fixing  $\pi$  was shown to be inadequate in the setting of linear model selection in Scott and Berger (2010). Specifically, a fixed value of  $\pi$  results in inadequate correction for multiplicity, which can lead to models that are too large, which in our context translates to overfitting. Scott and Berger (2010) recommend the use of a Beta prior on  $\pi$ , and show that this corrects for multiplicity and results in smaller models in the linear regression context while still preserving a closed

form prior model probability, which we will need for our model selection procedure. Following this approach we let  $\pi \sim \text{Beta}(a_\pi, b_\pi)$ , so that under the prior,  $\mathbb{E}(\pi) = a_\pi / (a_\pi + b_\pi)$ .

This prior provides several attractive features. As shown in section 2.3, it is not very sensitive to varying  $a_\pi$  and  $b_\pi$ , and those parameters can be used to control the prior expected number of nonzero coefficients in a way that is easy to interpret. Recalling again the properties of a branching process,  $\mathbb{E} \left( \sum_{r=1}^{\infty} \sum_{j=1}^{J(r)} \gamma_{j,r} \right) = J(1)/(1 - 2^d \mathbb{E}(\pi))$ , provided that  $2^d a_\pi / (a_\pi + b_\pi) < 1$ . The prior probability of a particular coefficient being nonzero is decreasing geometrically with resolution, as  $Pr(\gamma_{r,j} = 1) = \mathbb{E}(\pi)^{r-1}$ . The prior probability for a particular set of nonzero coefficients  $\gamma$  is

$$p(\gamma) = \frac{B \left( a_\pi + \sum_{r=2}^{\infty} \sum_{j=1}^{J(r)} \gamma_{r,j}, b_\pi + \sum_{r=2}^{\infty} \left[ 2^d \sum_{j=1}^{J(r-1)} \gamma_{r-1,j} - \sum_{j=1}^{J(r)} \gamma_{r,j} \right] \right)}{B(a_\pi, b_\pi)}.$$

where  $B(a, b)$  is a Beta function. For further interpretability of the hyperparameters we use the alternative parameterization  $\theta = a_\pi + b_\pi$  and  $\mu = a_\pi / (a_\pi + b_\pi)$ .

It is important to notice that in the multi-resolution context, the Beta prior on  $\pi$  induces more sparsity in the coefficients than the model with fixed  $\pi$ . To demonstrate this, consider a simple context in a one dimensional space. Compare a prior with fixed  $\pi = .5$ , and denote this as  $p_1$ , to a prior with  $\pi \sim \text{Beta}(1, 1)$ , and denote this as  $p_2$ . These two priors produce the same prior expected number of knots, but have very different prior odds in favor of a smaller model. Let  $m_0$  be a model with  $J(1)$  first resolution knots, and no additional knots, and  $m_1$  be a model with a single second resolution knot, and the same  $J(1)$  first resolution

knots. Under the first prior, the prior odds are

$$\frac{p_1(m_0)}{p_1(m_1)} = \frac{1}{(1 - \pi)} = 4,$$

which is constant in  $J(1)$ . Under the second prior, using the fact that  $B(x+1, y) = B(x+y)x/(x+y)$ , the prior odds are

$$\frac{p_2(m_0)}{p_2(m_1)} = \frac{(2J(1) + 2)(2J(1) + 3)}{2J(1) + 1}.$$

This expression indicates that under  $p_2$ , the prior odds in favor of the smaller model are increasing as  $J(1)$  increases, which favors the smaller model more strongly for larger models. This has been confirmed by our empirical explorations, which indicate that using a random prior on  $\pi$  in our spatial multi-resolution model produces a smaller number of knots than the one that is obtained with a fixed value of  $\pi$ , without compromising goodness of fit.

### 2.2.2 Prior for the nonzero $\beta_j^r$ , $\alpha$ , and $\sigma^2$

The prior on the nonzero coefficients must be compatible for the spatial structure as well as computationally tractable. Let  $\boldsymbol{\beta} = \{\beta_1^1, \dots, \beta_{J(1)}^1, \beta_1^2, \dots, \beta_{J(2)}^2, \dots\}$ . Conditional on the vector  $\boldsymbol{\gamma}$ , we let  $\boldsymbol{\beta}_\gamma$  be a vector of length  $\sum_{r=1}^{\infty} \sum_{j=1}^{J(r)} \gamma_j^r$  that contains the nonzero  $\beta_j^r$ . Since  $\boldsymbol{\gamma}$  specifies which  $\beta_j^r$  are zero,  $p(\boldsymbol{\beta}, \boldsymbol{\gamma}) = p(\boldsymbol{\beta}|\boldsymbol{\gamma})p(\boldsymbol{\gamma}) = p(\boldsymbol{\beta}_\gamma|\boldsymbol{\gamma})p(\boldsymbol{\gamma})$ , so we can focus on specifying a prior on the nonzero coefficients  $\boldsymbol{\beta}_\gamma$ . In order to define the design matrix, let  $\mathbf{K}_r$  be an  $n \times J(r)$  matrix where the entry  $K_r(i, j) = K(s_i, s_j^r | \phi_r, \nu)$ , and  $\mathbf{K}_{r,\gamma}$  be the  $n \times \sum_{i=1}^{J(r)} \gamma_{r,i}$  matrix with columns that correspond to nonzero  $\gamma_{r,i}$ . Finally, let  $\mathbf{K}_\gamma = [\mathbf{K}_{1,\gamma}, \mathbf{K}_{2,\gamma}, \dots]$  be a  $\sum_{r=1}^{\infty} \sum_{j=1}^{J(r)} \gamma_j^r \times n$  matrix. This is the design matrix that corresponds to the nonzero  $\beta_j^r$ .



A g-prior (Zellner, 1986) on the coefficients associated with the knots, coupled with a reference prior on  $\sigma^2|\gamma$  satisfies our desiderata, and has analytically tractable marginals. Note that putting a reference prior on coefficients common to all models being compared, and a g-prior on the other coefficients is a commonly used approach in the model selection context (Liang et al., 2008). For this multi-resolution model, the g-prior is of the form

$$p(\boldsymbol{\beta}_\gamma, \sigma^2|\gamma) = p(\boldsymbol{\beta}_\gamma|\sigma^2, \gamma)p(\sigma^2|\gamma)$$

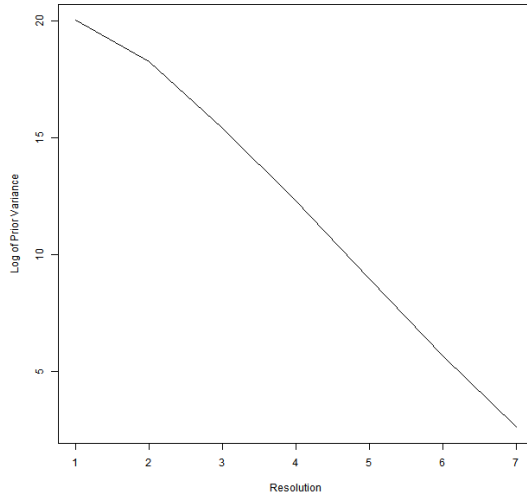
where

$$p(\boldsymbol{\beta}_\gamma|\sigma^2, \gamma) = N(0, g\sigma^2(\mathbf{K}_\gamma^T \mathbf{K}_\gamma)^{-1}),$$

with a reference prior on the fixed effects  $\boldsymbol{\alpha}$ , the error  $\sigma^2$ , and  $p(\gamma)$  specified in the manner of section 2.2.1.

Notice that, usually, the argument for using a reference prior on  $\boldsymbol{\alpha}$  is made by assuming that the columns of  $\mathbf{K}_\gamma$  have mean zero. However, centering this matrix would result in our basis functions no longer being compact. Fortunately, as Li and Clyde (2018) point out, the posterior distributions of the centered and non-centered models would have equivalent posteriors through a change of variables. Therefore, we will not center our design matrix.

An important property of the g-prior is that it induces shrinkage to high resolution knots that is, on average, larger than the one applied to low resolution ones. This behavior is due to the fact that more locations are in the range of kernels at lower resolutions. Therefore, the prior variance for the coefficients associated with the low resolution knots is higher than for the high resolution knots. We demonstrate this with a simple simulation. First, 10,000 locations are generated from a Uniform(0,10) distribution. Then, a number of design matrices corresponding



**Figure 2.1:** Average shrinkage under the G prior at different resolutions with equally spaced data

to multi-resolution sets of knots  $\mathbf{K}|\tau, \nu$  are formed for the Bezier kernel with a smoothness  $\nu = 1$  and a kernel width  $\tau = 1.5$ , 7 resolution, and 5 knots at 1. This is approximately equally spaced data with a dense grid of knots unlikely to occur in MSSS, but is useful for demonstration purposes. We compute the diagonal of  $(\mathbf{K}^t \mathbf{K})^{-1}$  and take the average by resolution. The results are displayed in figure 2.1. We observe that the shrinkage is approximately linear on the log scale, save for the jump from resolution 1 to 2, which makes the shrinkage geometric in resolution.

To set the value of  $g$  we observe that small values of  $g$  result in large shrinkage of the posterior mean. A popular default choice is  $g = n$ , which is known as a unit information prior (Kass and Wasserman, 1995), and provides reasonable performance in our context. The marginal likelihood for fixed  $g$  is available in closed form. However, Liang et al. (2008) observe that choosing  $g$  in this manner produces an information paradox. This paradox occurs because the marginal probability of model should approach 1 as  $r^2 \rightarrow 1$ , but in the case of a g-prior

with fixed  $g$ , this quantity converges to a constant. We can resolve this issue by using the hyper-g prior suggested by the authors, which is of the form  $g/(1+g) \sim \text{Beta}(1, a/2 - 1)$ . This prior resolves the information paradox for non-null models and still results in a closed expression for the marginal likelihood that involves the Gauss hypergeometric  ${}_2F_1$  function. Due to instability in the computation of  ${}_2F_1$ , for moderate to large  $n$ , this will require a Laplace approximation.

As a final note, the g-prior is improper if any columns of the design matrix are empty. In the context of this multi-resolution spatial model, this means that the prior does not make sense for a kernel function that has no data points within its range. To account for this, we propose to set  $\beta_j^r = 0$  if it is associated with an empty column, regardless of the resolution.

We have now specified a prior on the model space  $\gamma$ , and the marginal likelihood of the data conditional on  $\gamma$ , so up to a normalizing constant, our posterior model probabilities are

$$p(\gamma|\mathbf{y}) \propto \frac{a-2}{\sum_{i=1}^{J(r)} \gamma_{r,j} + a - 2} {}_2F_1 \left( \frac{n-1}{2}, 1, \frac{\sum_{i=1}^{J(r)} \gamma_{r,j} + a}{2}, R_\gamma^2 \right) \times \frac{B \left( a_\pi + \sum_{r=2}^{\infty} \sum_{j=1}^{J(r)} \gamma_{r,j}, b_\pi + \sum_{r=2}^{\infty} \left[ 2^d \sum_{j=1}^{J(r-1)} \gamma_{r-1,j} - \sum_{j=1}^{J(r)} \gamma_{r,j} \right] \right)}{B(a_\pi, b_\pi)}.$$

Therefore, for a particular  $\gamma$ , we can compute the posterior model probability by forming the design matrix  $\mathbf{K}_\gamma$ , estimating  $\beta_\gamma$  using least squares, calculating  $R_\gamma^2$ , and using the Laplace approximation to compute the  ${}_2F_1$  function. In the next sections, we will discuss how to use these model probabilities to explore the space of possible knot configurations, and how to update the least squares estimate of  $\beta_\gamma$  in a computationally efficient manner.

### 2.2.3 Extending shotgun stochastic search

Since the priors we have chosen results in closed form marginal model probabilities of particular configurations of knots, we can use shotgun stochastic search (Hans et al., 2007) to explore the space of possible knot configurations in a quick manner that takes advantage of modern computing architecture, namely multiple core processors. Shotgun stochastic search (SSS) proceeds as follows:

1. Given a current model  $m_c$ , a set of the top  $Q$  models evaluated, and their respective marginal model probabilities and coefficients, define a neighborhood of possible new models  $N$ .
2. Evaluate the marginal probability of each model in  $N$  in parallel, and update the top  $Q$  models.
3. Choose a new current model from the neighborhood with probabilities proportional to their marginal probabilities.

In order to fit spatial fields with locally varying resolution, we would like to extend SSS, but rather than selecting variables from a finite set, selecting configurations of multi-resolution knots arranged in nested grids. Note that this is a countably infinite set, as we are not truncating the number of resolution to consider. To use SSS, we need to define the neighborhood in a manner that is consistent with the prior from section 2.2.1.

To perform SSS,  $N$  is split into three groups,  $N = N_- \cup N_o \cup N_+$ .  $N_-$  is defined as all models of size  $p-1$  that contain predictors that are all selected from  $\gamma$ . Moving to a model in this set is termed a *deletion move*.  $N_+$  is defined as all models of size  $p+1$  that contain all  $p$  predictors from  $\gamma$  and one from  $\kappa$ . Moving to a model in this set is termed an *addition move*.  $N_o$  is defined as all models of

size  $p$  that contain  $p-1$  predictors from  $\gamma$  and one from  $\kappa$ . Moving to a model in this set is termed a *replacement move*.

In the multi-resolution knot selection context, if  $m_c$  is the current model, and  $\kappa = \{\kappa_1, \dots, \kappa_p\}$  is the set of knots in mode the restrictions above lead to the following neighborhood definitions. For addition moves, only models that add a single knot that is a child of one of the knots already in  $m_c$  will be considered. The potential knots to add  $S_+$  will be defined as  $S_+ = \{children(\kappa_i) \mid i \in \{1, \dots, p\} \setminus \kappa\}$ . So  $N_+$  is just all models one knot from  $S_+$ , and every knot in  $\kappa$ . For deletion moves, only knots that have no children will be considered for deletion. Formally, the potential deletion  $S_-$  will be defined as  $S_- = \{\kappa_i : [children(\kappa_i) \setminus \kappa] = children(\kappa_i)\}$ . Therefore,  $N_-$  is just all models with all but one knot in  $\kappa$ , with the knot removed  $\kappa_{del} \in S_-$ .

It is not very reasonable in our context for  $N_0$  to be all possible swap moves. This is because our space of possible variables is quite different in nature to the regression context. In regression, the swap moves are designed to explore spaces with correlated variables. For example, consider two possible predictors  $x_i$  and  $x_j$  that are highly correlated. If  $m_c$  contains  $x_i$ , it would be relatively unlikely for an add move to bring  $x_j$  into the model. But in the spatial context with compactly supported kernels, the columns that will have the highest correlations are parents and children, which cannot be swapped due to the restrictions we place on the knot placements. Knots on the same resolution have fairly low correlation as long as the kernel width is not very wide. For example, in a one dimensional setting, with uniformly distributed locations and one resolution of knots, for a kernel width of 1.5 and a smoothness of 1 (which we suggest as a default in section 2.3.4), the correlation between adjacent knots is only about .5.

## 2.2.4 Computational details

Given these choices, we can now formulate the algorithm for multi-scale shotgun stochastic search (MSSS). Given a current model  $m_c$ , and a list of the  $Q$  top models,

1. Form  $N = N_+ \cup N_-$  as defined above.
2. In parallel, for every  $m_p \in N$ , evaluate the marginal probability using the expressions above, and update the top  $Q$  models.
3. Sample  $m_{p-}$  from  $N_-$  and  $m_{p+}$  from  $N_+$  with probability proportional to the marginal model probabilities. Then sample a new  $m_p$  from  $\{m_{p+}, m_{p-}\}$  with probabilities proportional to their marginal probabilities. Return to step 1.

We run this algorithm until it reaches a local maximum, i.e. when the set of  $Q$  top models does not change for some number of iterations.

For fast calculation of the marginal model probabilities, we obtain formulas to update the regression parameters of a model for all possible one knot additions and subtractions, without computing the entire regression from scratch.

Let  $\hat{\beta}_\gamma$  be the current least squares estimator for  $m_\gamma$ ,  $\mathbf{K}_\gamma = [\mathbf{X}, \mathbf{K}_1, \mathbf{K}_\gamma]$  be the design matrix for  $m_\gamma$  with  $p_\gamma$  columns, and  $\Sigma_\gamma = (\mathbf{K}_\gamma^T \mathbf{K}_\gamma)^{-1}$ . First the updating rule will be derived for subtraction moves, i.e. for  $m_{\gamma-} \in N_-$ . Without loss of generality, permute the columns of  $\mathbf{K}_{\gamma-}$  such that the knot being removed is in the last column of the design matrix. We first partition  $\Sigma_\gamma$  in the following manner:

$$\Sigma_\gamma = \begin{pmatrix} \Sigma_- & \tilde{\Sigma}_2 \\ \tilde{\Sigma}_2^T & \Sigma_{22} \end{pmatrix}.$$

Then the updates can be computed as

$$\boldsymbol{\Sigma}_{\gamma-} = \boldsymbol{\Sigma}_- - \frac{\tilde{\boldsymbol{\Sigma}}_2^T \tilde{\boldsymbol{\Sigma}}_2}{\boldsymbol{\Sigma}_{22}}, \text{ and } \hat{\boldsymbol{\beta}}_{\gamma-} = \hat{\boldsymbol{\beta}}_{(-p_\gamma)} - \frac{\tilde{\boldsymbol{\Sigma}}_2}{\boldsymbol{\Sigma}_{22}} \hat{\boldsymbol{\beta}}_{(p_c)}.$$

Next, the updating rule will be derived for addition moves, i.e. for  $m_{\gamma+} \in N_+$ . Let the knot being added be  $s_+$ . The column associated with  $s_+$ ,  $\mathbf{K}_{s_+}$ , will be placed at the beginning of the design matrix. Let  $Q_{11} = \sum_{i=1}^n K(s_i - s_+)^2$ . Note that this can be computed with a subset of the data since our kernels are sparse. Let  $\tilde{\mathbf{Q}}_1 = \mathbf{K}_{s_+}^T \mathbf{K}_\gamma$ . Then

$$\boldsymbol{\Sigma}_{\gamma+} = \begin{pmatrix} 0 & \tilde{\mathbf{0}}^t \\ \tilde{\mathbf{0}} & \boldsymbol{\Sigma}_\gamma \end{pmatrix} + \frac{1}{Q_{11} - \mathbf{Q}_1^T \boldsymbol{\Sigma}_\gamma \mathbf{Q}_1} \begin{pmatrix} 1 & -(\boldsymbol{\Sigma}_\gamma \mathbf{Q}_1)^T \\ -\boldsymbol{\Sigma}_\gamma \mathbf{Q}_1 & \boldsymbol{\Sigma}_\gamma \mathbf{Q}_1 (\boldsymbol{\Sigma}_\gamma \mathbf{Q}_1)^T \end{pmatrix},$$

and letting  $\mathbf{S}_{\gamma+}$  be the first column of  $\boldsymbol{\Sigma}_{\gamma+}$ ,

$$\boldsymbol{\beta}_{\gamma+} = \begin{pmatrix} 0 \\ \hat{\boldsymbol{\beta}}_\gamma \end{pmatrix} + (Q_{11} - \mathbf{Q}_1^T \boldsymbol{\Sigma}_\gamma \mathbf{Q}_1) \mathbf{S}_{\gamma+} \mathbf{S}_{\gamma+}^T.$$

In general, calculating the least squared regression coefficients requires  $O(np^2 + p^3)$  operations, where  $p$  is the number of parameters. We perform this calculation for the first resolution, estimating  $p = J(1) + q$  coefficients. Adding one additional knot requires  $O(n(p+1) + (p+1)^2)$  operations. This is significantly faster than calculating the regression naively for each model.

## 2.2.5 Prediction and interval estimation

To get predictions that account for model uncertainty, we use Bayesian model averaging over the top knot configurations. Let the top  $Q$  configurations of knots

found be  $M = \{m_1, \dots, m_Q\}$  with marginal model probabilities  $\{p_1, \dots, p_Q\}$ . Correspondingly, consider their  $R^2$  values,  $\{R_1^2, \dots, R_Q^2\}$ , least squared estimates of the coefficient vectors,  $\hat{\beta}_1, \dots, \hat{\beta}_Q$ , least squared estimates of the error variance,  $\{\hat{\sigma}_1^2, \dots, \hat{\sigma}_Q^2\}$ , the covariance matrices of the estimates  $V_1, \dots, V_Q$ , and the number of knots,  $\{b_1, \dots, b_Q\}$ . For prediction at a point  $s_{new}$  in the spatial field, Bayesian model averaging works as follows:

1. For each of the  $Q$  knot configurations, calculate the values of the kernel functions at  $s_{new}$ . Denote them as  $\{\mathbf{k}_{new,1}, \dots, \mathbf{k}_{new,Q}\}$ .
2. Using each of the  $Q$  kernel function vectors, calculate the expected value  $\mathbb{E}(y(s_{new})|m_i)$  for each  $m_i \in M$ . For the hyper-g prior, we have that

$$\mathbb{E}(y(s_{new})|m_i) = E\left(\frac{g}{1+g} \middle| m_i\right) \mathbf{k}_{new,i}^T \hat{\beta}_i,$$

where

$$E\left(\frac{g}{1+g} \middle| m_i\right) = \hat{s} = \frac{2}{p_i + a_g} \frac{{}_2F_1(.5(n-1), 2, .5(p_i + a_g), R_i^2)}{{}_2F_1(.5(n-1), 1, .5(p_i + a_g), R_i^2)}.$$

3. The Bayesian model averaging estimate is

$$y_{new}^*(s) = \frac{\sum_{i=1}^Q \mathbb{E}(y(s)|m_i) \times p_i}{\sum_{i=1}^Q p_i}.$$

In practice, the largest of the posterior model probabilities is usually much larger than the others, so the averaging step is not always necessary. For intervals, the same averaging procedure can be used, but instead of using the expected value, we use the quantiles of the posterior predictive distribution. Since the posterior predictive distribution under the hyper-g prior is not analytically available, we use the plug in estimator for the shrinkage factor,  $\hat{s}$ , from step 2 above. Then,



conditional on the plug in estimator, the posterior predictive distribution is

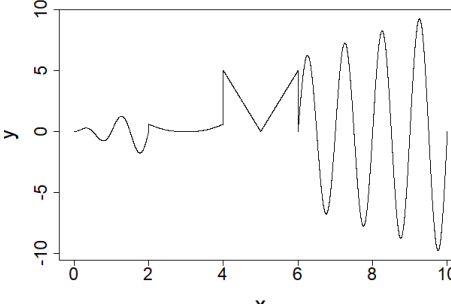
$$p(y(s_{new})|m_i) \sim T_{n-p}(\mathbb{E}(y(s_{new})|m_i), \hat{\sigma}_i^2(1 + k_{new,1}(V_i)k_{new,1})).$$

## 2.3 Assessing the proposed model

We assess the predictive accuracy and runtime of the model using a holdout set when changing the values of parameters that affect model size, model fit, and the smoothness of the predicted surface. Specifically, we vary the prior sparsity parameters  $a_\pi$  and  $b_\pi$ , the size  $J(1)$  of the first resolution grid, the kernel width  $\tau$ , and the kernel smoothness  $\nu$ . For each of a number of simulated datasets and parameters, we fit an MSSS with a 10% randomly chosen holdout group, and quantify the predictive accuracy for the different parameter combinations. In addition, we compare the performance and runtime of our model to that of other multi-resolution models that we were able to implement. There are many possible competing models (see, Heaton et al., 2019, for example), but here we limit ourselves with models that have a multi-resolution structure. We focus on the model proposed in Nychka et al. (2015), abbreviated as LK, for which the R package `LatticeKrig` (Nychka et al., 2016) is available, and the multiresolution process convolution model of Guhaniyogi and Sansó (2017), referred to as MDCT. To demonstrate how the multi-resolution process convolution models behave differently than single resolution models, we will also compare the model with a single resolution process convolution, abbreviated as DPC, with a varying number of resolution knots.

Another natural competitor is the model in Katzfuss (2017), abbreviated as MRA. Code for implementing this model on two dimensional spatial fields is available in the R package `GPVecchia`, which is available on Github. However, we

**Table 2.1:** Equation for and plot of the mean function for the nonstationary 1D example data.

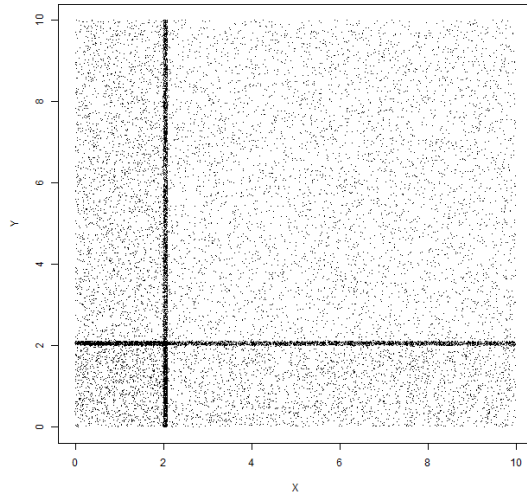
Function	Plot
$f(x) = \begin{cases} \sin(2\pi x) + 5 & \text{if } 0 \leq x < 2 \\  \sin(x - 3) ^3 + 5 & \text{if } 2 \leq x < 4 \\ 5 x - 5  + 5 & \text{if } 4 \leq x < 6 \\ \sin(2\pi x)x + 5 & \text{if } 6 \leq x < 10 \end{cases}$	

were unable to do parameter estimation using this model due to instabilities in the likelihood evaluation. As a compromise, for the two synthetic two-dimensional datasets discussed in section 2.3.1, the correct Gaussian process parameters were passed to the package. Therefore, the predictions and interval estimations we use from this package do not account for parameter uncertainty, and the timings do not account for estimation of the parameters.

### 2.3.1 The datasets

Our first example consists of a one dimensional piecewise function that is meant to demonstrate the flexibility of our method in tackling highly nonstationary processes, and was used in Guhaniyogi and Sansó (2017). We generated one example with 20,000 observations from the mean curve, and added  $N(0,1)$  noise. Plots and details of the function are presented in table 2.1.

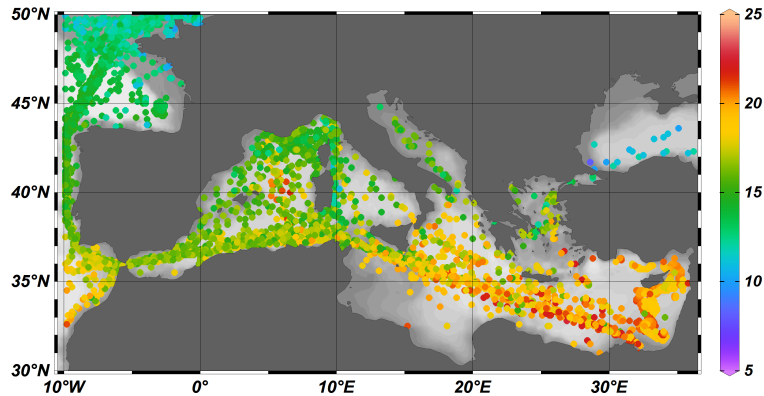
The next three simulated datasets consist of two-dimensional fields. The first two were generated from stationary Gaussian processes with Matern covariance functions using the `RandomFields` package (Schlather et al., 2017a) on the interval



**Figure 2.2:** Sample locations for the 3 simulated 2D datasets

$[0, 10] \times [0, 10]$ . For the first of these examples, which we will denote as dataset A, the scale parameter was 1 and the smoothness parameter  $\psi = 2$ . For the second example, which we will denote as dataset B, the scale parameter was 1 and  $\psi = .5$ . Dataset B is continuous, but non-differentiable. For both, 20,000 observations were sampled from unequally spaced locations, and random noise with variance .1 was added to the generated data. The third example, which we will denote as dataset C was generated from the nonstationary kernel convolution model of Lemos and Sansó (2009) using a 9 by 9 grid of kernels that are rotated differently across the space. This makes for a very smooth, nonstationary field. The same unequally spaced sampling and variance of .1 were used. The unequal spacing is displayed in the figure 2.2 material, and the fields are displayed with the results in figure 2.5.

The last example corresponds to 12,210 temperature in situ measurements from the Mediterranean Sea during the month of December 2003. These data are obtained from four different types of devices, namely: buckets launched from navigating vessels; readings from the water intake of ship's engine rooms; moored



**Figure 2.3:** December 2003 Mediterranean sea surface temperature observations in Celsius.

buoys; and drifting buoys. The result is a set of very unequally spaced, with many observations taken along shipping lanes, and large areas of the ocean scarcely covered by the sampling. In addition, it is known that the complexity of the shapes of the coastlines and the action of the currents, produce a very heterogeneous field of temperatures.

### 2.3.2 Parameter settings and competitor details

For each of the examples discussed in the previous section we implemented MSSS with an intercept term, and a number of different parameter settings under a fully crossed design, resulting in 243 total runs. For prior sparsity, kernel size, and kernel smoothness, the parameter settings are listed in table 2.2. The number of knots at the first resolution was varied between 10, 15, and 30 in the one dimensional example, and 42, 132 and 272 in the two-dimensional simulated examples, and 91, 312, and 663 in the SST data example. These knots were placed in a grid across the domain being modeled, including a small buffer region to minimize edge effects. For example, in the one dimensional example, the domain was  $(0, 10)$ , and the kernels were placed between  $-0.5$  to  $10.5$ . For each setting of parameters, the

top 100 models were stored for creating the prediction and intervals described in 2.2.5. Since we have run hundreds of different configurations of MSSS parameters, in the numerical summaries, we will show the best, worst, and median result for each individual statistic, and in the graphical summaries, we will show plots of the best and worst of the MSSS models measured by the top posterior model probability.

$\theta$	$\mu$	$\tau$	$\nu$
1	.1	1.5	1
5	.2	2	2
10	.5	2.5	3

**Table 2.2:** Different parameter settings for the prior used in the fully crossed design for the simulation study.  $\mu = \frac{a_\pi}{a_\pi + b_\pi}$  and  $\theta = a_\pi + b_\pi$ .

MSSS was implemented in C++ using OpenMP to take advantage of the parallel nature of the stochastic search. All data preparations were done in R, and the RCPP package was used to pass information from R to C++. The code was run on a Unix machine using 10 Intel Xeon E5-4650 processors and 16 gigabytes of RAM.

The competitor models were run in Microsoft R Open using the Intel multi-ple kernel library on a Windows desktop with an Intel i7-2600k processor with 4 cores and 16 gigabytes of ram. This i7 processor performs single core operations more quickly than the Xenon. The single resolution DPC competitor model was implemented using MCMC with a kernel width of 1.5 times the distance between knots under independent priors, and run for 10,000 iterations with varying numbers of resolution 1 knots. The MDCT competitor model of Guhaniyogi and Sansó (2017) was implemented using MCMC and run for 10,000 iterations under varying numbers of first resolution knots and three resolutions, as recommended by the authors. The competitor model in Nychka et al. (2015) was implemented using the `LatticeKrig` package with varying number of first resolution basis functions and

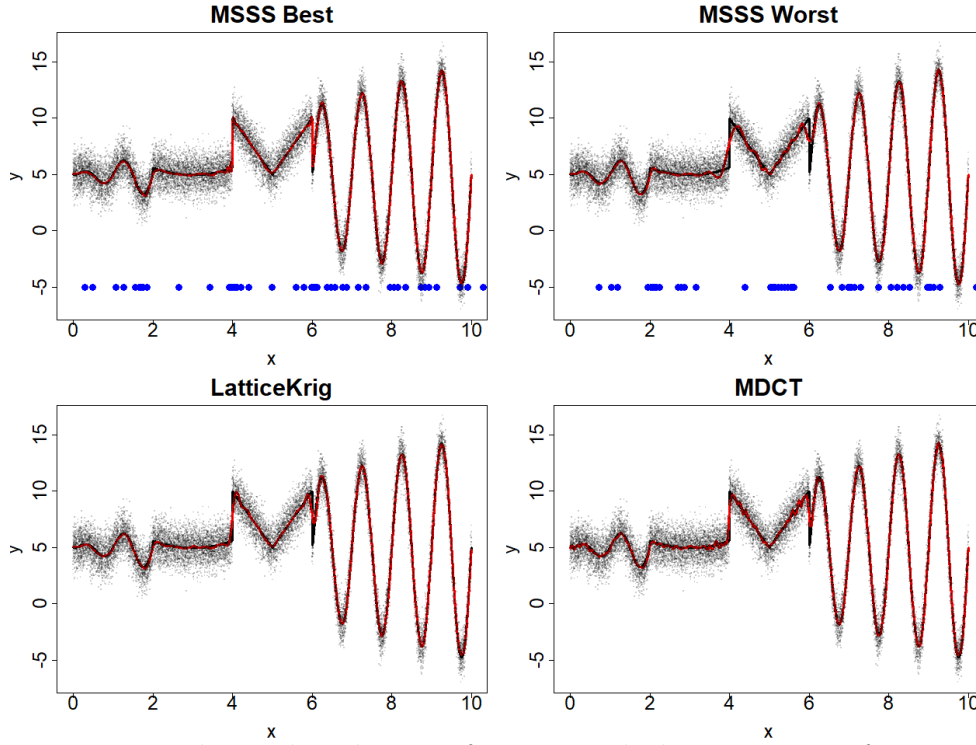
three levels. The  $\nu$  parameter was set to 1, which induces a variance that decreases sharply as the resolution increases. The parameter `a.wght` in the `LatticeKrig` package was set to 4.1, as recommended by the authors to approximate a thin plate spline while reducing edge effects. The competitor model in Katzfuss (2017) was implemented using the `GPVecchia` package, with the covariance parameters set to the correct values in the GP models. The three approximation parameters, M, R and J, were set to 3, 4, and 3 respectively, which results in 89 basis functions.

### 2.3.3 Results

For the one dimensional example, plots of the estimated mean function under the different models are shown in figure 2.4, and numerical results are found in table 2.3. In this setting, MSSS worked better than other models when  $J(1)$  was large, and the kernel width  $\tau$  was small. When a small number of wide, smooth kernels were used, the MSPE increased to as high as 1.1, but the interval coverage was still very close to .9. Given the piecewise smoothness of the function, this finding is not surprising. It is also clear from the kernel locations that a large number of knots are added near the locations with non-differentiability of the mean function. In fact, around those points, MSSS used as many as eight resolutions to account for the rapid change in the mean of the data. This corresponds to our intuition, as MSSS explicitly captures local, high frequency change that occurs at those points.

The MDCT also performed quite well, as long as enough first resolution kernels were used, though the limited number of resolutions caused some lack of fit at the points where the mean function changes abruptly. Despite very good results with respect to the estimation of the mean function, `LatticeKrig` had prediction interval coverage that was higher than the intended confidence level. This behav-

ior repeats is also observed in the two-dimensional examples, and reinforces the empirical findings of Heaton et al. (2019), where `LatticeKrig` demonstrated the same characteristics.



**Figure 2.4:** Red: predicted mean function; Black: true mean function; Blue dots: kernel locations for MSSS; Gray dots: observed data. All models fit the true mean function pretty well over most of the domain. However only the MSSS with smaller kernel width fit the data well at the discontinuities ( $x=4$  and  $x=6$ ).

For the two-dimensional simulated examples, predicted surfaces are displayed in figure 2.5. Numerical summaries for the case  $\psi = 2$ , that corresponds to a smooth random field, are presented in table 2.4. Results for the case  $\psi = 1$ , that corresponds to a jagged random field, are presented in table 2.5 cases are presented in the appendix. For  $\psi = 2$ , all models, with the exception of the DPC with the fewest kernels, showed good predictive performance. Interval estimation was also good for all of the models save for `LatticeKrig`, which showed some over-coverage. `LatticeKrig` and the DPC models all were very fast, as was the

Model	MSPE	90% Coverage	Runtime (sec)
MSSS Min	.99	.89	4
MSSS Med	1.02	.90	13
MSSS Max	1.10	.91	76
MDCT 10	1.17	.9	381
MDCT 20	1.02	.9	518
LK 10	1.08	.94	102
LK 20	1.03	.95	105
LK 40	1.00	.95	107
DPC 10	15.30	.84	269
DPC 100	1.17	.89	419
DPC 1000	1.13	.89	911

**Table 2.3:** Numerical summaries for the different models in the one-dimensional example. MSSS always provides excellent predictive interval coverage, and gives excellent out-of-sample fit when either enough kernels at the first resolution are used or the kernels are of the appropriate shape.

MSSS with appropriate kernel settings (smooth and wide). However, when using very narrow kernels, MSSS took a long time to converge, requiring a large number of fairly dense resolutions to produce a smooth response surface. MDCT was also quite slow because of the relatively complex MCMC required.

Model	MSPE	90% Coverage	Runtime (sec)
MSSS Min	.11	.87	372
MSSS Med	.11	.89	1198
MSSS Max	.12	.90	3596
MDCT 42	.11	.88	1853
MDCT 132	.11	.86	5126
LK 10	.11	.94	249
LK 20	.11	.94	493
DPC 42	.30	.90	570
DPC 132	.17	.90	800
DPC 462	.11	.89	673
MRA	.11	.88	56

**Table 2.4:** Numerical summaries for the models on the two-dimensional GP dataset with a Matern correlation kernel  $\psi = 2$ . Notice that runtimes for MRA do not include parameter estimation.

Every model struggled with the extremely jagged GP that is produced when



Model	MSPE	90% Coverage	Runtime (sec)
MSSS Min	.2	.88	587
MSSS Med	.22	.89	3606
MSSS Max	.25	.90	15016
MDCT 42	.21	.89	2733
MDCT 132	.18	.86	4923
LK 10	.19	.96	271
LK 20	.17	.95	557
DPC 42	.47	.91	469
DPC 132	.32	.90	677
DPC 462	.24	.89	941

**Table 2.5:** Numerical summaries for the models on the 2D Gaussian Process dataset with a Matern correlation kernel,  $\psi = .5$ .

Model	MSPE	90% Coverage	Runtime (sec)
MSSS Min	.10	.90	36
MSSS Med	.10	.90	156
MSSS Max	.10	.91	566
MDCT 42	.10	.89	2975
MDCT 132	.10	.87	4716
LK 10	.10	.95	298
LK 20	.10	.95	660
DPC 42	.14	.91	400
DPC 132	.10	.90	579
DPC 462	.10	.90	1036

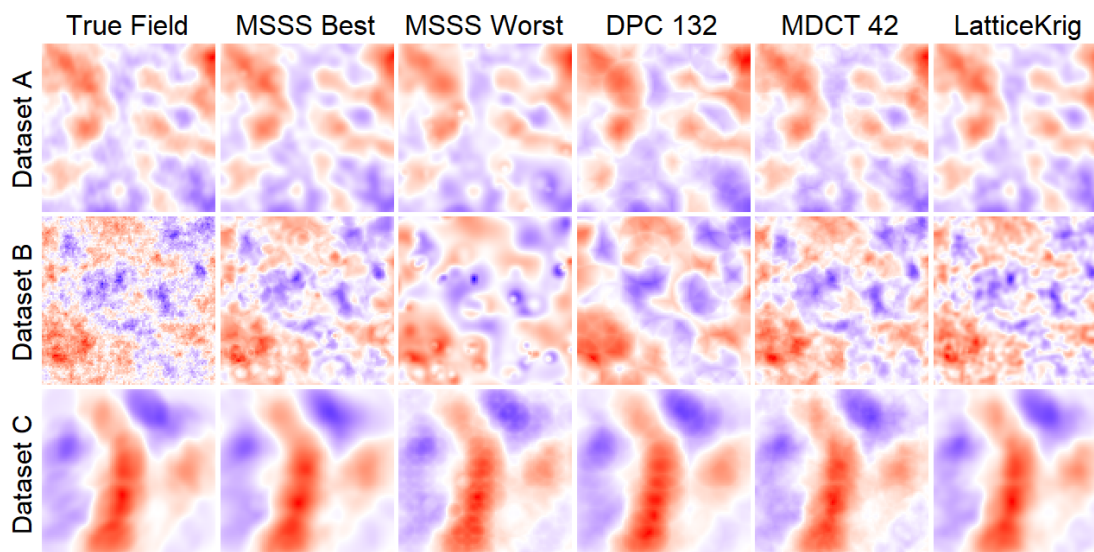
**Table 2.6:** Numerical summaries for the models on the 2D nonstationary kernel convolution dataset.

$\psi = .5$ . The MSPE was much higher than the true variance (which was .1) for every model. However, MSSS, the DPCs, and the MDCT all did well in interval coverage. The best of the MSSS models, which had the largest number of initial kernels and the least smooth basis functions, did particularly well both in prediction and interval coverage. Similar computational results to the smooth GP were observed, with fast performance for single resolution DPC's, `LatticeKrig`, and well specified MSSS, and long runtimes for the MDCT and the misspecified MSSS. All of the models performed well in fitting the simulations from the nonstationary kernel convolutions (table 2.6), with good MSPE for every model except

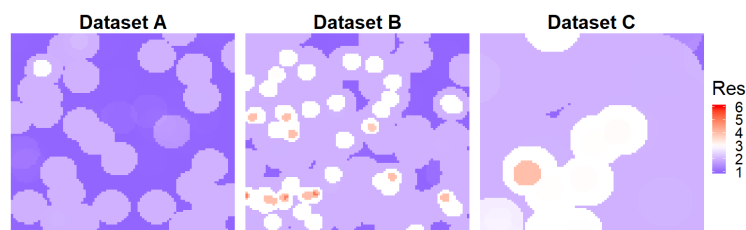
for the DPC with only 42 kernels, and excellent coverage probabilities for all but the `LatticeKrig`. It is worth mentioning that for this surface, the MSSS with a very large number of knots at the first resolution sometimes added no knots at all, which is a desirable behavior when one resolution is sufficient.

The spatially varying resolution created by MSSS allows for an additional visualization. We can plot the posterior average number of resolutions active at each point in the space, as seen for the best MSSS fit in figure 2.6. Note that, as we are using model averaging over the top 100 models, this quantity can be a fraction. The figure provides information about the regions of the space where there is fine scale variation. The smooth, stationary GP with  $\psi = 2$  requires fewer resolutions than the jagged GP with  $\psi = .5$ . The stationarity in these datasets is reflected by a similar pattern in resolutions across the space. In other words, there is not a single area in the space where the resolution is much higher than in other places. When MSSS is fit to the nonstationary kernel convolution, the behavior is quite different. The number of resolutions required is different across the space, with just one section of the space requiring three or four, while the vast majority just requires two.

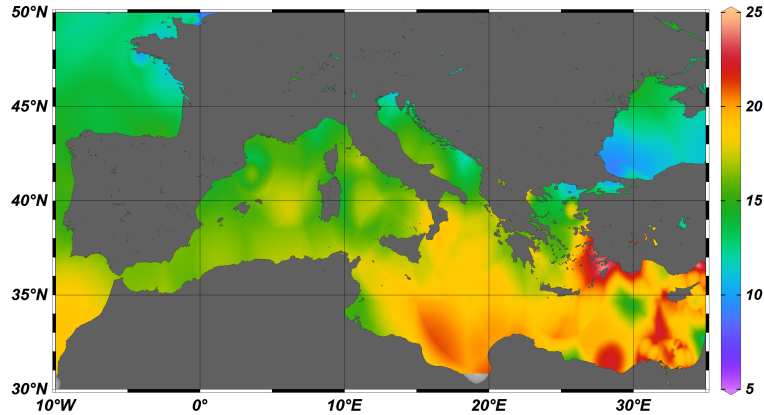
For the SST data we observe a very unequal distribution of the observation's locations. To avoid numerical instabilities and ensure reasonable ranges for out of sample predictions we fitted MSSS requiring that kernels were only allowed to enter the model if there was at least one data point within one kernel width from the center of the kernel. As for the other settings, we set  $\nu = 3$  and the kernel width to 2.5 since for sea surface temperature, we expect a relatively smooth mean function. The SST estimates are shown in figure 2.7, and the number of resolutions at each point is shown in figure 2.8. The plot of the number of resolutions at each point in the Mediterranean identifies regions with temperatures that vary



**Figure 2.5:** Two-dimensional simulated and predicted surfaces on the unit square, one row per dataset. First column is the true surface, and each additional column corresponds to the predicted mean surface from the considered models.



**Figure 2.6:** Plots of the maximum resolution (Res) active at each point on the unit square for the best MSSSS model by marginal model probability for each of the three simulated two-dimensional datasets. For the two stationary examples, datasets A and B, the pattern in the multi-resolution structure does not change greatly across the domain.

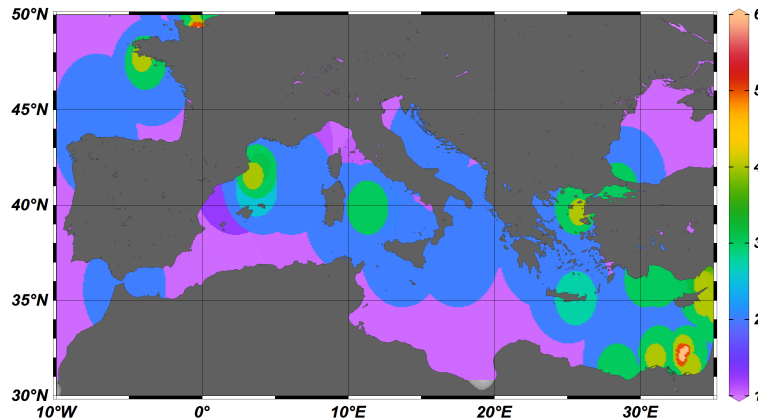


**Figure 2.7:** Predicted SST in Celsius with  $\nu = 3$ , a kernel width of 2.5, and the additional restrictions discussed above

differently than the surrounding areas. Some areas with higher resolutions include the region between Palma and Sardinia, which is warmer than its surroundings, the region adjacent to the Brittany peninsula on the northwest end of the dataset that is colder than its surroundings, and the southeast end of the Mediterranean, which has a large amount of temperature variation in the observed data, with observations varying between 15 and 23 degrees in a very small region. Numerical results are in table 2.7. MSSS and the MDCT with 42 kernels were the only models with both low MSPE and well calibrated interval coverage. Unlike the GP examples, predictions were substantially better using MSSS when compared to `LatticeKrig` or the MDCT.

Model	MSPE	90% Coverage	Runtime (sec)
MSSS	.63	.89	235
MDCT 42	.70	.88	1020
MDCT 132	1.58	.86	1505
LK 10	.68	.95	162
LK 20	.67	.94	204
DPC 91	.99	.88	309
DPC 312	.88	.89	433
DPC 1144	1.12	.87	717

**Table 2.7:** Numerical summaries for the models on the SST dataset.



**Figure 2.8:** Plot of the maximum resolution active at each point on the surface for the MSSS model on SST in the Mediterranean.

### 2.3.4 Default parameters

The results obtained in our data analysis lead to some guidelines for the selection of the parameters of the MSSS. First, the different parameters used in the beta-binomial prior on  $\gamma$  do not change the resulting surface or sparsity substantially, unless very extreme values are used. Therefore, we propose setting  $\mu = 1/2^d$  and  $\theta = 2$  as a safe default for data of the size that was dealt with here.

The remaining parameters  $\tau$ ,  $J(1)$ , and  $\nu$  can be set by maximizing the predictive distribution over a grid of possible values. For large datasets such strategy can impose a steep computational cost. For the kernel parameters we require that  $\tau > 1.5$ . This ensures enough kernel overlap to prevent gaps. Beyond this strict restriction, the ability of MSSS to include an unlimited number of resolutions provides some robustness with respect to  $\tau$  and  $J(1)$ . This is demonstrated in the simulation study, where the MSPE does not change very much among the different settings. For example, if  $J(1)$  is not large enough to fit the data well, MSSS is able to add more kernels at high resolution to compensate for the lack of fit. Some attention must be paid, though, to the smoothness parameter  $\nu$ , as the shape of the resulting predicted surface can be highly dependent on this

parameter. However, specific knowledge of the application can inform the choice of  $\nu$ . For example, in the SST dataset, it would be unreasonable for a predicted field to be very jagged, so a larger value of  $\nu$  is preferable.

## 2.4 Discussion

We have proposed a novel method that leverages Bayesian variable selection and model averaging to fit nonstationary spatial models. A stochastic process prior on the tree structure created by a recursive partition of the domain achieves spatially varying resolution for the resulting predictive field. By avoiding MCMC, utilizing sparse matrix methods, using efficient formulas to update regression coefficients when one column is added or deleted, and taking advantage of modern parallel computing, MSSS shows competitive computational performance, when compared to other multi-resolution spatial methods. We have also shown that MSSS provides competitive out of sample fit and uncertainty quantification on a variety of unequally spaced spatial datasets, both stationary and non-stationary. Estimation of the spatially varying resolution enforced by MSSS allows for simple and explicit identification of non-stationarity in spatial datasets, which can have physical meaning in the context of specific applications.

# Chapter 3

## Multi-Scale Spatial Optimization

### 3.1 Introduction

In the previous chapter, we formulated a prior that achieves spatially varying resolution by setting knot coefficients to zero exactly. We then derived an algorithm to explore the space of potential knots, and described how to perform model comparison and averaging on the results. In this chapter, we will develop a prior that results in a posterior maximum that has the property of spatially varying resolution. Samples from the posterior under these priors will *not* be sparse, the sparsity is achieved only at the maximum. This can be interpreted as a penalized optimization method. These methods take the form

$$\min_{\boldsymbol{\beta}} L(\mathbf{y}|\mathbf{X}, \boldsymbol{\beta}) + \lambda P(\boldsymbol{\beta}) \quad (3.1)$$

where  $L$  is a loss function, and  $P$  is a penalty function. The solutions to penalized optimization methods can often be interpreted as maximum a posteriori (MAP) estimates, and in this chapter, the minimum of the penalty and the maximum a posteriori (MAP) of the associated posterior are equivalent, and we will use both

terms interchangeably.

The canonical example of a penalty that induces zeros at its minimum is the least absolute shrinkage and selection operator (LASSO) (Tibshirani, 1996). It has the tendency to select only one of a group of correlated predictors (Zou and Hastie, 2005). This is not compatible with spatially varying resolution, since the coefficients associated with children and parent knots are highly correlated. We address this issue by adapting the composite absolute penalties family (Zhao et al., 2009) to our setting. The prior proposed will be presented in the same multi-resolution setting that has been discussed previously, imposing no limit on the number of resolutions considered. This is challenging due to the fact that most maximization routines require the formation of the entire design matrix. We will adapt the hierarchical multiple kernel learning framework (Bach, 2009) to our setting. Bach (2009) show that under certain conditions sequentially maximizing finite problems in a specific manner is equivalent to maximizing the infinite dimensional problem under certain conditions.

We will first review optimization methods that induce sparsity in different patterns, discuss their Bayesian analogues, discuss the existence of algorithms for finding the maximum, and review consistency results. Each method reviewed will be presented in the context of a spatial, multi-resolution knot structure. Most of the methods we review do not support an infinite number of resolutions, so we will present them with a finite number of resolutions  $R$ . After reviewing these approaches, we will develop a prior that results in spatially varying resolution at its maximum a posteriori estimate. In the same manner as discussed in previous chapters, we will default to having all resolution 1 coefficients be nonzero.



### 3.1.1 LASSO

Let the response  $\mathbf{y} \in \mathbb{R}^n$ , the kernel matrix  $\mathbf{K} \in \mathbb{R}^{n \times \sum_{r=1}^R J(r)}$ , the coefficients associated with the kernels  $\boldsymbol{\beta} \in \mathbb{R}^{\sum_{r=1}^R J(r)}$ , the design matrix  $\mathbf{X} \in \mathbb{R}^{n \times p}$ , and the fixed effect coefficients  $\boldsymbol{\alpha} \in \mathbb{R}^p$ . In this setting, the standard linear model would be

$$\mathbf{y} \sim N(\mathbf{K}\boldsymbol{\beta}, \sigma^2).$$

As discussed in chapter 1, spatially varying resolution is only coherent if sparsity is permitted only at resolutions above 1. Therefore, in the context of a multi-resolution knot structure, the LASSO penalized optimization problem is

$$\min_{\boldsymbol{\beta}} (\mathbf{y} - \mathbf{X}\boldsymbol{\alpha} - \mathbf{K}\boldsymbol{\beta})^t (\mathbf{y} - \mathbf{X}\boldsymbol{\alpha} - \mathbf{K}\boldsymbol{\beta}) + \lambda \sum_{r=2}^R \sum_{j=1}^{J(r)} |\beta_j^r|. \quad (3.2)$$

This is analogous to a double exponential prior on the knot coefficients for knots at resolutions above 1, and an improper flat uniform prior on all other coefficients, i.e.

$$p(\beta_j^r) \stackrel{\text{i.i.d.}}{\sim} \frac{\lambda}{2} \exp(-\lambda|\beta_j^r|) \text{ for } r > 1.$$

This prior induces sparsity at its maximum, where some coefficients are set to zero exactly. This maximum can be calculated efficiently for a number of  $\lambda$ , often called the solution path, via a number of different maximization routines. The LARS algorithm (Efron et al., 2004) is commonly used. This algorithm computes all possible LASSO solutions, and works by increasing predictors that are correlated with residuals of smaller models. The computational speed is partially due to the continuity of the LASSO solution. If we let  $\hat{\boldsymbol{\beta}}(\lambda)$  be a solution to the LASSO for  $\lambda$ , then the function  $\hat{\boldsymbol{\beta}}(\lambda)$  is continuous. Therefore  $\hat{\boldsymbol{\beta}}(\lambda)$  is an excellent starting value for gradient descent when finding  $\hat{\boldsymbol{\beta}}(\lambda + \epsilon)$ .

A number of results on variable selection and prediction consistency under different conditions have been proven in the years since the LASSO was proposed. Of particular interest is Bickel et al. (2009), which shows that under some sparsity and regularity conditions, LASSO has sparsity oracle properties for prediction loss in nonparametric regression. They prove bounds for prediction loss in terms of the best possible approximation of a function under the sparsity constraint induced by LASSO.

In Park and Casella (2008), the authors point out that the above prior can result in a bimodal posterior, and propose the Bayesian Lasso, which modifies this prior to be conditional on the variance, i.e.

$$p(\beta_j^r | \sigma^2) \stackrel{\text{i.i.d.}}{\sim} \frac{\lambda}{2\sigma} \exp\left(-\lambda \frac{|\beta_j^r|}{\sigma}\right),$$

which guarantees unimodality. The authors also provide a mixture representation for this prior that permits Gibbs sampling to simulate from the posterior distributions of  $\sigma^2$  and  $\beta$ . The authors provide a conjugate prior for  $\lambda$ , but recommend instead using Empirical Bayes to set this parameter.

The LASSO estimator does not satisfy our desiderata for spatially varying resolution, where  $\beta_j^r = 0 \implies \text{children}(\beta_j^r) = 0$ , and the shrinkage and sparsity induced by the LASSO does not increase in resolution. Modifications can be made that increase the penalty as the resolution increases. For example, this simple generalized lasso (Tibshirani et al., 2011)

$$\min_{\beta} (\mathbf{y} - \mathbf{X}\boldsymbol{\alpha} - \mathbf{K}\boldsymbol{\beta})^t (\mathbf{y} - \mathbf{X}\boldsymbol{\alpha} - \mathbf{K}\boldsymbol{\beta}) + \sum_{r=2}^R \lambda_r \sum_{j=1}^{J(r)} |\beta_j^r|, \quad (3.3)$$

with  $\lambda_r > \lambda_{r-1}$  results in a larger penalty for higher resolutions. Unfortunately, this prior would not directly enforce or encourage spatially varying resolution at

its maximum, and the sparsity would not vary in space. A modified version of the LASSO that applies different penalties to different parameters is the spike and slab lasso (Ročková and George, 2016), and appears promising at first glance. Let  $P = \sum_{r=2}^R J(r)$ ,  $\psi(x|\lambda)$  be the double exponential distribution evaluated at  $x$  with parameter  $\lambda$ ,  $\lambda_1 < \lambda_0$ , and  $p_\theta^*(\beta_j^r) = \frac{\theta\psi(\beta_j^r|\lambda_1)}{\theta\psi(\beta_j^r|\lambda_1) + (1-\theta)\psi(\beta_j^r|\lambda_0)}$ . Then the spike and slab lasso (SSL) is a penalty of the form

$$\min_{\beta} (\mathbf{y} - \mathbf{X}\alpha - \mathbf{K}\beta)^t (\mathbf{y} - \mathbf{X}\alpha - \mathbf{K}\beta) - \lambda_1 \sum_{r=2}^R \sum_{j=1}^{J(r)} |\beta_j^r| + \log \left[ \frac{\int \frac{\theta^P}{\prod_{r=2}^R \prod_{j=1}^{J(r)} p_\theta^*(\beta_j^r)} d\pi(\theta)}{\int \frac{\theta^P}{\prod_{r=2}^R \prod_{j=1}^{J(r)} p_\theta^*(0)} d\pi(\theta)} \right]. \quad (3.4)$$

This penalization method adaptively shrinks to zero some  $\beta_j^r$  via the sharp spike induced by the large  $\lambda_0$ , and allows larger effects to be penalized less due to the fat tailed slab induced by  $\lambda_1$ . The authors show in the supplemental material that penalized optimization is equivalent to maximum a posteriori inference under the Bayesian finite mixture model

$$\mathbf{y}|\beta \sim N_n(\mathbf{K}\beta, \sigma^2) \quad (3.5)$$

$$p(\beta_1) \propto 1 \quad (3.6)$$

$$p(\beta_j^r|\gamma_j^r) = \gamma_j^r L(0, \lambda_1) + (1 - \gamma_j^r) L(0, \lambda_0) \quad (3.7)$$

$$Pr(\gamma_j^r = 1|\theta) = \theta \quad (3.8)$$

$$\theta \sim p(\theta). \quad (3.9)$$

Using this model as a starting point, a promising enhancement that appears to strongly encourage spatially varying resolution can be built. This is accomplished by hierarchical assignment to the spike and slab in a manner that respects the

tree structure by replacing equation (3.8) with

$$Pr(\gamma_j^r = 1 | \gamma_{\lfloor \frac{j-1}{p} \rfloor + 1}^{r-1} = 1) = \theta, \quad (3.10)$$

$$Pr(\gamma_j^r = 1 | \gamma_{\lfloor \frac{j-1}{p} \rfloor + 1}^{r-1} = 0) = 0. \quad (3.11)$$

We will term this the Tree SSL. This prior will result in spatially varying shrinkage in a manner similar to Benedetti et al. (2018), where different regions of the space are assigned a different amount of shrinkage. At its maximum, the Tree SSL does not enforce spatially varying resolution, but appears to encourage it a priori. If a parent is in the spike, then all of its children must be as well, and variables assigned to the spike should be much more likely to be set to zero at the MAP. Unfortunately, properties of the LASSO work against this hierarchy. If a group of covariates are correlated, the LASSO tends to select only one of them (Zou and Hastie, 2005). In the context of multi-resolution basis function sets, the covariates associated with parents and children are very highly correlated because their domains are almost entirely overlapping. This means that, in our setting, the maxima Tree SSL results in sparsity patterns that do not meet the definition of spatially varying resolution, and in fact do not even encourage it very much. Since the Tree SSL, is not sufficient to induce spatially varying resolution, we will now review priors that enforce sparsity in groups of variables together.

### 3.1.2 Group Lasso

The group lasso (Yuan and Lin, 2006) extends the LASSO in a manner that can set groups of coefficients to zero together. If  $\beta_{gl}$  is the vector of coefficients corresponding to group  $gl$ , and there are  $L$  groups total, then the optimization problem is

$$\min_{\beta} (\mathbf{y} - \mathbf{X}\alpha - \mathbf{K}\beta)^t (\mathbf{y} - \mathbf{X}\alpha - \mathbf{K}\beta) + \lambda \sum_{l=1}^L \sqrt{\beta_{gl}^t \beta_{gl}}. \quad (3.12)$$

This allows for correlated variables to be selected together as long as the groups are specified a priori. A fundamental condition for the efficient computation of the solution is that groups are non-overlapping. Under these conditions, the authors discuss how the solution path can be computed using LARS. In addition, as long as the groups do not overlap, this solution is still continuous in  $\lambda$ . However, once the group overlap, this continuity disappears, and other algorithms are required for the fitting of this model, such as proximal methods or the algorithms discussed in (Bach, 2008). Variable selection consistency results for the Group Lasso in a nonparametric setting are proven in Bach (2008) under general group structures.

A grouped extension to the Bayesian formulation of Park and Casella (2008) in the case of non-overlapping groups was developed in Kyung et al. (2010). They propose the prior

$$p(\beta_{gl}|\sigma^2) \stackrel{\text{i.i.d.}}{\sim} \exp\left(-\frac{\lambda}{\sigma} \sqrt{\beta_{gl}^t \beta_{gl}}\right)$$

and provide a mixture representation for this prior that allows for Gibbs sampling, and an Empirical Bayes procedure for selection the optimal  $\lambda$ .

### 3.1.3 Composite Absolute Penalties

An extension of the Group Lasso called Composite Absolute Penalties (CAP) (Zhao et al., 2009) supports a general directed graph structures for the inclusion of variables in a hierarchical manner, which is what is required for spatially varying resolution with a finite  $R$ . The authors show that hierarchy in group variable

selection at the maximum can be enforced if all descendants of a variable are included in its group. For example, if the goal is for group  $B$  to be selected only if group  $A$  has been selected, then the two groups included in the penalty should be  $A' = A \cup B$  and  $B$ . To adapt this to our setting, if we let  $\beta_{t_j^r}$  be the vector of coefficients corresponding to the subtree with root  $\beta_j^r$ , then the penalty defined by

$$\min_{\beta} (\mathbf{y} - \mathbf{X}\alpha - \mathbf{K}\beta)^t (\mathbf{y} - \mathbf{X}\alpha - \mathbf{K}\beta) + \lambda \sum_{r=2}^R \sum_{j=1}^{J(r)} \|\beta_{t_j^r}\|_q. \quad (3.13)$$

will enforce our inclusion hierarchy. This is because, for all  $r$  and  $j$ , the descendants of  $\beta_j^r$  are included in  $\beta_{t_j^r}$ . If we let  $\beta_{-1}$  be  $[\beta_2^t, \dots, \beta_{\mathbf{R}}^t]^t$ , then the analogous prior distribution is of the form

$$p(\beta_{-1}) \propto C(\lambda) \exp \left( -\lambda \sum_{r=2}^R \sum_{j=1}^{J(r)} \|\beta_{t_j^r}\|_q \right),$$

By the factorization theorem, if two knots do not have a common ancestor, then they will be independent under this prior. This induces prior independence in knots that are far from each other, and dependence within a subdomain, which is very sensible in the spatial multi-resolution context. However, adjacent knots at the same resolution that are part of different subtrees will not have any dependence enforced, which could be a drawback. In section 3.2 we partially alleviate this issue by allowing for some dependence across subtrees at the first resolution.

A fully Bayesian implementation of inference under this prior is potentially impossible to implement. This prior has a normalizing constant  $C(\lambda)$  that is extremely difficult to compute, and the authors do not attempt to evaluate it. That is because the structure of the summation in the prior results in individual

coefficients that appear in multiple terms of the summation. In the context of our problem, a coefficient  $\beta_j^r$  will appear in  $r - 1$  different terms of the sum. There is some literature about intractable normalizing constants in the context of MCMC (Liang et al. (2016) and Herbei and Berliner (2014) among others). These methods usually rely on being able to simulate from the intractable distribution. Unfortunately, the mixture representation of this prior that permits simulation runs into the same combinatorial issues, making this approach difficult.

The authors provide an algorithm for finding the maximum a posteriori under this CAP prior when  $q = \infty$ , but this does not preserve the analogy to the Group Lasso presented above. To preserve this analogy, we set  $q = 2$ . Finding the maximum a posteriori when  $q = 2$  is possible using proximal gradient methods. A recent review of proximal methods from the statistical perspective is presented in Green et al. (2015). Proximal methods allow for MAP estimation even when the prior is not differentiable everywhere, which is the case for the CAP prior. A modern, accelerated proximal gradient method that is fast and stable estimating the MAP under CAP penalties is the Fast Iterative Shrinkage Thresholding Algorithm (FISTA), proposed by Beck and Teboulle (2009).

### 3.1.4 Hierarchical Multiple Kernel Learning

The Hierarchical Multiple Kernel Learning (H-MKL) framework of (Bach, 2009) extends CAP to an infinite dimensional, RKHS setting. We adapt the penalty function introduced in Bach (2009) as

$$\min_{\beta} \left( \mathbf{y} - \mathbf{X}\boldsymbol{\alpha} - \sum_{r=1}^{\infty} \sum_{j=1}^{J(r)} (\mathbf{K}_j^r)^t \beta_j^r \right)^t \left( \mathbf{y} - \mathbf{X}\boldsymbol{\alpha} - \sum_{r=1}^{\infty} \sum_{j=1}^{J(r)} (\mathbf{K}_j^r)^t \beta_j^r \right) + \lambda \sum_{r=2}^{\infty} \sum_{j=1}^{J(r)} c_j^r \|\boldsymbol{\beta}_{t_j^r}\|_2. \quad (3.14)$$

This could be thought of as a prior for an infinite dimensional set of coefficients, of the form

$$p(\boldsymbol{\beta}_{-1}) \propto C(\lambda, c) \exp \left( -\lambda \sum_{r=2}^{\infty} \sum_{j=1}^{J(r)} c_j^r \|\boldsymbol{\beta}_{t_j^r}\|_q \right). \quad (3.15)$$

Bach (2009) show that the maximum under this infinite dimensional penalty can be computed by repeatedly maximizing finite problems. In the multi-resolution setting, their algorithm is as follows. First, begin with all resolution 1 and 2 knots. Then compute the MAP estimate for that finite dimensional problem, resulting in some nonzero and some zero coefficients at resolution 2. Next, for the resolution 2 knots that have nonzero coefficients, create their resolution 3 children, and compute the MAP again. This continues until non nonzero children occur

To illustrate, the algorithm would initialize with the CAP penalty for just the second resolution,

$$\min_{\beta} \left( \mathbf{y} - \mathbf{X}\boldsymbol{\alpha} - \sum_{r=1}^2 \sum_{j=1}^{J(r)} (\mathbf{K}_j^r)^t \beta_j^r \right)^t \left( \mathbf{y} - \mathbf{X}\boldsymbol{\alpha} - \sum_{r=1}^2 \sum_{j=1}^{J(r)} (\mathbf{K}_j^r)^t \beta_j^r \right) + \lambda \sum_{j=1}^{J(2)} c_j^2 \|\beta_j^2\|_2. \quad (3.16)$$

That would result in a set of of nonzero coefficients at resolution 2. We would



then consider all resolution 3 coefficients that are children of the  $\hat{\beta}_2$ , and call them  $\tilde{\beta}_3$  and find the maximum under the CAP penalty of this restricted problem. If we let  $J(\tilde{r})$  be the set of  $j$  such that  $\beta_j^r \in \tilde{\beta}_r$  and let  $\tilde{\beta}_2 = \beta_2$ , then we'd maximize

$$\min_{\beta} \left( \mathbf{y} - \mathbf{X}\alpha - \sum_{r=1}^3 \sum_{j \in J(\tilde{r})} (\mathbf{K}_j^r)^t \beta_j^r \right)^t \left( \mathbf{y} - \mathbf{X}\alpha - \sum_{r=1}^3 \sum_{j \in J(\tilde{r})} (\mathbf{K}_j^r)^t \beta_j^r \right) + \lambda \sum_{r=2}^3 \sum_{j \in J(\tilde{r})} c_j^r \|\beta_{t_j^r}\|_2. \quad (3.17)$$

where the  $\beta_{t_j^r}$  are truncated to resolution 3 for the nodes that are active at resolution 2, and truncated at resolution 2 for the nodes that are zero. This continues iteratively, where at each iteration, we update the  $J(\tilde{r})$ ,  $\hat{\beta}_r$ , and  $\tilde{\beta}_r$  until no more nonzero coefficients are added. Bach (2009) show that this iterative method can compute the full, infinite dimensional maximum. They also show that maximizing this penalty achieves selection consistency under some regularity conditions. The  $c_j^r$  are suggested in the paper to be a constant greater than 1 raised to the power of the depth of the graph, which in our case works out to be  $c_j^r = c^{r-1}$  with  $c > 1$ . This condition is required for the consistency result to hold, and is sensible as a way of penalizing models with very high resolution components.

## 3.2 Multi-resolution Spatial Models

We propose to use the penalization method from H-MKL on all resolutions greater than 2, but on the first resolution, use a Gauss Markov random field (GMRF). GMRF's are a natural prior to use on the coefficients of a process convolution because they encourage spatially structured dependence on the coefficients through a sparse precision matrix. We will use the intrinsic GMRF, which

results in dependence on only the nearest neighbors. Let  $\mathbf{W}$  be a  $J(1) \times J(1)$  matrix with

$$W_{ij} = \begin{cases} n_i & \text{if } i = j \\ -1 & \text{if } 0 \leq i \sim j \\ 0 & \text{otherwise} \end{cases}$$

where  $n_i$  is the number of neighbors of knot  $i$  and  $i \sim j$  denotes that knots  $i$  and  $j$  are neighbors. Then, our full model can be written as

$$\mathbf{y}|\boldsymbol{\alpha}, \boldsymbol{\beta}, \sigma^2 = N_n \left( \boldsymbol{\alpha}\mathbf{X} + \sum_{r=1}^{\infty} \sum_{j=1}^{J(r)} (\mathbf{K}_j^r)^t \boldsymbol{\beta}_j^r, \sigma^2 \right)$$

$$p(\boldsymbol{\beta}_1|\tau) = N_{J(1)}(0, \sigma^2\tau^{-1}\mathbf{W}^{-1})$$

$$p(\boldsymbol{\beta}_{-1}|\lambda, c, \sigma^2) \propto C(\sigma^2, \lambda, c) \exp \left( -\frac{\lambda}{\sigma^2} \sum_{r=2}^{\infty} \sum_{j=1}^{J(r)} c^{r-1} \|\boldsymbol{\beta}_{t_j^r}\|_2 \right)$$

$$p(\tau) = \frac{b^a}{\Gamma(a)} \exp(-b\tau)$$

$$p(\boldsymbol{\alpha}, \sigma^2) \propto \frac{1}{\sigma^2}.$$

The GMRF prior that is assigned to resolution one enforces, a priori, some negative correlation between the coefficients corresponding to neighboring knots on the first resolution. This encourages some smoothness in the resulting field. The coefficients associated with resolutions above one receive the infinite dimensional prior introduced in section 3.1.4, which induces sparsity at its maximum in a manner that enforces spatially varying resolution, with shrinkage as the resolution increases.

### 3.2.1 Maximum A Posteriori Estimation

We propose to use the expectation conditional maximization algorithm (Meng and Rubin, 1993) to find the MAP estimate for  $\{\beta, \sigma^2\}$ , while using the E step to integrate out  $\tau^2$ . For fixed  $\tau$ ,  $\sigma^2$ , and  $\lambda$ , the conditional maximum for  $\{\boldsymbol{\alpha}, \boldsymbol{\beta}\}$  can be found using proximal gradient descent. First, we will write this as a penalized optimization problem with a composite penalty, and then discuss the maximization. As an optimization problem, we will need to find

$$\min_{\boldsymbol{\beta}} \frac{1}{2} \left( \mathbf{y} - \mathbf{X}\boldsymbol{\alpha} - \sum_{r=1}^{\infty} \sum_{j=1}^{J(r)} (\mathbf{K}_j^r)^t \beta_j^r \right)^t \left( \mathbf{y} - \mathbf{X}\boldsymbol{\alpha} - \sum_{r=1}^{\infty} \sum_{j=1}^{J(r)} (\mathbf{K}_j^r)^t \beta_j^r \right) + \quad (3.18)$$

$$\frac{1}{2} \tau \boldsymbol{\beta}_1^t \mathbf{W} \boldsymbol{\beta}_1 + \lambda \sum_{r=2}^{\infty} \sum_{j=1}^{J(r)} c^{r-1} \|\boldsymbol{\beta}_{t_j^r}\|_2. \quad (3.19)$$

If we add a small amount of noise to the diagonal of  $\mathbf{W}$ , we can form its Cholesky factor  $\mathbf{Q}$ , and can write this optimization problem as

$$\min_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \frac{1}{2} \left( \mathbf{y} - \mathbf{X}\boldsymbol{\alpha} - \sum_{r=1}^{\infty} \sum_{j=1}^{J(r)} (\mathbf{K}_j^r)^t \beta_j^r \right)^t \left( \mathbf{y} - \mathbf{X}\boldsymbol{\alpha} - \sum_{r=1}^{\infty} \sum_{j=1}^{J(r)} (\mathbf{K}_j^r)^t \beta_j^r \right) + \quad (3.20)$$

$$\frac{1}{2} \tau \|\boldsymbol{\beta}_1\|_{\mathbf{Q}} + \lambda \sum_{r=2}^{\infty} \sum_{j=1}^{J(r)} c^{r-1} \|\boldsymbol{\beta}_{t_j^r}\|_2. \quad (3.21)$$

where  $\|\cdot\|_{\mathbf{Q}}$  is the Mahalanobis norm.

A few notes on the details of this conditional maximization step. Using the Mahalanobis norm as a modifier to the 2-norm penalty in this manner is sometimes referred to as Tikhonov regularization or generalized ridge regression (Hastie et al., 2009). Since these penalties are separable (i.e. they apply to different vari-

ables) the proximal operator is the product of the proximal operators of the two penalties (Beck, 2017), and the proximal operators for both Tikhonov regularization and H-MKL are available in closed form, so we can still use FISTA to find our MAP estimator. However, for Tikhonov regularization, the closed form proximal operator relies on the existence of the Cholesky factor, so  $\mathbf{W}$  must be full rank, which is not the case under the intrinsic GMRF. This is the reason for adding some noise to  $\mathbf{W}$ .

Next we consider the variance parameter  $\sigma^2$ . Conditional on the other random variables, the posterior for  $\sigma^2$  is

$$p(\sigma^2) \propto \left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}+1} \exp\left[-\frac{1}{\sigma^2} \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2 + \tau \beta_1^t \mathbf{W} \beta_1}{2}\right] \times \\ C(\sigma^2, \lambda, c) \exp\left(-\frac{\lambda}{\sigma^2} \sum_{r=2}^{\infty} \sum_{j=1}^{J(r)} c^{r-1} \|\beta_{t_j^r}\|_q\right)$$

with  $\hat{y}_i = \mathbf{X}_i \boldsymbol{\alpha} + \sum_{r=1}^{\infty} \sum_{j=1}^{J(r)} (\mathbf{K}_j^r)^t \beta_j^r$ . Unfortunately, the normalizing constant  $C(\sigma^2, \lambda, c)$  is not tractable due to the issues discussed in section 3.1.3, so finding the mode of this distribution is not possible. We propose to use only the first resolution to estimate this quantity, and ignore the higher resolutions, so our simplified conditional posterior is

$$p(\sigma^2 | \dots) \propto \left(\frac{1}{\sigma^2}\right)^{\frac{n}{2}+1} \exp\left[-\frac{1}{\sigma^2} \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2 + \tau \beta_1^t \mathbf{W} \beta_1}{2}\right]$$

which corresponds to an Inverse Gamma density, so following the ECM, we update

$\sigma^2$  as

$$\sigma_{new}^2 = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2 + \tau \beta_1^t W \beta_1}{n + 2}.$$

To impute  $\tau$  we will use its conditional expectation, following the ECM algorithm. For a particular value of  $\lambda$ , and a MAP estimate of  $\hat{\beta}$ ,  $\tau$  will be updated with

$$\tau_{new} = \frac{a + \frac{J(1)}{2}}{b + \hat{\beta}_1^t \frac{W}{\sigma^2} \hat{\beta}_1}.$$

Due to the complexity of the  $C(\sigma^2, \lambda, c)$  term in the prior for resolutions greater than 1, ECM can not be used with  $\lambda$ . Therefore we propose to calculate a solution path in a similar manner to the LASSO, maximizing the posterior for a number of potential values of  $\lambda$ . We can then use model comparison or model averaging techniques on the resulting maxima. There are two potential strategies for computing these solutions. One is to attempt to solve the solution path, where we start with a very large  $\lambda$ , compute the MAP, then decrease  $\lambda$  and use the previous solution as a warm start. This will, for a set of potential tuning parameters  $\lambda_1 > \lambda_2 > \dots > \lambda_T$ , result in  $T$  separate maxima. In the LASSO, the solutions are piecewise linear, which makes the computation of a solution path extremely efficient. Unfortunately, the solutions to group lasso with an overlap are only piecewise differentiable in  $\lambda$  (Bach, 2008), so the solution path cannot be computed as quickly when one of these non-differentiable points are in between two potential values of  $\lambda$ .

However, solution paths are still often used for problems with penalties that do not have solutions paths that behave nicely (Chen et al., 2012). This is because the  $\beta$  that is the maximum at the previous value of  $\lambda$  still serves as an adequate starting value for the next  $\lambda$ . Another advantage is that model comparison heuristic based stopping rules, such as BIC or posterior model probabilities, can

be tracked. A drawback to this strategy is that computation must be done serially. Another approach that is faster if a large amount of computational resources are available to the user is to compute each solution in parallel. For this approach, the same starting value would be used for each  $\lambda_j$ , and the routine would be run on a number of processors simultaneously. We will discuss the use of model comparison techniques to choose or average these  $T$  solutions in a subsequent chapter.

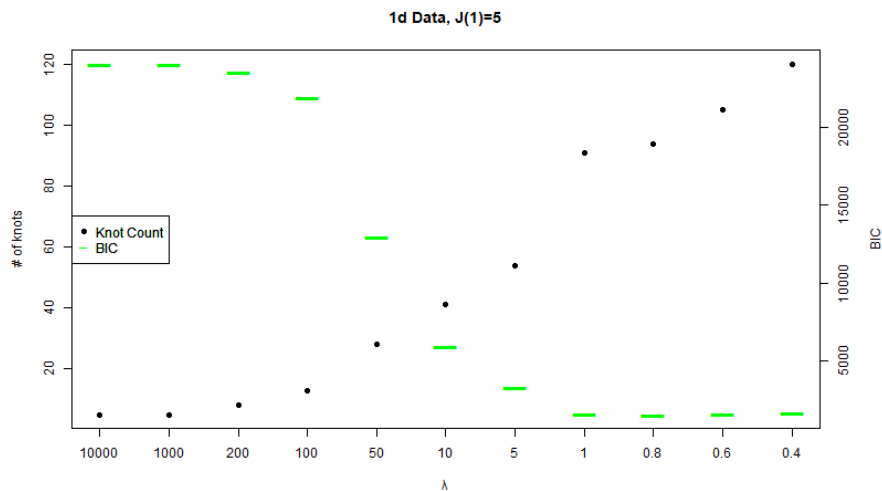
### 3.2.2 Simulation Study

The procedure outlined above will be fit to the first three simulated datasets used in Kirsner and Sansó (2020), which have about 18,000 training and 2000 testing data and are unequally spaced, but have very few large gaps, and to the Mediterranean sea surface temperature dataset, which is unequally spaced and has many large gaps. The parameters  $a$  and  $b$  which control the Markov Random Field scale were set to be 10, and the initial value for the scale parameter  $\tau$  was 1. The initial value for  $\beta$  was set at the MAP estimator for the model with a single resolution of knots. The parameter  $c$ , which controls how much the sparsity increases with the resolution, was set to  $2^D$  where  $D$  is the number of spatial dimensions. For the one dimensional example,  $J(1)$  was varied between 5, 10, and 20, for the two dimensional Gaussian Process examples, between 25, 100, and 400, and for the Mediterranean data between 40 and 160. Out of sample MSPE, runtime, and BIC as an approximation to the marginal model probability were calculated for each of the models evaluated using predictions made at the MAP. For each setting of potential variables and pair of  $\lambda$ ,  $\tau$ , FISTA was used to find the MAP.

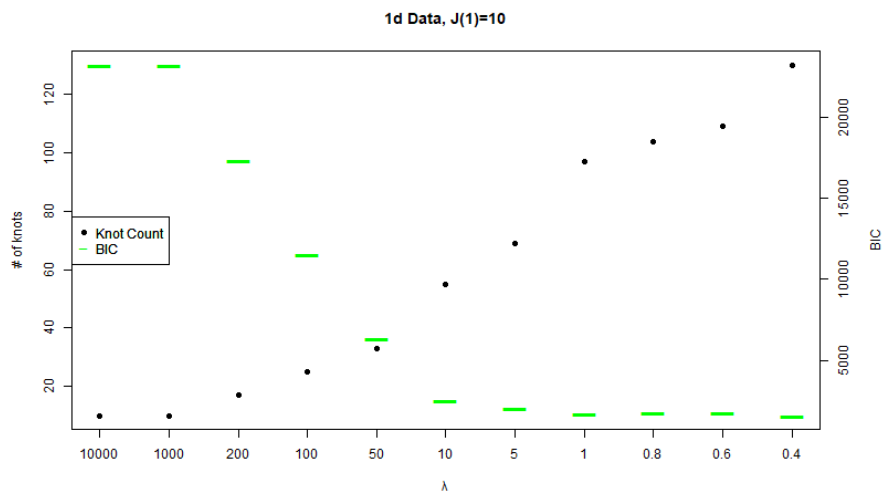
This model was fit for  $\lambda \in \{10000, 1000, 200, 100, 50, 10, 5, 1, .8, .6, .4\}$ . The code for setting up design matrices and keeping track of knots was written in R,

while the optimization was carried out in C++, and called using RCPP. Routines that use FISTA to solve problems of the form of equation (3.13) with  $q = 2$  are implemented in the SPAMS optimization toolbox (Mairal et al., 2010) in C++. However, this software does not support the composition of this regularization and a Mahalanobis norm, which corresponds to the MAP estimate under the prior described in equation (3.21). Therefore, using SPAMS as a starting point, we wrote FISTA optimization code to find the MAP. For each dataset, the solution path approach was used, and computation stopped at either the smallest value of  $\lambda$  or at once the BIC began to increase. All computational routines were run on a Windows desktop with an Intel i7-2600k processor with 4 cores and 16 gigabytes of RAM.

Figures 3.1, 3.2, and 3.3 contain the results for the 1 dimensional dataset. Runtimes and out of sample prediction error were comparable to MSSS, with the paths fit in about 6 minutes and an RMSE of around .1. Figures 3.4 3.5, and 3.6, contain the results for the two dimensional, twice differentiable Gaussian Process, and figures 3.7, 3.8, and 3.9 contain the same for the non-differentiable GP. Predictive performance is similar to MSSS. Computational performance is similar to MSSS for the smallest value of  $J(1)$ , with runtimes on the order of 20 minutes but is substantially worse than MSSS for larger  $J(1)$ , with solution paths taking about 3 hours to compute.



**Figure 3.1:**  $J(1)=5$  1d Optimization Path Summary



**Figure 3.2:**  $J(1)=10$  1d Optimization Path Summary



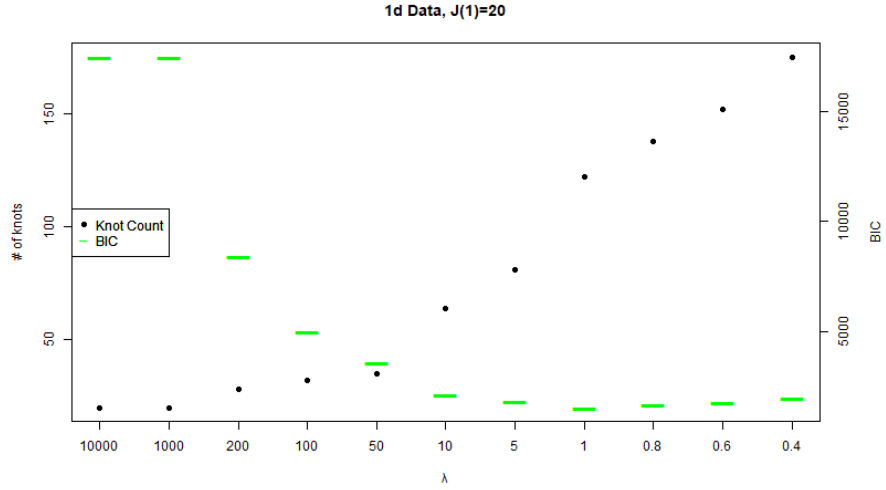


Figure 3.3:  $J(1)=20$  1d Optimization Path Summary

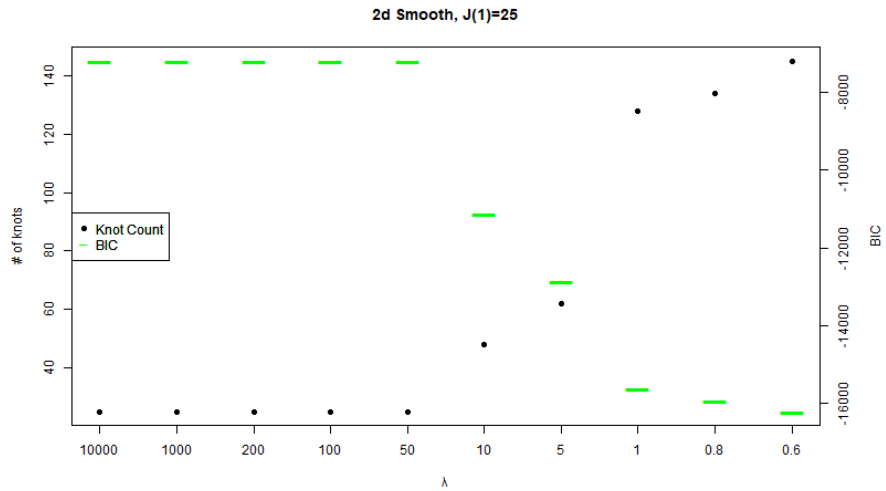


Figure 3.4:  $J(1)=25$  2d Smooth GP Optimization Path Summary

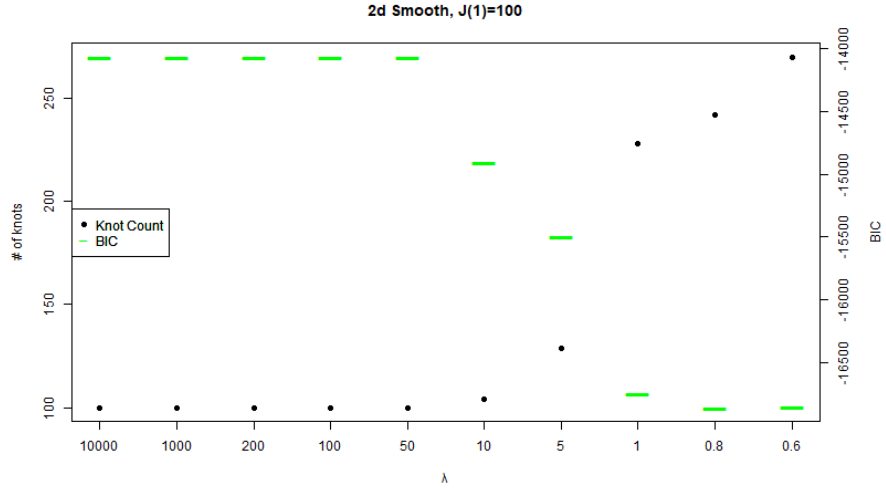


Figure 3.5:  $J(1)=100$  2d Smooth GP Optimization Path Summary

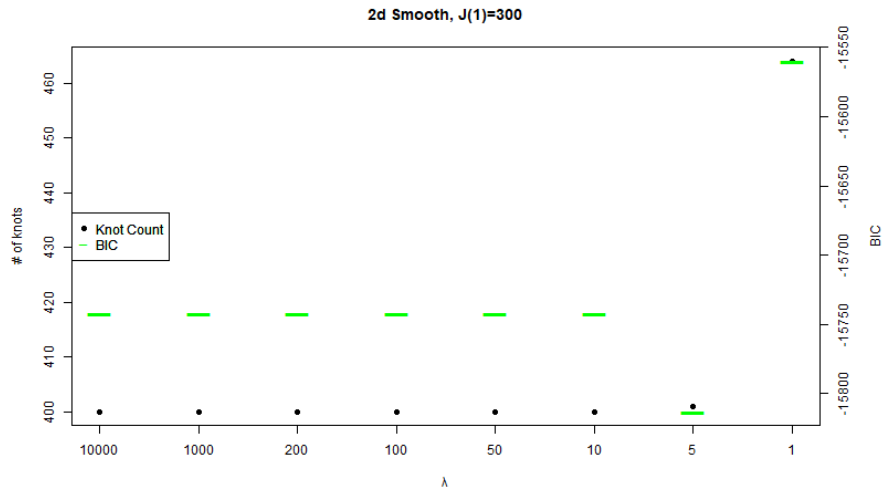


Figure 3.6:  $J(1)=400$  2d Smooth GP Optimization Path Summary

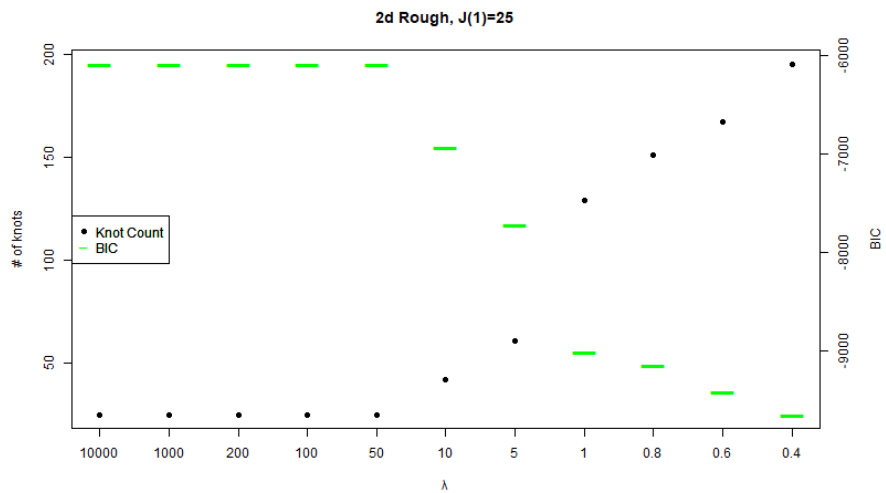


Figure 3.7:  $J(1)=25$  2d Rough GP Optimization Path Summary

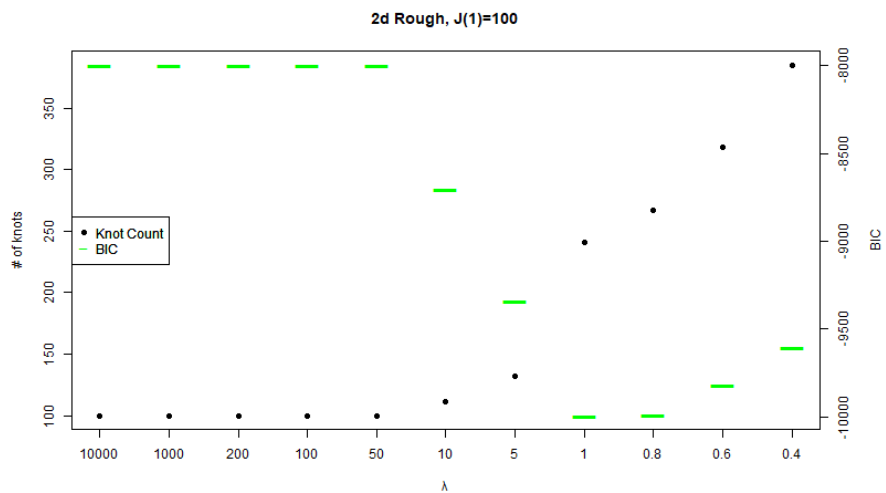
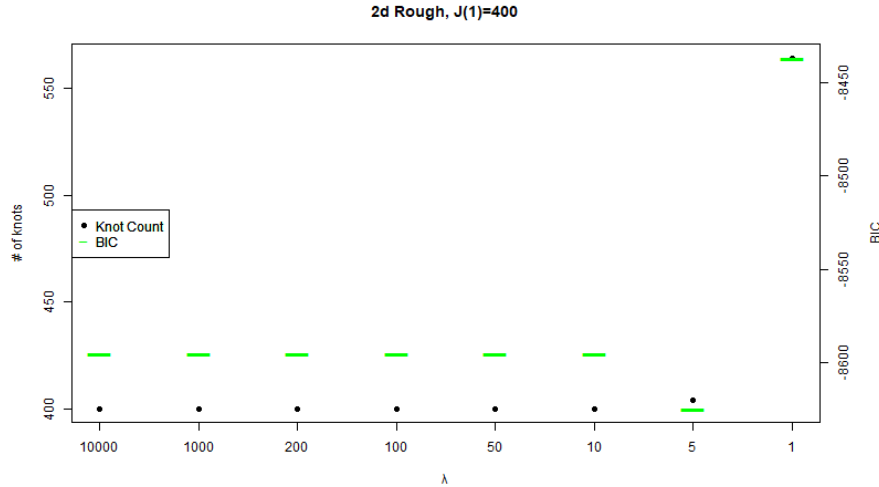
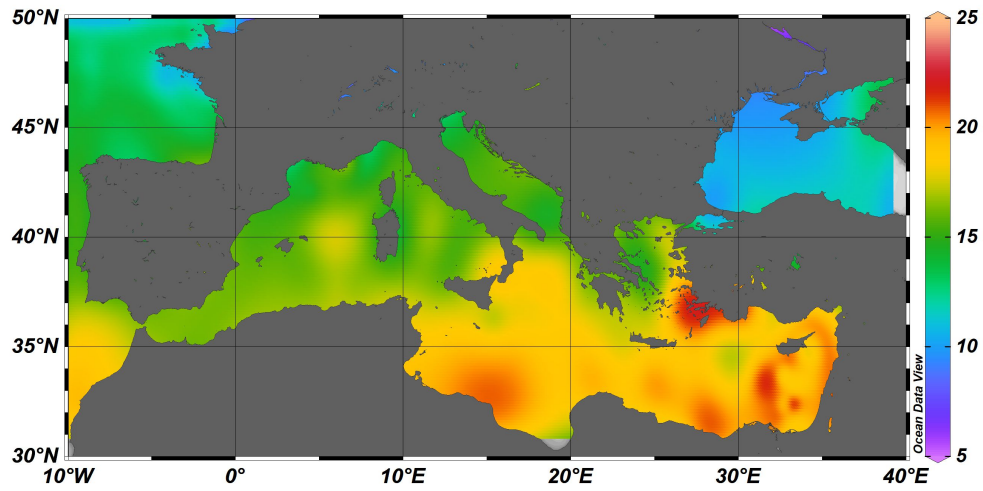


Figure 3.8:  $J(1)=100$  2d Rough GP Optimization Path Summary

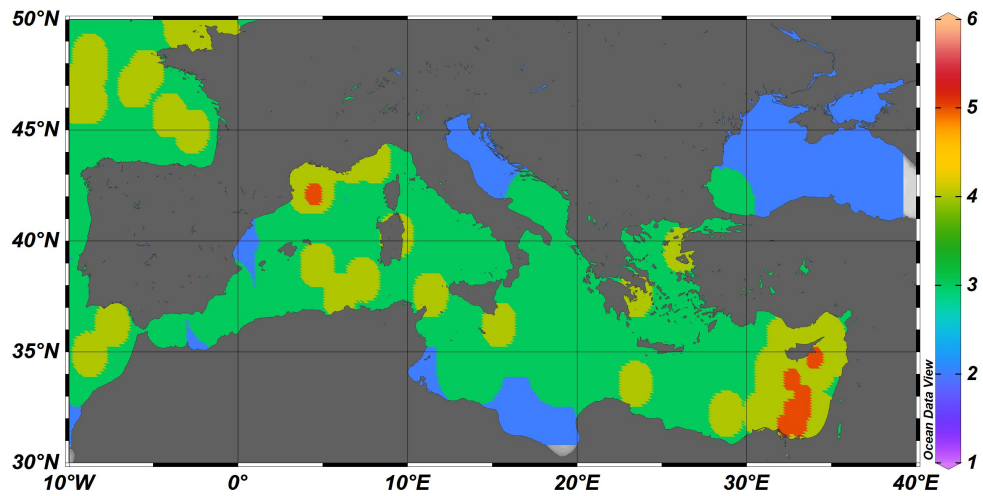


**Figure 3.9:**  $J(1)=400$  2d Rough GP Optimization Path Summary

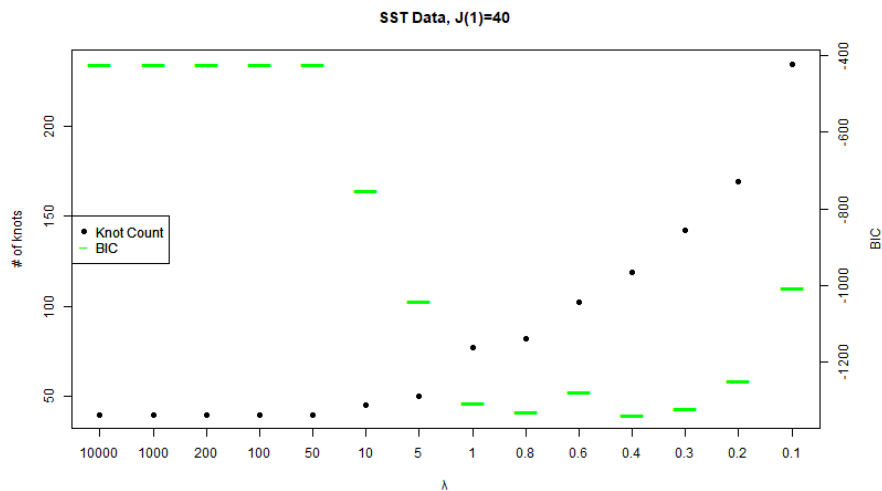
Results for the Mediterranean SST data with  $J(1) = 40$  is in figure 3.12 and results for  $J(1) = 160$  is in figure 3.13. Computational performance was slower than MSSS, with a runtime of about an hour for the smaller  $J(1)$  and about 2 hours for the larger  $J(1)$ . A plot of the resulting predicted surface for the model with  $J(1) = 160$  and  $\lambda = .8$ , which is the model with the best BIC, is displayed in figure 3.10. The resulting resolutions used are displayed in figure 3.11. The resolutions plot is very similar to the results from chapter 2, with more resolutions near Palma and Sardinia due to a local warm spot, near the Brittany peninsula due to a cold spot, and near southeast end of the data due to a large amount of local variation. The resulting predictions from the MAP procedure are smoother than the predictions from MSSS, which could be a desirable feature in some applications. This is because the prior described in section 3.2 provides more shrinkage to the nonzero coefficients than the prior in chapter 2. This shrinkage is applied to the sum of the 2-norm of the tree, so shrinkage applied to a parent coefficient is applied to all of its children in a manner reminiscent of the tree shrinkage prior in Guhaniyogi and Sansó (2017).



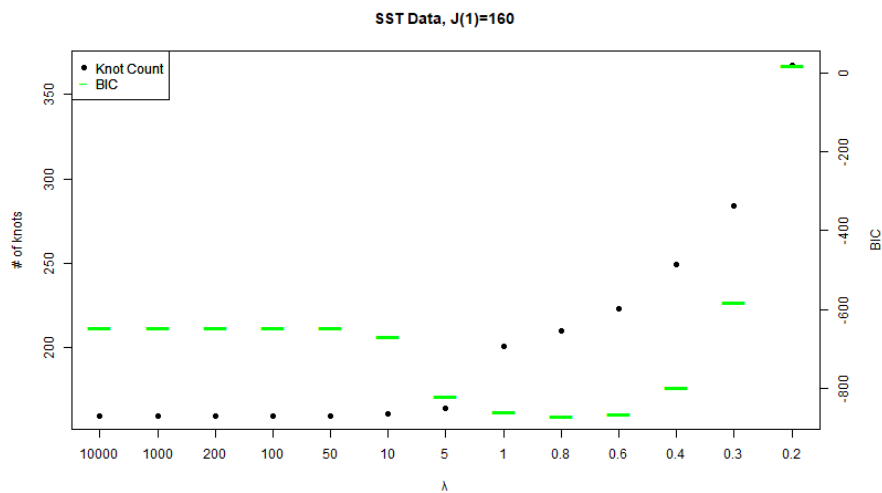
**Figure 3.10:** Predictions over the Mediterranean for the model with the best BIC



**Figure 3.11:** Resolutions active over the Mediterranean for the model with the best BIC



**Figure 3.12:**  $J(1)=40$  SST Optimization Path Summary



**Figure 3.13:**  $J(1)=160$  SST Optimization Path Summary

This method can be run more quickly if run in parallel, but this requires a computer with an very large amount of memory. To highlight this, the model was run for the same settings as 3.11 but in parallel. This reduced the runtime by approximately 30% when compared to the serial computation. However, the memory usage was greater by a factor of about 5.

### **3.3 Conclusion**

We have developed a method for enumerating a sequence of models with spatially varying resolution via the solution path of Bayesian optimization. This method relied on an infinite dimensional version of a composite absolute penalties prior. To fit the model, a proximal gradient descent algorithm was developed. This work allows for point predictions to be made using the MAP estimator under this prior. However, this does not directly result in any ability to perform interval estimation. It also does not allow for selecting the best of a set of models or averaging between a set of them. In chapter 4, we will develop methods for doing model selection and averaging among the models in the solution path.

# Chapter 4

## From Optimization to Bayesian Model Averaging

### 4.1 Introduction

The methods developed in section 3.2 allow for maximum a posteriori estimation for each value in a sequence of  $\lambda$ . This could be thought of as set of potential models of length  $T$ . However, there is no direct way to use a MAP estimator to do uncertainty quantification, such as interval estimation, or model selection, without making other assumptions. In this section, we consider the set of solutions of this optimization trajectory as a finite set of models, and use Bayesian model averaging to improve prediction and perform uncertainty quantification.

Similar approaches have been proposed in the regression context. Fraley and Percival (2015) propose an alternate method of doing Bayesian model comparison in a linear model with a large parameter space of size  $p$ . To perform an exhaustive search,  $2^p$  models must be evaluated. Stochastic search methods have a goal of evaluating only high probability models in that large space, and do not attempt



to exhaustively evaluate all models. The authors propose to use the LASSO solution path as a set of potential models, which reduces the size of the problem substantially. To then perform model averaging and selection, they choose to use MCMCMC (Madigan et al., 1995) with a uniform prior over the model space, and BIC as an approximation to the integrated likelihood for a particular model. Zhou and Wu (2014) also propose a similar approach in the context of their more general penalty structure. Liu (2017) implement this approach in the context of an even more complex model space. They consider the graphical LASSO, which selects from all possible graphical models. They use a refitting scheme, where the LASSO path is used only for selection of the model, then to compute the BIC, they refit the model without regularization, but restricted to the parameters selected by the LASSO.

Both of these approaches perform averaging where the space of potential models is finite, but in the case of our model, initially, the model space is infinite. We will use the MAP procedure described in section 3.2.1 to reduce the problem to a finite set of potential models, and develop a model averaging scheme that performs uncertainty quantification in this setting. On datasets of moderate size, BIC or the hyper-g prior discussed in chapter 2, coupled to the prior developed in section 2.2.1 result in reasonable model selection behavior. For larger datasets, however, this approach always favors models of arbitrarily large size. We will review this issue, showing that Bayesian additive regression trees (Chipman et al., 2010), (Francom et al., 2018), MSSS (chapter 2), and the approach developed in this chapter often select larger models or more complex basis functions when data sizes are large. We will review the approaches to this issue taken by other authors, and propose a solution that alleviates the issue for multi-scale optimization. Specifically, we find a setting for the hyperprior of the parameter that controls the

sparsity level  $\pi$  that is dependent on the data size. This prior leads to reasonable prior behavior with large datasets, and is small enough for small datasets to not change the behavior of the model significantly.

## 4.2 Model Averaging Along Optimization Paths

We will follow the notation of chapter 2, using  $\boldsymbol{\gamma}$  as a vector of 0-1 variables that correspond to a set of nonzero coefficients. The method developed in chapter 3 results in MAP estimates that display spatially varying resolution for each of the  $T$  values of the penalization parameter  $\lambda \in \{\lambda_1, \lambda_2, \dots, \lambda_T\}$  that were chosen to form the solution path. We will let these parameters index the model space. Each  $\lambda_l$  in this set corresponds to a set of coefficient estimates under the prior developed in chapter 3. We will denote the nonzero coefficient estimates for  $\lambda_l$  as  $\hat{\boldsymbol{\beta}}(\lambda_l)$ . We will form the 0-1 variables  $\gamma_j^r(\lambda_l)$  such that

$$\gamma_j^r(\lambda_l) = 1 \iff \beta_j^r \in \hat{\boldsymbol{\beta}}(\lambda_l).$$

The infinite length vector of all of these  $\gamma_j^r(\lambda_l)$  will be referred to as  $\boldsymbol{\gamma}(\lambda_l)$ . To perform Bayesian model averaging, if  $Q$  is our quantity of interest, the resulting estimate is

$$\hat{Q} = \sum_{l=1}^L \frac{\hat{Q}(\lambda_l) \times p(\boldsymbol{\gamma}(\lambda_l))p(\mathbf{y}|\boldsymbol{\gamma}_l)}{\sum_{j=1}^J \times p(\boldsymbol{\gamma}(\lambda_j))p(\mathbf{y}|\boldsymbol{\gamma}_j)}$$

We will first discuss  $p(\boldsymbol{\gamma}(\lambda_l))$ , the prior on the model space. We can then adopt

the same prior on the model space as was used in chapter 2, i.e.

$$Pr(\gamma_{1,j} = 1) = 1 \quad (4.1)$$

$$Pr(\gamma_{r,j} = 1 | \gamma_{r-1}) = \pi \times \gamma_{r-1, \lfloor \frac{j-1}{2^d} \rfloor + 1} \quad (4.2)$$

$$\pi \sim Beta(a_\pi, b_\pi). \quad (4.3)$$

Note that we showed in chapter 3 that the penalized optimization routine results in sparsity that obeys the constraints built into this prior. This prior can be evaluated at  $p(\boldsymbol{\gamma}(\lambda_l))$  for each  $\lambda_l \in \{\lambda_1, \lambda_2, \dots, \lambda_T\}$  to obtain the prior model probabilities with the expression

$$\begin{aligned} p(\boldsymbol{\gamma}(\lambda_l)) &= \frac{a(\lambda_l), b(\lambda_l)}{B(a_\pi, b_\pi)} \text{ with} \\ a(\lambda_l) &= a_\pi + \sum_{r=2}^{\infty} \sum_{j=1}^{J(r)} \gamma_j^r(\lambda_l) \text{ and} \\ b(\lambda_l) &= b_\pi + \sum_{r=2}^{\infty} \left[ 2^d \sum_{j=1}^{J(r-1)} \gamma_j^{r-1}(\lambda_l) - \sum_{j=1}^{J(r)} \gamma_j^r(\lambda_l) \right]. \end{aligned}$$

As for the marginal  $p(\mathbf{y}|\boldsymbol{\gamma})$ , several options exist. One option is to adopt the same hyper-g prior used in chapter 2. To evaluate the marginal distribution under the hyper-g prior, we must obtain the least squares estimator  $\hat{\boldsymbol{\beta}}_\gamma(\lambda_l)$  and calculate the resulting  $SSE_\gamma$ . To avoid having to calculate the regression on a potentially very large design matrix from scratch, we will use gradient descent with the solution to the penalized problem  $\hat{\boldsymbol{\beta}}(\lambda_l)$  as a starting point. The resulting marginal likelihood estimate is

$$p(\mathbf{y}|\boldsymbol{\gamma}(\lambda_l)) = \frac{a-2}{\sum_{i=1}^{J(r)} \gamma_j^r(\lambda_l) + a - 2} {}_2F_1 \left( \frac{n-1}{2}, 1, \frac{\sum_{i=1}^{J(r)} \gamma_j^r(\lambda_l) + a}{2}, \frac{SSE_\gamma(\lambda_l)}{SSE_0} \right).$$

This is extremely similar to the approach taken by Liu (2017), where the penalized optimization procedure was used to limit the size of the model space only, but then the coefficients from the optimization are discarded and the model is refit. Under the hyper-g prior, model averaging and interval estimation can be performed using the expressions derived in section 2.2.5.

This approach for estimation of the marginal likelihood would discard the smoothness properties that we observed empirically on the Mediterranean data in figure 3.10. Another possibility for the marginal likelihood that keeps the smoothness properties of the prior developed in section 3.2.1 is to use the BIC as an approximation to the marginal likelihood, and rather than refitting the model, use the penalized estimates obtained from the MAP estimation. In this case, if  $\hat{y}_i(\lambda_l)$  is the predicted value for  $y_i$  from the model indexed by  $\lambda_l$ , then the marginal likelihood would be calculated as

$$\log(p(\mathbf{y}|\boldsymbol{\gamma}(\lambda_l))) \approx -\frac{BIC(\lambda_l)}{2}$$

and

$$-\frac{BIC}{2} = -\frac{\log(n)}{2} \times \left( \sum_{r=1}^{\infty} \sum_{j=1}^{J(R)} \gamma(\lambda_l)_j^r + q \right) - \frac{1}{2} \log \left( \sum_{i=1}^n (y_i - \hat{y}_i(\lambda_l))^2 \right).$$

This approach does not naturally lead to an estimate for  $\sigma^2$ , which is needed for our interval estimation. We will use the plug in estimator that was used in the optimization procedure, with

$$\hat{\sigma}^2(\lambda_l) = \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2 + \tau \hat{\boldsymbol{\beta}}_1(\lambda_l)^t \mathbf{W} \hat{\boldsymbol{\beta}}_1(\lambda_l)}{n + 2}.$$

The BIC based averaging approach tends to prefer models with fewer knots.

This is because when the model is refit using the hyper-g prior, the resulting estimated coefficients have substantially less shrinkage applied to them than the estimates that are sourced from the optimization procedure.

### 4.3 Mediterranean Data

The same predictions that were displayed in 3.11 and figure 3.10 were used for demonstrating the two model averaging approaches. Under both the hyper-g prior with refitting strategy, and the strategy that uses BIC at the MAP, the log marginal model probabilities were calculated and displayed in table table 4.1. As mentioned in the previous section, the shrinkage at the point estimate causes the BIC based method to favor smaller models. Using BIC, the BMA predictions will very strongly favor the top 2 models, with

$$\hat{\mathbf{y}} = .93\hat{\mathbf{y}}(.6) + .07\hat{\mathbf{y}}(.8).$$

The rest of the models will have virtually no mass.

$\lambda$	# of parameters	BIC Log Model Probability	Hyper-G Log Model Probability
10000	40	676.26	759.85
1000	40	678.88	762.46
200	40	679.23	762.81
100	40	679.29	762.86
50	40	679.30	762.87
10	40	679.30	762.87
5	45	791.75	1098.24
1	65	1158.83	1587.60
0.80	68	1194.63	1593.76
0.60	74	1197.26	1891.91
0.40	98	1171.91	1594.91
0.30	116	1074.06	1925.21
0.20	119	1178.78	2026.67
0.10	171	853.69	2119.71

**Table 4.1:** Results for the Mediterranean data with  $J(1) = 40$ .

## 4.4 Issues With Large Data

In the context of fixed design matrices and linear regression, Bayesian model averaging using hyper-g priors have a number of very strong consistency results (Bayarri et al. (2012), Liang et al. (2008), etc.) for model selection as the size of the data grows. However in the context of models that have a huge number of potential basis functions, or with a moderate number of basis functions that are extremely flexible, large data can result in complex models.

In multi-scale shotgun stochastic search (chapter 2), this behavior is exhibited when data sizes are approximately  $10^6$ . MSSS controls the (potentially infinite) number of basis functions via the prior in section 2.2.1. Usually, MSSS stops after a reasonable number of iterations because the posterior model probabilities stop increasing. However, with large data, the model will run for thousands of iterations, continually adding knots to a degree that is undesirable. This issue can be mitigated by limiting the number of iterations to some fixed constant.

Bayesian additive regression trees (Chipman et al., 2010) (BART) also exhibits similar behavior. BART is an extremely flexible semi-parametric model that has a fixed number of basis functions, which can be thought of as trees. Each tree can have an infinite number of nodes, but the number of nodes is controlled by a prior. However, with large data, this prior is not strong enough to control the size of the tree, resulting in large trees and poor computational performance (Chipman et al., 2010). As the sample size grows, so does the size of the trees. Their proposed solution to this issue is to limit the number of iterations to some fixed constant.

Bayesian multivariate adaptive splines (BMARS) exhibits this behavior as well. BMARS is an extremely flexible approach that relies on reversible jump MCMC to select the number of knots for a multivariate spline. The number of knots is controlled by a prior distribution. In the settings examined in Francom

et al. (2018), the estimated number of basis functions had a tendency to grow too large. The authors solved this issue by setting a  $Poisson(\mu)$  prior on the number of basis functions, and a  $Gamma(1, 10^{300})$  hyperprior on  $\mu$ , resulting in a prior mean number of basis functions that is essentially zero.

These issues arise because the marginal likelihood grows exponentially in  $n$ . This means that if a model is slightly misspecified or if there is enough data for it to display very complex features, these flexible models will be encouraged to account for it in a manner that ultimately produces overfitting. To combat this we will adopt a similar approach to BMARS, and use a very strong prior on the parameter that controls the sparsity in the basis functions. Specifically, we will set

$$\pi \sim Beta(a_\pi, b_\pi) \tag{4.4}$$

$$a_\pi = 1 \tag{4.5}$$

$$b_\pi = 10^{3 \times n / 10^4}. \tag{4.6}$$

This reflects an unrealistic prior expected number of basis functions, and could cause numerical overflow if  $n$  is too large. When  $n$  is very small, the prior expected number of basis functions under this prior is essentially infinite, but with very small prior sample size. When  $n$  is large, the prior expectation is virtually infinite with a huge amount of weight. Empirically, however, this prior setting leads to reasonable model selection behavior. With small sample sizes, the  $a_\pi$  and  $b_\pi$  parameters do not matter much. For example, applying this prior to the Mediterranean model discussed in table 4.1 only changes the results slightly, resulting in

$$\hat{\mathbf{y}} = .84\hat{\mathbf{y}}(.6) + .14\hat{\mathbf{y}}(.8).$$

With extremely large  $n$ , the prior is large enough to stop the Bayesian model averaging from picking the largest model every time. To demonstrate this, we use GHRSSST Level 3C North Atlantic Regional data, which consists of very high resolution sea surface temperature datasets. We use data from December 2017 collected over the North Atlantic at a .02 degree resolution. This data was restricted to be over the Mediterranean. It was collected over the entire month, so each point on the surface has more than one observation. This was reduced to having one observation per point on the grid by taking the arithmetic mean. This results in just over a million data. However, by using the prior from equation (4.5), we obtain the results in table 4.2. Through this prior that is highly dependent on the data size, we are able to obtain model averaging and selection behavior that results in reasonably sized models.

$\lambda$	# of parameters	BIC	Log Model Probability
10000	91	14395.32	
1000	107	39737.20	
200	180	118611.85	
100	251	149240.45	
10	597	119653.19	

**Table 4.2:** Results for the Mediterranean data with  $J(1) = 40$ .

## 4.5 Summary

We have developed a Bayesian model averaging approach that allows for uncertainty quantification and interval estimation for spatially varying resolution that is obtained via optimization rather than stochastic search. We discuss the different options available for calculating the marginal distribution, and recommend BIC evaluated at the MAP of the CAP family prior. We also discussed the issues that arise with extremely large datasets and flexible, semi-parametric



models outside of the realm of spatially varying resolution. We propose a solution that resolves this issue in the context of this model, resulting in finite models even in the presence of large datasets.

# Chapter 5

## MSSS: An R package for Fitting Surfaces With Spatially Varying Resolution

### 5.1 Introduction

The R package `MSSS` was developed to provide an easy to use implementation of spatial models with multi-resolution kernel convolutions that result in spatially varying resolution as considered in chapter 2. This package is able to fit models with this property, generate out of sample predictions from fitted models, and generate datasets of the number of resolutions used, which can be useful in feature identification (see chapter 6). The package can be downloaded at <https://github.com/daktx2/MSSS> or installed via the command `install_github("https://github.com/daktx2/MSSS")` command in R. If the optimization methods of chapter 3 are desired, then the package `multires`, located at <https://github.com/daktx2/multires> will also need to be installed.

There are a large number of R packages that implement spatial methods. A comprehensive list can be found at the CRAN task view for spatial data analysis at <https://cran.r-project.org/web/views/Spatial.html>. Many of the methods discussed in chapter 1 and in the forthcoming case study (chapter 6) have R packages that implement their models. These can be found in table 5.1.

Package	Citation
spNNGP	Finley et al. (2017)
LatticeKrig	Nychka et al. (2015)
FRK	Cressie and Johannesson (2008)
spBayes	Finley et al. (2009)
INLA	Lindgren et al. (2011)
gapfill	Gerber et al. (2018)
laGP	Gramacy et al. (2016)
GPvecchia	Katzfuss (2017)
tgp	Gramacy and Lee (2008)

**Table 5.1:** Partial list of R package implementations of discussed methods.

We introduce the package as follows. In section 5.2 we describe a spatial model that results in spatially varying resolution. We then discuss the two potential strategies for selection and averaging over potential sets of knots with spatially varying resolution that are provided by this package. We also discuss the different options available in the package for parts of the model, including prior distributions and kernel functions, and some computational details of how these options are implemented. We then provide an example of how to install the package and fit the model to a simulated Gaussian process dataset discussed in chapter 2, and show how the predicted surface changes when different options are used. We also show how using Wendland kernels affect the fit to the Mediterranean observational SST data.

## 5.2 Spatially Varying Resolution

We begin with a standard spatial regression model,

$$y(s)_i = \mathbf{x}(s)_i^T \boldsymbol{\alpha} + w(s) + \epsilon(s)_i, \quad \epsilon(s)_i \stackrel{i.i.d.}{\sim} N(0, \sigma^2),$$

where  $\mathbf{x}(s)_i$  is a  $q \times 1$  vector of individual level predictors,  $\boldsymbol{\alpha}$  is the  $q \times 1$  vector of fixed effect regression coefficients associated with the predictors,  $w(s)$  is the spatial effect,  $i$  is the index for replicates at a particular point  $s$ , and  $\epsilon(s)_i$  Gaussian random noise. We let the spatial process be represented by the multi-resolution process convolution

$$w(s) = \sum_{r=1}^{\infty} \sum_{j=1}^{J(r)} K(s, s_j^r | \phi_r, \nu) \beta_j^r.$$

where  $K(\cdot)$  is a kernel function.

To achieve spatially varying resolution, we require structured sparsity in  $\beta_j^r$ . If we let  $\gamma_j^i$  be a 0-1 random variable with  $Pr(\beta_j^r = 0 | \gamma_j^r = 0) = 1$ , a prior on the model space that enforces spatially varying resolution is

$$\begin{aligned} Pr(\gamma_j^1 = 1) &= 1 \\ Pr(\gamma_j^r = 1 | \gamma_{\lfloor \frac{j-1}{2^d} \rfloor + 1}^{r-1} = 1) &= \pi \times \gamma_{r-1, \lfloor \frac{j-1}{2^d} \rfloor + 1} \\ Pr(\gamma_j^r = 1 | \gamma_{\lfloor \frac{j-1}{2^d} \rfloor + 1}^{r-1} = 0) &= 0 \\ p(\pi) &= Beta(a_\pi, b_\pi). \end{aligned}$$

This package provides two strategies for fitting models with this feature. The first strategy is to attempt to enumerate the top  $Q$  models via multi-resolution shotgun stochastic search (Kirsner and Sansó, 2020). The second strategy is to

restrict the set of potential models to a much smaller set via the solution path of an optimization procedure. This is discussed in detail in chapter 3.

### 5.2.1 Kernel Functions

The model described in section 2.2.3 is subject to the choice of kernel function. Two choices of kernel function are implemented in the software. Both of these choices offer compact support, which substantially increases the computational efficiency of these methods. Compact support is not required in general, though it improves the computational efficiency of the stochastic search. One choice of kernel is the Bezier kernel function (Brenning, 2001), which is defined as

$$K(s, s_j^r, \phi_r, \nu) = \begin{cases} \left(1 - \left(\frac{\|s - s_j^r\|}{\phi_r}\right)^2\right)^\nu & \|s - s_j^r\| < \phi_r \\ 0 & \text{otherwise.} \end{cases}$$

This extremely flexible kernel offers a range parameter  $\phi_r$  and a smoothness parameter  $\nu$  that controls the differentiability. Also available is the Wendland kernel, which is used in Nychka et al. (2015) and Guhaniyogi and Sansó (2017) among others. If  $l = \lfloor d/2 \rfloor + 2$ , then this kernel is defined as

$$K(s, s_j^r, \phi_r) = \begin{cases} \left(1 - \frac{\|s - s_j^r\|}{\phi_r}\right)^{l+1} \left[1 + (l+1)\frac{\|s - s_j^r\|}{\phi_r}\right] & \|s - s_j^r\| < \phi_r \\ 0 & \text{otherwise.} \end{cases}$$

This kernel has only a range parameter  $\phi_r$ , and is four times differentiable. It results in continuously differentiable realizations of a spatial surface (Guhaniyogi and Sansó, 2017). For both potential choices of kernels, we set  $\phi_r = \eta \|s_j^r - s_{j-1}^r\|$ , which results in narrower kernels at higher resolutions. It is required that  $\eta > 1$  for this to be a coherent spatial model, as smaller values will result in gaps. If Bezier

kernels are used, the software default smoothness parameter is  $\nu = 1$ . However, in chapter 2 we showed that the predictive properties of these models are extremely robust to varying kernel choices.

### 5.2.2 Priors for $\alpha$ , the nonzero $\beta_j^r$ , and $\sigma^2$

The above model is completed with a prior on the nonzero  $\beta_j^r$ , the fixed effects  $\alpha$ , and the variance  $\sigma^2$ . Specifically, if we let  $\beta_\gamma$  be the vector of  $\beta_j^r$  for  $r > 1$  where  $\gamma_j^r = 1$ , then

$$p(\alpha, \beta_1, \beta_\gamma, \sigma^2 | \gamma) = p(\beta_\gamma | \alpha, \beta_1, \sigma^2) p(\beta_1 | \sigma^2, \alpha) p(\alpha, \sigma^2 | \gamma)$$

The software restricts the choice of prior on the fixed effects and variance to be the reference prior,

$$p(\alpha, \sigma^2 | \gamma) \propto \frac{1}{\sigma^2}.$$

For  $\beta_j^r$  at any resolution, if the design matrix has no nonzero entries, then the software will force the coefficient to be zero. This is necessary for numerical stability. Two choices are offered for the prior on the first resolution coefficients. A flat reference prior is the default, i.e.

$$p(\beta_1 | \alpha, \sigma^2) \propto 1.$$

Alternative, if more shrinkage is desired at the first resolution, a multivariate normal distribution, i.e.

$$p(\boldsymbol{\beta}_1 | \boldsymbol{\alpha}, \sigma^2) = N_{J(1)}(\mathbf{0}, \sigma^2 \mathbf{V}) \quad (5.1)$$

could be used. A suggested option is

$$p(\boldsymbol{\beta}_1 | \boldsymbol{\alpha}, \sigma^2) = N_{J(1)}\left(\mathbf{0}, \sigma^2 \tau^2 (\mathbf{W} + \delta \mathbf{I}_{J(1)})^{-1}\right)$$

where  $\mathbf{W}$  is the precision matrix for an intrinsic Markov Random Field (GMRF). The intrinsic GMRF is a common choice as a prior for process convolutions (for example, Lemos and Sansó (2009)). This option is useful to enforce spatial coherence when the number of resolution 1 knots is large. The matrix  $\mathbf{W}$  has 0's in all elements except for the diagonals, which have the number of neighbors of those knots, and the off diagonal entries corresponding to neighbors, which are set to -1.  $\delta$  corresponds to a small amount of diagonal noise to ensure that this matrix is invertible, which is necessary for the computation of posterior model probabilities.

Two options are offered for the prior on the nonzero knot coefficients at resolutions greater than 1. One option is the hyper-g prior Liang et al. (2008). If we let

$$\mathbf{Z}_0 = \begin{bmatrix} \mathbf{X}_0 & \mathbf{K}_1 \end{bmatrix},$$

then the hyper-g prior is

$$p(\boldsymbol{\beta}_\gamma | \boldsymbol{\alpha}, \boldsymbol{\beta}_1, \sigma^2, g) = N(0, g\sigma^2(\mathbf{V}_\gamma^T \mathbf{V}_\gamma)^{-1}) \quad (5.2)$$

$$p(g) = \left(\frac{a-2}{2}\right) (1+g)^{-a/2} \quad (5.3)$$

$$\text{with } \mathbf{V}_\gamma = \left(\mathbf{I}_{n+p+J(1)} - \tilde{\mathbf{Z}}_0(\tilde{\mathbf{Z}}_0^t \tilde{\mathbf{Z}}_0)^{-1} \tilde{\mathbf{Z}}_0\right) \tilde{\mathbf{K}}_i. \quad (5.4)$$

This results in a closed form posterior model probability of

$$p(\boldsymbol{\gamma} | \mathbf{y}) = \frac{a-2}{\sum_{i=1}^{J(r)} \gamma_{r,j} + a - 2} {}_2F_1 \left( \frac{n-1}{2}, 1, \frac{\sum_{i=1}^{J(r)} \gamma_{r,j} + a}{2}, \frac{SSE_\gamma}{SSE_0} \right) \times \quad (5.5)$$

$$\frac{B\left(a_\pi + \sum_{r=2}^{\infty} \sum_{j=1}^{J(r)} \gamma_{r,j}, b_\pi + \sum_{r=2}^{\infty} \left[2^d \sum_{j=1}^{J(r-1)} \gamma_{r-1,j} - \sum_{j=1}^{J(r)} \gamma_{r,j}\right]\right)}{B(a_\pi, b_\pi)} \quad (5.6)$$

Specifying a hyper-g prior using the matrix  $\mathbf{Z}_0$  instead of  $\mathbf{K}\boldsymbol{\gamma}$  makes the prior orthogonal to the fixed effects and resolution 1 knots, which allows for the easily computed Bayes factor above. In Kirsner and Sansó (2020), this orthogonalization was not carried out, which makes the base model in the comparison a model with an intercept only. The space of models is infinite, so to enumerate high probability models, a stochastic one knot at a time search algorithm called Shotgun Stochastic Search (Hans et al., 2007) was modified to this setting. Further details about Shotgun Stochastic Search are given in section 2.2.3 and section 5.4.1. The other choice for this prior is the tree structured prior of chapter 3. If we let  $\boldsymbol{\beta}_{t_j^r}$  be the vector of coefficients corresponding to the subtree with root  $\beta_j^r$ , then

$$p(\boldsymbol{\beta}_{-1} | \sigma^2, \lambda, d) = C(\sigma^2, \lambda, d) \exp \left( -\frac{\lambda}{\sigma^2} \sum_{r=2}^{\infty} \sum_{j=1}^{J(r)} c^{r-1} \|\boldsymbol{\beta}_{t_j^r}\|_2 \right) \quad (5.7)$$



where  $C(\sigma^2, \lambda, d)$  is an intractable normalizing constant. This software permits computation of the maximum a posteriori estimate under this prior for fixed  $\lambda$  and  $c$ .

For computation of the posterior model probabilities shown in 5.2.2 with under the prior on resolution 1 shown in section 5.2.2 and the hyper-g prior shown in equation (5.4), the data augmentation form of conjugate priors. (Gelman and Hill, 2006) must be used. Let

$$y_* = \begin{bmatrix} \mathbf{y} \\ 0_{p+J(1)} \end{bmatrix},$$

and let the augmented design matrix be

$$\mathbf{Z}_0 = \begin{bmatrix} \mathbf{X}_0 & \mathbf{K}_1 \\ \mathbf{0} & \mathbf{I}_{J(1)} \end{bmatrix}.$$

Then the new regression covariance matrix will be

$$\mathbf{U} = \begin{bmatrix} \mathbf{I}_n & 0 \\ 0 & \tau^2 \mathbf{Q}^{-1} \end{bmatrix} = \mathbf{W}^t \mathbf{W} = \begin{bmatrix} \mathbf{I}_n & 0 \\ 0 & \tau \mathbf{\Lambda}^{-1} \end{bmatrix}^t \begin{bmatrix} \mathbf{I}_n & 0 \\ 0 & \tau \mathbf{\Lambda}^{-1} \end{bmatrix}.$$

Note that the Cholesky decomposition can be calculated for each block separately (because we have a block diagonal matrix), and the Cholesky of the identity is the identity, so we only need to calculate  $\mathbf{\Lambda}^{-1}$ , the Cholesky factor for  $\mathbf{V}$ , the variance matrix of the prior. In fact, for estimation, we only need  $\mathbf{\Lambda}$ , which is the Cholesky factor of the precision matrix of this prior. We will create a transformed response and predictor matrix to remove the correlated errors and end up with regression

under independence. Applying this, we obtain

$$\tilde{y} = \mathbf{W}^{-1}y_* = y_*, \quad \tilde{\mathbf{Z}}_0 = \mathbf{W}^{-1}\mathbf{Z}_0$$

Note that since the last entries of  $y_*$  are zero,  $\tilde{y} = y_*$ . Also, by properties of matrix multiplication,

$$\tilde{\mathbf{Z}}_0 = \begin{bmatrix} \mathbf{X}_0 & \mathbf{K}_1 \\ \mathbf{0} & \tau\Lambda \end{bmatrix}.$$

All kernel matrices at a resolution greater than 1 will need to be padded with zeros, namely

$$\tilde{\mathbf{K}}_\gamma = \begin{bmatrix} \mathbf{K}_\gamma \\ \mathbf{0} \end{bmatrix}.$$

With the augmented design matrix and response vector, we can now use the expression in section 5.2.2 to compute posterior model probabilities and Bayes factors. Adding this proper prior encourages some smoothness and stability in the resolution one coefficients.

## 5.3 Fitting the Models

This package offers two options for fitting models with spatially varying resolution. The first choice is multi-scale shotgun stochastic search (MSSS), which is documented in detail in chapter 2, and the second is multi-scale spatial optimization, which is developed in chapter 3.

### 5.3.1 Multi-scale Shotgun Stochastic Search

Multi-resolution Shotgun stochastic search attempts to enumerate the highest probability part of the model space via a stochastic search. It relies on the ex-

istence of closed form marginal distributions, so requires that the hyper-g prior (equation (5.4)) is selected for the prior on nonzero coefficients. Its speed relies on parallel computing, and is most efficiently run with 10-20 cores.

To perform MSSS, we first must define the set of neighbors  $N$  of a model  $m_0$ , which is of size  $p$ . MSSS exhaustively evaluates the posterior model probability of each element in  $N$  in parallel, and then randomly selects one of them to be the next  $m_0$  with probability proportional to the posterior model probability. The top  $Q$  of these exhaustive evaluations are saved for model averaging after the procedure finishes. In this setting, we split  $N$  into two groups,  $N = N_- \cup N_+$ .  $N_-$  is defined as all models of size  $p - 1$  that remove only one childless knot from  $m_0$ . Moving to a model in this set is termed a *deletion move*.  $N_+$  is defined as all models of size  $p + 1$  that contain all predictors from  $m_0$ , and then add one knot to a parent with an open spot for a child. Moving to a model in this set is termed an *addition move*.

The initial  $m_0$  is set to be all resolution 1 knots, so iteration 1 does not consider any deletion moves. The software runs this procedure for a maximum number of iterations  $I$ , but will exit early if the set of top models  $Q$  is not updated for 3 iterations, which means that the procedure is unlikely to find any additional models with high posterior model probability.

To run the stochastic search procedure, the function `msss_fit` is provided. table 5.2 lists the arguments for this function. Only the first four are mandatory. In chapter 2 we show that this method is quite robust to the choice of kernel parameters  $\eta$ ,  $\nu$  and prior parameters  $a_\pi$ , and  $b_\pi$ . The defaults provide for a quite rough predicted surface, but either increasing  $\nu$  or switching to Wendland kernels will result in a much smoother predicted surface. Reasonable values of  $a_\pi$  and  $b_\pi$  have little effect on the resulting model size if more than 1,000 data

points are used. Details of each argument to this function is displayed in table 5.2. Predictions, intervals, and resolutions used via Bayesian model averaging at out of

Argument	Description
locations	locations of the spatial observations as a matrix
yy	response vector
knots_r1	resolution 1 knots as a matrix
spatial_dimension	dimension of the spatial field, can be 1 or 2
maxiters	maximum number of iterations for the stochastic search, 100 is default
cores	number of cores, 1 is default but 10-20 is recommended
design_mat	default is NULL, but at least an intercept is recommended
a_pi	a for beta prior on $\pi$ if pi_method=1, 1 is default
b_pi	b for beta prior on $\pi$ if pi_method=1, 5 is default
a_g	parameter for hyper-g prior, 3 is default, $2 \leq a_g \leq 4$ recommended
kernel_width	$\eta$ , Bezier kernel width parameter, should be 1.5 or greater, 1.5 is default
nu	$\nu$ , Bezier kernel smoothness parameter, 1 is default
pi_method	Beta prior for $\pi$ , 1 (default), 2 is for fixed $\pi$
R1_prior	$J(1) \times J(1)$ covariance matrix with for R1 knots if desired
Kernel_type	kind of kernel to use, the default is the flexible 'bezier' kernel but 'wendland' is also an option

**Table 5.2:** Parameters for `msss_fit`.

sample locations from MSSS are obtained from the function `msss_pred`. Details of each argument to this function are displayed in table 5.3. Section 5.4 will go into further detail about the effects that different parameters have on the predicted surface, and the usage of the above function.

### 5.3.2 Multi-scale Spatial Optimization

The function `mr_optim_fit` uses the optimization method described in chapter 3 to evaluate a set of models with spatially varying resolution corresponding to a set of penalization parameters. Details of each argument to this function is displayed in table 5.4.

Argument	Description
locations	locations where predictions/intervals are desired
results	results from <code>msss_fit</code>
design_mat	design matrix for fixed effects
level	default is .95, confidence level for interval, not needed if type='pred' or type='resplot'
model_used	number of models to use for Bayesian model averaging
type	default is 'pred' for prediction interval, 'mean' for mean interval, 'resplot' for resolutions plot, 'noint' for prediction only

**Table 5.3:** Parameters for `msss_pred`.

Predictions and interval estimation from this method are obtained from the function `mr_optim_pred`. This function considers the maxima found to be the space of potential models for Bayesian model comparison, as described in chapter 4. The prior on the model space is taken to be section 5.2. For knots at resolutions greater than 1, two options are available: **(1)** the hyper-g prior (equation (5.4)) in which case the posterior model probabilities are available exactly via the expressions in section 5.2.2; **(2)** The infinite dimensional CAP prior (section 5.2.2), in which case the marginal model probabilities must be approximated using BIC in the manner of chapter 4. Details of each argument to this function are displayed in table 5.5.

## 5.4 Synthetic Data Example

To demonstrate the effects of the different choices above, we will generate 10000 observations from a Gaussian process observed on a grid on  $[0, 10] \times [0, 10]$  with Matern covariance, a scale parameter of 1, and a smoothness parameter of 2 using the `RandomFields` package (Schlather et al., 2015). We add random normal noise with variance .1 to this mean function, and then use the above `msss_fit` to fit MSSS under different settings. We discuss the reasons the model with different settings results in predicted surfaces with different features.

Argument	Description
yy	response vector
locations	locations of the spatial observations as a matrix
knots_r1	resolution 1 knots as a matrix
spatial_dimension	dimension of the spatial field, can be 1 or 2
kernel_width	Kernel width parameter for Bezier kernel
smoothness	smoothness parameter for Bezier kernel
shrinkage	c value for the sparsity penalty, default is 2
maxiter	maximum number of iterations for the optimization, 100 is default
lambdatree_seq	descending sequence of $\lambda$ for the sparsity penalty default is c(1000,100,10,5,2,1,.5)
tau_init	initial value for the MRF precision, default is 1
tau_a	gamma parameter a for $\tau$ , default is 1
tau_b	gamma parameter b for $\tau$ , default is 1
em_tol	default is .001, tolerance for convergence of $\tau$
m_tol	default is .00001, tolerance for convergence of $\beta$
design_mat	optional design matrix, an intercept is recommended

**Table 5.4:** Parameters for `mr_optim_fit`.

We generate the data with the following code.

```

library(RandomFields)
set.seed(12345)
model <- RMmatern(nu=2)
locations = expand.grid(seq(0,10,length=100),seq(0,10,length
  =100))
simu <- RFsimulate(model, x=locations[,1], y=locations[,2])
obs = simu$variable1+rnorm(10000,0,sqrt(.1))

```

We then create a set of 100 knots that rang on a grid from -2 to 12 (to control for edge effects). and fit a multi-resolution model via multi-scale shotgun stochastic search using the default settings with the code below, and make predictions at each point on the grid. Note that the number of cores is set to 20 below. On computers with fewer than 20, this value would cause oversubscribing, where more processes are called than there are logical cores, which can cause poor performance. For

Argument	Description
locations	locations where predictions/intervals are desired
results	results from <code>msss_fit</code>
design_mat	design matrix for fixed effects
marg_type	could be 'bic_CAP' for BIC + composite absolute penalty
int_type	default is "pred", could be 'mean' for mean interval,
level	default is .95, confidence level for interval, not needed if int_type='pred' or type='resplot' or could be "hyperg_refit" for refitting the model using hyper-g priors
type	default is 'pred' for prediction interval 'resplot' for resolutions plot, 'noint' for prediction only
a_beta	a parameter for beta prior on pi, default is 1
b_beta	b parameter for beta prior on pi, default is 1 but $10^{3n/1000}$ is recommended for large $n$
a_g	a parameter for hyper-g prior, default is 3

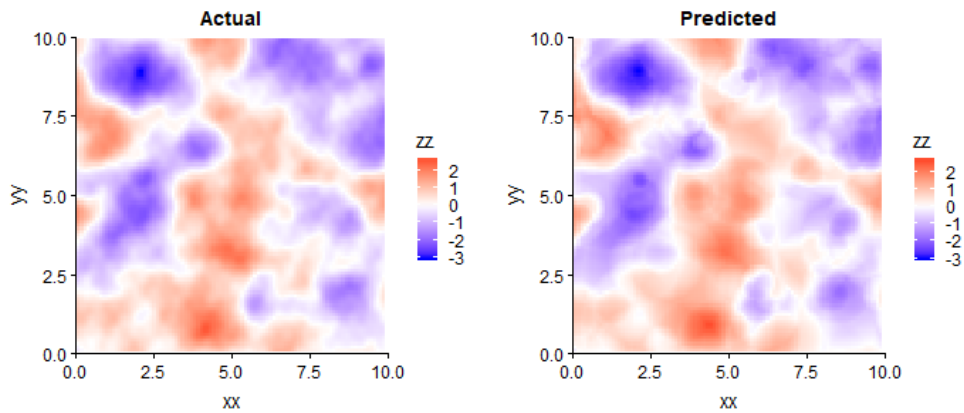
**Table 5.5:** Parameters for `mr.optim.pred`.

stability, 1 is the default value, but this method will run faster if this parameter is set to either 20, or the number of cores on the computer, whichever is lower.

```
knots_r1=r1_create(locations,2,10,2);numcores=20;
default_fit=msss_fit(as.matrix(locations),obs,knots_r1,2,
  maxiters=maxiters,cores=numcores)
preds=msss_pred(locations,default_fit,type="noint")
```

To demonstrate how to create a simple predicted plot for the default settings alongside the observed values, we show figure 5.1. Code for generating this plot is displayed below.

```
df_default=data.frame(xx=locations[,1],yy=locations[,2],zz=c(
  preds$preds))
df_actual=data.frame(xx=locations[,1],yy=locations[,2],zz=simu$
  variable1)
plot_default=ggplot(df_default, aes(x = xx, y = yy, z = zz,
  fill = zz)) + geom_tile()+scale_fill_gradient2(low = "blue",
  mid = "white", high = "red", midpoint = 0)+
```



**Figure 5.1:** Predicted vs Actual.

```

scale_x_continuous(limits = c(0, 10),expand=c(0,0))+scale_y_
  continuous(limits = c(0, 10),expand=c(0,0))+ggtitle("
    Predicted")
plot_actual=ggplot(df_real, aes(x = xx, y = yy, z = zz, fill =
  zz)) + geom_tile()+scale_fill_gradient2(low = "blue", mid =
  "white", high = "red", midpoint = 0)+
scale_x_continuous(limits = c(0, 10),expand=c(0,0))+scale_y_
  continuous(limits = c(0, 10),expand=c(0,0))+ggtitle(" Actual"
  )
library(cowplot)
plot_grid(plot_actual, plot_default)

```

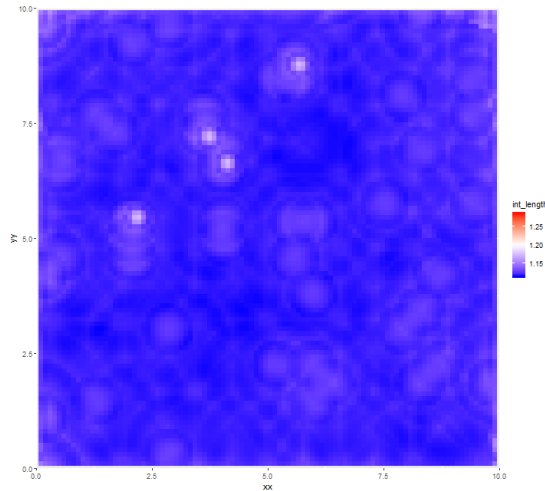
Uncertainty quantification can also be shown, through the length of a prediction interval with a fixed level at each point in the space. This is displayed for a 90% interval in figure 5.2, and code for generating this plot is below.

```

default_preds=msss_pred(locations, default_fit, design_mat =
  design_mat, model_used = 1, type="pred", level = .9)
df_default=data.frame(xx=locations [,1], yy=locations [,2], int_
  length=c(default_preds$upper–default_preds$lower))

```





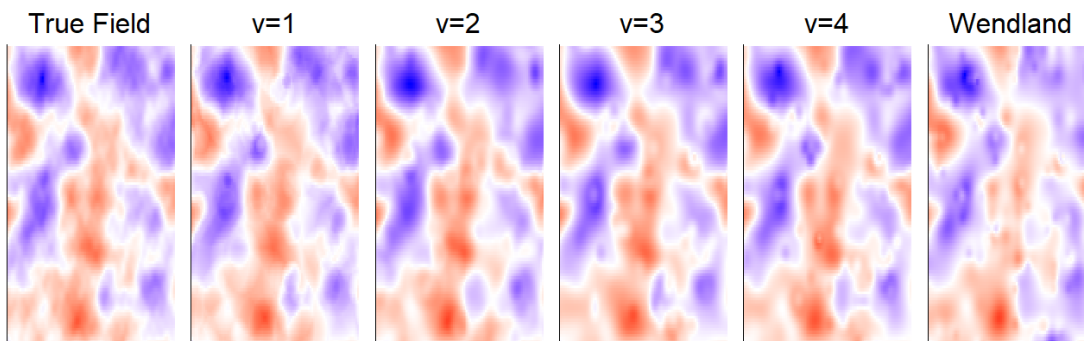
**Figure 5.2:** Length of a 90% prediction interval.

```

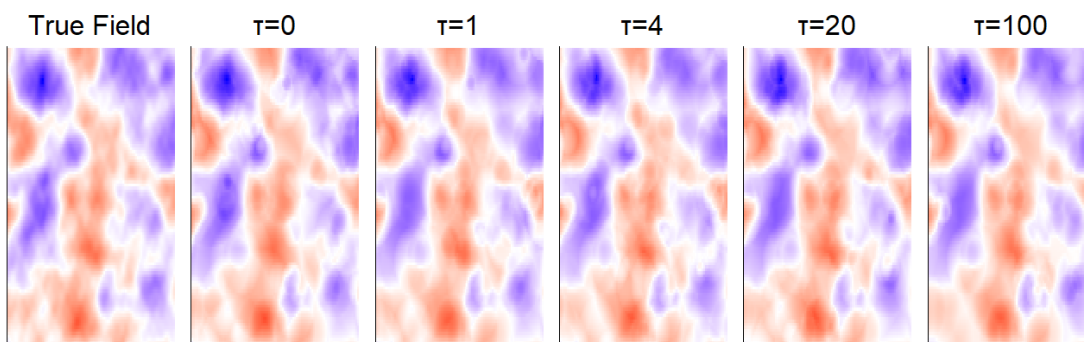
plot_default_uncertainty=ggplot(df_default , aes(x = xx, y = yy ,
  z = int_length , fill = int_length)) + geom_tile()+scale_
  fill_gradient2(low = "blue" , mid = "white" , high = "red" ,
  midpoint =1.2 )+
scale_x_continuous(limits = c(0, 10) ,expand=c(0,0))+scale_y_
  continuous(limits = c(0, 10) ,expand=c(0,0))+
theme(axis.title.x=element_blank())
plot_default_uncertainty

```

We now would like to explore what happens to the predicted surface as we vary the differentiability parameter  $\nu$ , and switch to Wendland kernels in figure 5.3. We can see that the  $\nu = 2$  appears to have characteristics closest to the actual field, and as the smoothness gets higher and higher, the artifacts of the kernels reappear. This is because the model compensates for an inappropriate amount of smoothness in the kernels by adding additional kernels at high resolutions. Recall that  $\nu = 1$  is the default. A Markov Random Field can be introduced in the default setting, with 100 kernels and  $\nu = 1$ , and its precision parameter  $\tau$  can be varied. The predictions are displayed in figure 5.4. We see that  $\tau$  controls some



**Figure 5.3:** Predicted surfaces obtained by varying smoothness parameters.



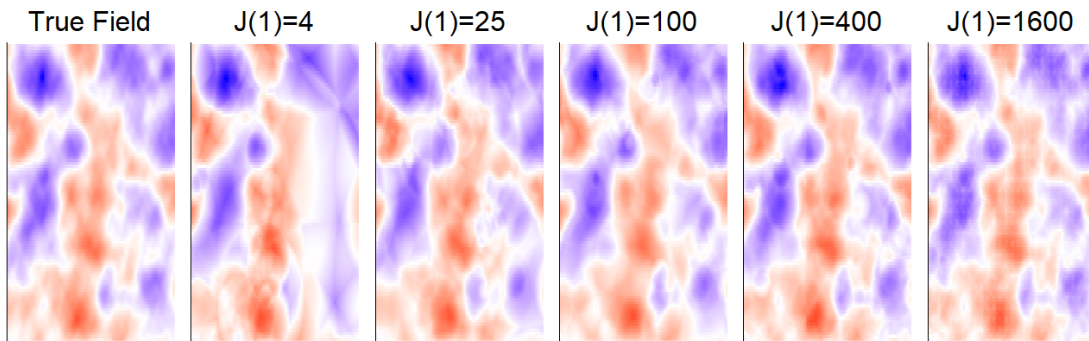
**Figure 5.4:** Predicted surfaces obtained by varying Markov random field parameters.

of the smoothness in the resulting fields. Large values of  $\tau$  lead to GMRF induced smoothness that is inappropriate for the dataset. Thus, the model compensates by adding more resolutions. Recall that  $\tau = 0$  is set as the default.

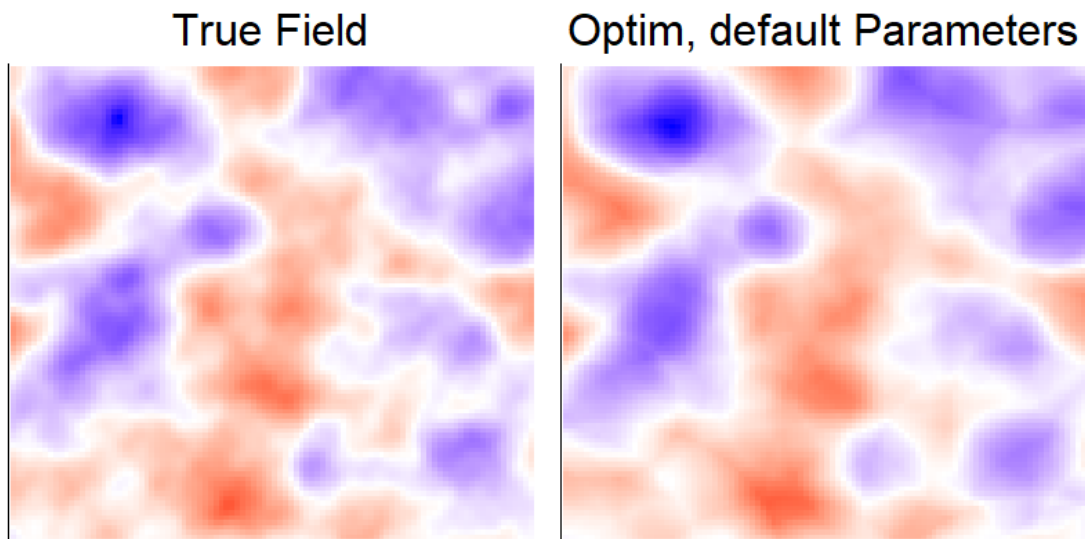
$J(1)$ , the number of kernels at the first resolution, can be varied as well. The results of varying  $J(1)$  from 4 to 1600 are displayed in figure 5.5. Notice that, the presence of kernels artifacts is strong when when  $J(1)$  is either very large or very small.

The same model can be fit using the optimization method using the code below.

```
library(multires)#loads the helper functions
fitt=mr_optim_fit(obs,locations,knots_r1,2)
preds=mr_optim_pred(locations,results)
```



**Figure 5.5:** Predicted surfaces obtained by varying  $J(1)$ .



**Figure 5.6:** Multi-scale spatial optimization fit to the data.

Predictions are displayed in figure 5.6 alongside the actual and the default MSSS plot. As was demonstrated in chapter 3, the results are smoother. All of these plots, save for the models with 4 or 1600 first resolution knots, provide reasonable fits to the data. The flexibility of spatially varying resolution allows for the resulting robustness of the predictive surfaces to different choices of kernel parameters.

If explicit comparison is desired, the marginal model probabilities can be compared between two sets of kernels. Below, we show code to extract the posterior model probabilities from the default model, with  $J(1) = 100$ , to the model with  $J(1) = 400$ .

```

knot_1600_fit=msss_fit(as.matrix(locations),obs,knots_r1,2,
  maxiters=maxiters,cores=numcores,design_mat = design_mat)
default_post_mod_probs=default_fit$cpp_run$top_models_log_
  likelihood[1:100]
knot_1600_post_mod_probs=knot_1600_fit$cpp_run$top_models_log_
  likelihood[1:100]

```

These values are substantially higher for the top models found with  $J(1)=100$ .

### 5.4.1 Implementation Details

The front end of the package is written in R. The code starts by creating the kernel matrix for resolution 1 using the observed locations, the kernel parameters, and the knot locations. The data augmentation necessary for the optional prior for the resolution 1 coefficients is also computed in this step. The resulting kernel matrix is very sparse, due to the compactly supported basis functions. Because of this, a substantial performance and memory footprint advantage can be obtained by using sparse matrix libraries. To this end, our implementation uses the `Matrix` package. The SSE of the linear model with only the fixed effects and resolution 1 knots is computed and stored, as it will be needed for computation of the model probabilities as described in section 5.2.2.

All computations for knots at resolutions higher than 1 are performed in C++ to reduce memory overhead and better leverage parallelization. To call the C++ routine from within R, the `Rcpp` package as used. The sparse matrices from the `Matrix` package must be converted into C++ sparse matrices. This is accomplished through the C++ matrix library Armadillo (Sanderson and Curtin, 2016), and its R interface `RcppArmadillo` (Eddelbuettel and Sanderson, 2014).

To perform shotgun stochastic search, a large of models must be fit in neigh-

borhood of the current model with either one knot added, or one knot removed. To compute the posterior model probability for each of these models, the covariance matrix and coefficients for the new model must be computed. If these computations were performed from scratch for each model, this would be  $O(np^2 + p^3)$ . However, the regression coefficients and covariance matrix can be updated using the formulas given in section 2.2.4 that allow for the updating of the regression coefficients and covariance matrix, which reduce the computational complexity to  $O(n(p+1) + (p+1)^2)$ . These computations can be done in parallel, and achieves its maximum efficiency at approximately 20 cores. With more cores than 20, the communication overhead is such that more cores do not improve the speed. A running tally of the top 100 models is kept by the search routine. To reduce the memory overhead, the design matrices of each of these 100 models are not stored permanently. Only the estimates of coefficients  $\beta$ , a matrix with the knots corresponding to a top model, and a vector of knot resolutions for each of the top models are returned. This however comes at a cost, as the functions that compute interval estimates must reconstruct the design matrix for the entire data, which can be time consuming.

The implementation of the multi-resolution optimization routine relies heavily on the optimization routines in the **SPAMS** optimization toolbox (Mairal et al., 2010), which is maintained by INRIA. To fit the this optimization, **SPAMS** was modified to provide the capability of composing the composite absolute penalty with the Tikhonov norm, which provides for the first resolution Markov random field.

## 5.5 Summary

The package `MSSS` provides an easy to use implementation of the methods developed in chapters 2, 3, and 4. It can be used for parallel computation with large computing environments, and provides many options with respect to the kinds of predicted surfaces that are created. These capabilities have been demonstrated on synthetic data.

# Chapter 6

## Case Study: Prediction and Feature Identification via Spatially Varying Resolution

### 6.1 Introduction

Gaussian processes do not directly model the physical mechanisms that drive the data that are modeled. However, spatially varying resolution results in the explicit characterization of the differences in regions via the number of resolutions active at different locations. This allows practitioners to use these models to not only predict and quantify uncertainty, but also to discover local features in spatial data, and identify regions that merit further investigation.

We will demonstrate this via two case studies. First, we will discuss in detail Heaton et al. (2019), a comparison of twelve alternatives to the full Gaussian Process is presented in the context of a case study competition. Each of the twelve models were implemented by the authors of the model. These models were then

fit on common hardware to provide an honest comparison of the computational performance. Using a satellite dataset of 150,000 land temperature observations from over the Ozark region, the prediction and uncertainty quantification of these models was compared on a holdout set. Modelers were instructed to, if applicable, use an exponential covariance function for their model, and to include fixed effects for latitude and longitude, in addition to an intercept. The data and code required to implement each of these models was published along with the results. We will review this competition, and compare the performance of MSSS to the performance of the other models discussed in terms of prediction, uncertainty quantification, and computational performance. We will then move to the feature identification. In the context of this land surface temperature study, one of the regions with a high posterior estimated number of resolutions is a mountain range. This suggests that adding an exogenous variable, namely elevation, could improve the model and remove the region of small scale behavior. We re-fit the model including elevation as a predictor, and show that the re-fit model no longer assigns a large number of resolutions to this mountain range. Though some of the competitor models considered in this case study can adapt to local features, the explicit identification of regions of fine scale behavior is unique to models that display spatially-varying resolution.

We will next demonstrate the feature identification on sea surface temperature data collected off the cost of California in the summer. During this season, all but the southernmost parts of California experiences an upwelling, with substantially colder temperature occurring close to the shore, and warmer temperature offshore. This results in highly non-stationary spatial process with a large amount of local variation near the shore, and less of this local behavior offshore.

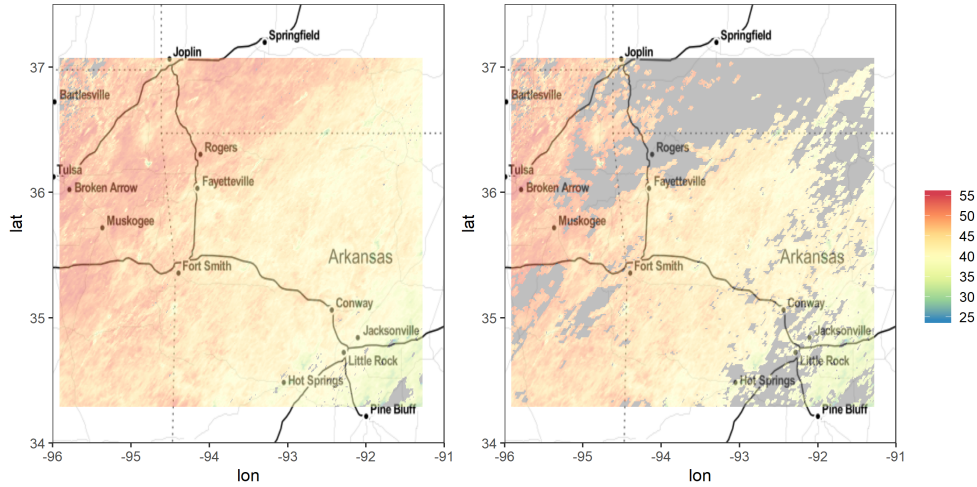


## 6.2 Ozark Data

The data was collected from the MODIS satellite using the Terra instrument on August 4th 2016. Satellite datasets are categorized by their level, which denote the amount of pre-processing that has gone into the data. Level-1 corresponds to raw measurements, rarely in standard units, with calibration information included. This is because, for example, satellites measure temperature indirectly, by measuring radiance. Level-2 data takes the raw measurements and converts them into geophysical units, which in this context would be temperatures. Level-3 data maps these measurements onto a standard grid, and Level-4 data has been derived from either models or multiple measurements, not only from a single instrument. The analyzed dataset of temperature is collected at Level-3, measured in Celsius. This dataset covers the Ozarks, including parts of Arkansas, Kansas, Missouri, and a small part of Nebraska. 150,000 observations, in a 500 by 300 grid, were used in the analysis. This data were chosen because they corresponds to a day with almost no missingness due to cloud cover. The holdout set was constructed using the cloud cover from August 6th, 2016, which results in a realistic holdout set of 42,740 observations. The full data compared to the training data are shown in figure 6.1. Temperatures are generally warmer in the northwest part of the data, and get cooler towards the southeast. This is a difficult domain for prediction as it requires extrapolation, i.e. predicting outside of the domain of the training data.

## 6.3 Competitor Models

The case study involves a number of statistical methods already discussed in chapter 1, LatticeKrig (Nychka et al., 2015), Covariance Tapering (Kaufman



**Figure 6.1:** Full vs. training satellite measurements from August 4th, 2016.

et al., 2008), the Multi Resolution Approximation (Katzfuss, 2017), Spatial Partitioning Kim et al. (2005), the NNGP (Datta et al., 2016), Stochastic PDEs (Lindgren et al., 2011), Fixed rank Kriging (Cressie and Johannesson, 2008), Periodic Embedding (Guinness, 2019), and the predictive process (Banerjee et al., 2008). In addition, several non-statistical approaches are considered in this competition. Metakriging (Guhaniyogi and Banerjee, 2018) proposes to fit a model on  $K$  subsets in parallel, then combine the results using the geometric median of the posteriors of these subsets. The authors show that this is a good approximation of the true posterior as the sample size grows for certain classes of Gaussian Processes. Gapfill (Gerber et al., 2018) is a distribution free method that relies only on local neighborhoods and quantile regression, which can allow this model to display non-stationarity. To provide a prediction at a point, local approximate Gaussian Processes (Gramacy et al., 2016) fit a Gaussian Process to a set of nearest neighbors of the point, and maximize the reduction of the predictive variance at each point. This not only speeds up computation, but allows for the model to fit nonstationary datasets. The models were compared on a number of metrics based on their out of sample predictions and their uncertainty quantification. The

metrics and their definitions are in table 6.1, and more details about the metrics can be found in Gneiting and Raftery (2007).

Metric	Definition
MAE	$\frac{1}{J} \sum_{j=1}^J  y_j - \hat{y}_j $
RMSE	$\sqrt{\frac{1}{J} \sum_{j=1}^J (y_j - \hat{y}_j)^2}$
CRPS	let $\hat{z}_j = \frac{y_j - \hat{y}_j}{\sigma_j}$ , $\Phi$ and $\phi$ be the standard normal cdf and pdf respectively, then $\frac{1}{J} \sum_{j=1}^J \hat{\sigma}_j (\hat{z}_j (2\Phi(\hat{z}_j) - 1) + 2\phi(\hat{z}_j) - 1/\sqrt{\pi})$
Interval score	Let $U_j$ and $L_j$ bet the upper and lower bounds to an interval with confidence level $1 - \alpha$ , then $\frac{1}{J} \sum_{j=1}^J U_j - L_j + \frac{2}{\alpha}(L_j - y_j)\mathbb{1}(y_j < L_j) + \frac{2}{\alpha}(y_j - U_j)\mathbb{1}(y_j < U_j)$
CVG	$\frac{1}{J} \sum_{j=1}^J \mathbb{1}(L_j < y_j < U_j)$

**Table 6.1:** Metrics for model comparison in Heaton et al. (2019).

## 6.4 MSSS Applied to the Ozark Data

Discrete process convolution based approaches cannot directly approximate an exponential covariance function. This is because the convolution that produces an exponential covariance is a spike (Higdon, 2002), but a spike is not a useful basis function for interpolation with a finite number of knots. To mimic the rough nature of this covariance function, (which results in a process that is differentiable only once) we choose a Bezier kernel with  $\nu = .3$ , and a kernel width of 1.5. Rather than the flat prior on the first resolution used in chapter 2, we adopt an intrinsic Markov Random field with a prior precision of 10 on the first resolution. This prior enforces some smoothness and allows for more borrowing strength in the first resolution, which we believe will improve this model’s ability to extrapolate. For computational convenience, we use 20 first resolution knots, and run the model for 100 iterations. Results are in table 6.2.

Convergence in Bayes factor had not yet occurred, but 100 iterations was

deemed to be sufficient as the R squared of the model was only increasing by .0003 at the 100th iteration. MSSS is competitive, neither the best or worst by any metric below. It outperforms the simple low rank methods Fixed Rank Kriging and the Predictive Process quite consistently, and is middle of the pack computationally, but is outperformed by the other multi-resolution approaches in prediction and interval coverage. The computational comparisons are not exactly one to one. The case study was performed on a dedicated computing environment with 256 GB of RAM and 28 2.4 GhZ Xenon cores, while MSSS was run on a shared environment with 32 Xenon cores at 2.7GhZ and 64 GB of RAM available to the user. However, MSSS experiences diminishing returns after about 20 cores (chapter 2), so these computing environments should result in similar performance.

Method	MAE	RMSE	CRPS	INT	CVG	Runtime (min)	Cores Used
MSSS	1.99	2.32	1.40	11.08	0.84	19.06	20
FRK	1.96	2.44	1.44	14.08	0.79	2.32	1
Gapfill	1.33	1.86	1.17	34.78	0.36	1.39	40
LatticeKrig	1.22	1.68	0.87	7.55	0.96	27.92	1
LAGP	1.65	2.08	1.17	10.81	0.83	2.27	40
Metakriging	2.08	2.50	1.44	10.77	0.89	2888.52	30
MRA	1.33	1.85	0.94	8.00	0.92	15.61	1
NNGP	1.21	1.64	0.85	7.57	0.95	2.06	10
Partitioning	1.41	1.80	1.02	10.49	0.86	79.98	55
Predtive Proc	2.15	2.64	1.55	15.51	0.83	160.24	10
SPDE	1.10	1.53	0.83	8.85	0.97	120.33	2
Tapering	1.87	2.45	1.32	10.31	0.93	133.26	1
Periodic Embed.	1.29	1.79	0.91	7.44	0.93	9.81	1

**Table 6.2:** Summary table of results from Heaton et al. (2019) with MSSS at the top

A plot of the predicted temperatures is displayed in figure 6.2, and a spatial residual plot, with truncated scales for clarity, is displayed in figure 6.3. It shows clearly that the model underpredicts in the holdout region at the northern extent of the data.

A feature of our model is the explicit identification which parts of the field

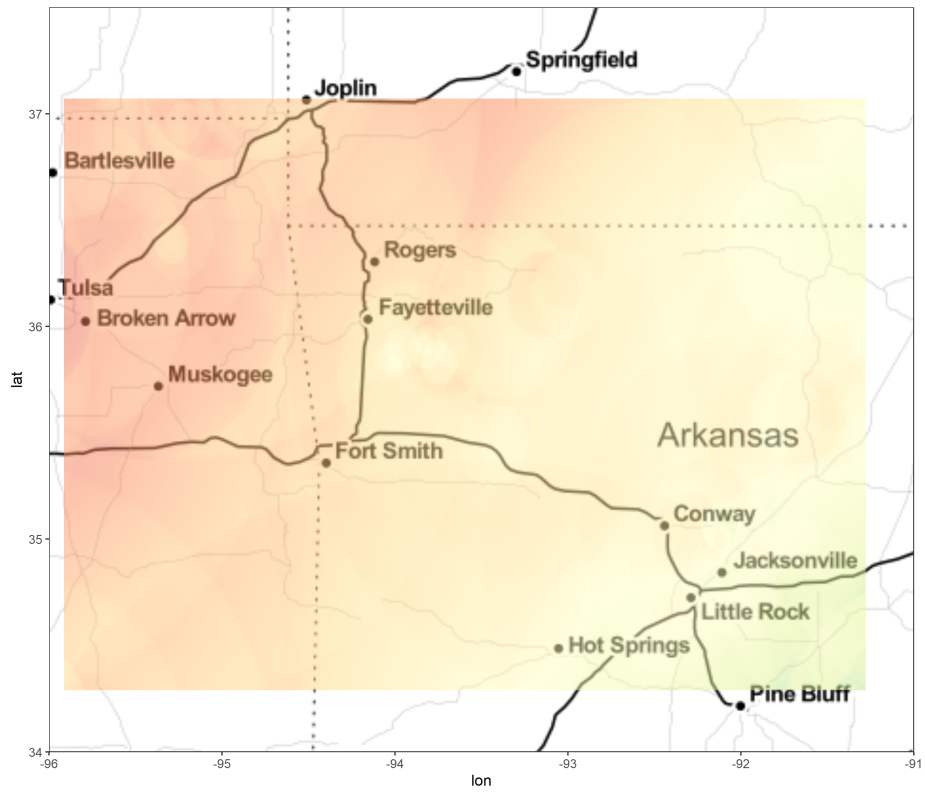


Figure 6.2: Predicted values for this model.

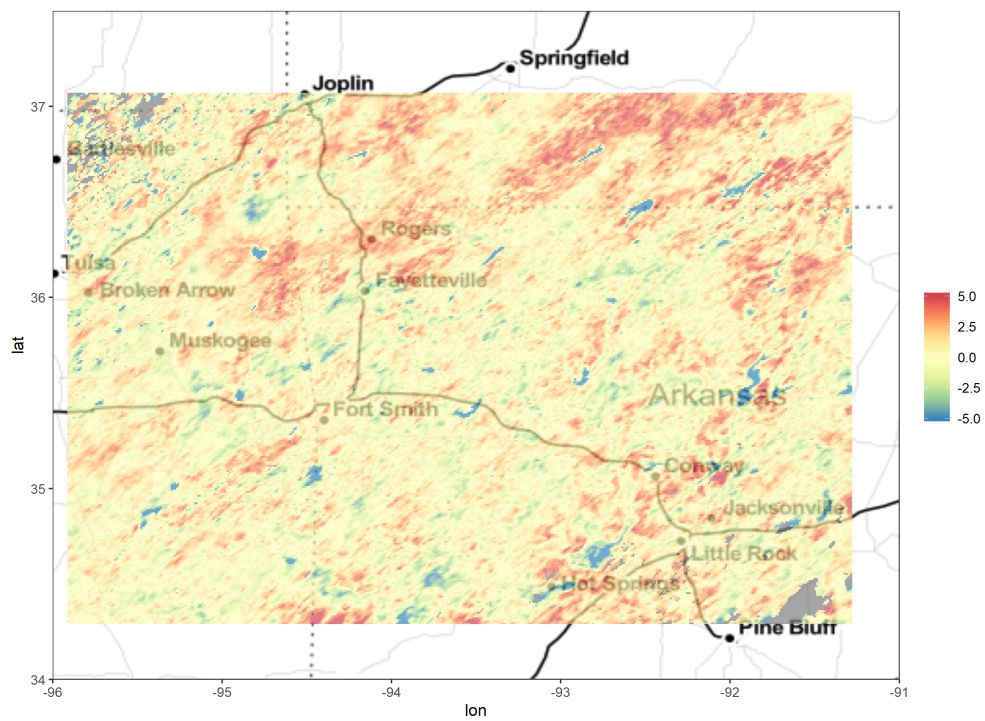


Figure 6.3: Spatial residual plot for this model.

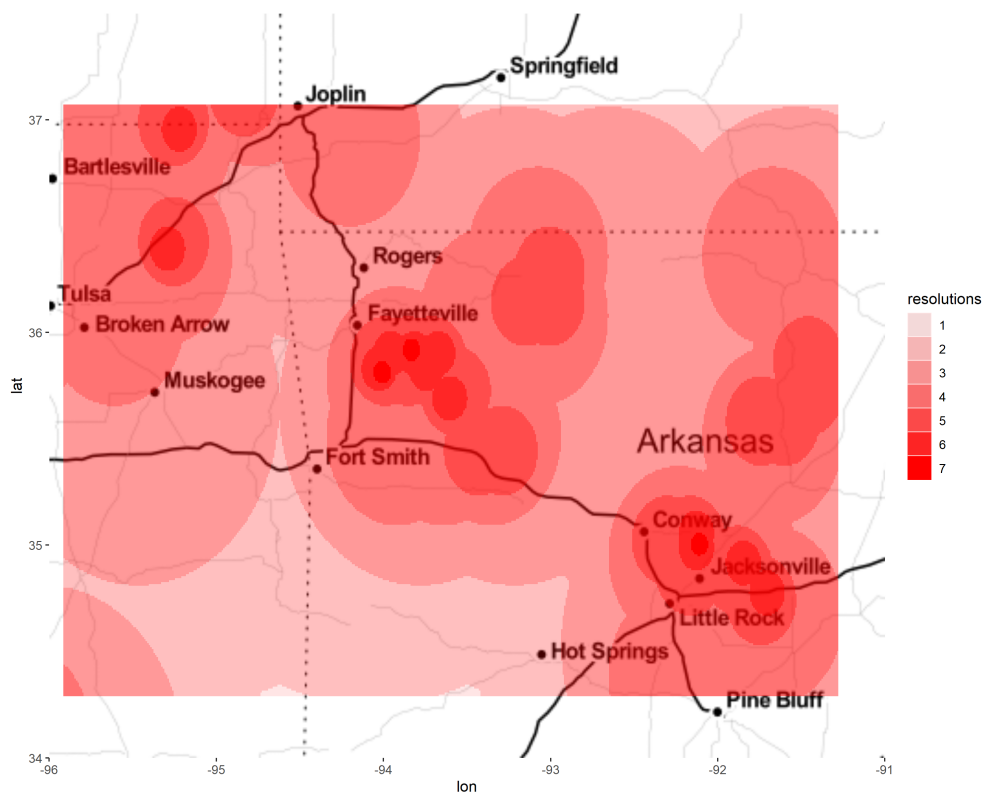
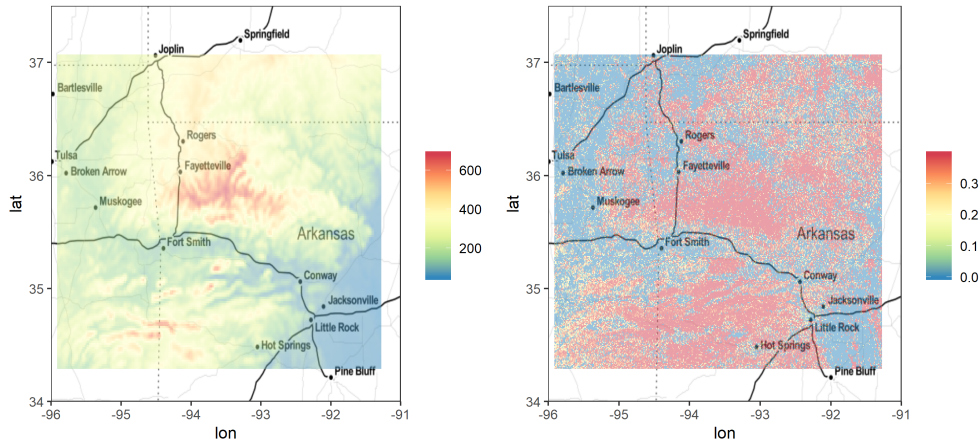
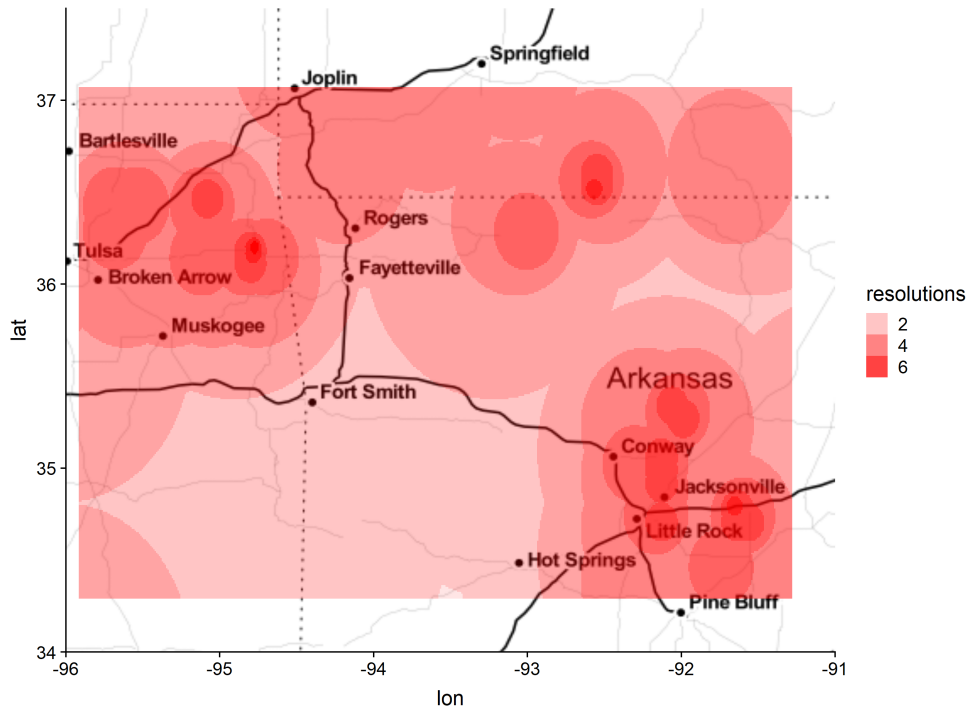


Figure 6.4: Resolutions used for this model.



**Figure 6.5:** Elevation and percent forest coverage over this region.

display more fine scale features through the posterior estimated number of resolutions. This is displayed in figure 6.4. A large posterior number of resolutions are estimated to be in effect north of Fort Smith, and near Little Rock. The area near Little Rock is by far the coldest part of this region, and is fairly small, so a large number of resolutions might be required here. However, the fine scale behavior north of Fort Smith merits further investigation. Two potential exogenous variables that could affect temperature are elevation and forest cover (displayed figure 6.5). Elevations in this region vary from about 200 meters in the southwest, to about 700 meters in the Boston mountains east of Fayetteville, with several other minor mountain ranges appearing in the southern extent of this dataset. These substantial elevation changes are likely to affect temperature, but are not directly modeled in Heaton et al. (2019). Elevation data were sourced from the R package `elevatr` (Hollister and Tarak Shah, 2017). Forest cover has also been shown to have a small effect on land surface temperatures (Alkama and Cescatti, 2016), with more forest cover associated with lower temperatures. Forest cover data was sourced from (Hansen et al., 2013). Elevation had a moderate correlation with the temperature, but forest cover had virtually no linear relationship to temperature in this region.



**Figure 6.6:** Resolution used for the model with elevation as a predictor.

The region north of Fort Smith that had a large number of posterior estimated resolutions appears to correspond to the Boston mountains. It would be more sensible for the effect of elevation to be directly modeled through an elevation predictor instead of forcing a spatial effect. However, since elevation generally increases with latitude, a new variable was created that orthogonalizes these two variables, and then the model was re-fit with this new predictor. The new model was similar in predictive ability to the previous model, but the posterior estimated number of resolutions, shown in figure 6.6, no longer is large in the Boston mountains. This demonstrates how the posterior estimated number of resolutions can identify local features. When those features are accounted for directly in the model, the number of resolutions assigned to those regions drops substantially. Predictions do not differ much from those displayed in figure 6.2



## 6.5 California Sea Surface Temperature Data

This level 3 satellite data were collected by the MODIS sensor and converted to SST by Goddard's Ocean Biology Processing Group. Data are collected on a  $.0125$  longitude  $\times$   $.0125$  latitude grid off of the West coast of the US, which works out to approximately a 1.5 kilometer resolution. We chose to analyze a monthly composite dataset to avoid any missingness due to cloud cover, and used data from July of 2003, which was a time with a very strong upwelling. We restricted the Latitude to be between 31 and 42 degrees North, and the longitude to be between 128 and 115 degrees West. This results in approximately 500,000 observations. The data are displayed in figure 6.7. We can see from this map that in Northern California, close to the shore, temperatures are much lower than the water farther from the shore. However, the area close to the shore in Southern California is very warm. This is due to an upwelling that occurs in the summer. Cold water from the ocean floor is driven to the shore by currents and winds, but the bathymetry of the southern part of California's coastal region prevents this cold water from reaching all the way to the shore. In this part of California, the upwelling and its resulting lower water temperatures are located offshore to the west of the Channel Islands. In this area, the ocean floor gets much deeper. This can be seen in figure 6.8. This prevents the cold water upwelling from occurring any closer to the shore.

The multi-resolution structure of the fitted MSSS is able to identify this feature. MSSS was run on this data for 100 iterations with latitude included as a linear predictor in the model, and a Markov Random field with  $\tau = 10$  for first resolution coefficients. Predictions were displayed in figure 6.9. The monthly averaged temperatures are quite smooth, and the resulting model fit is extremely good, with an in sample RMSE  $.30$  and a mean absolute error of  $.23$ . The areas

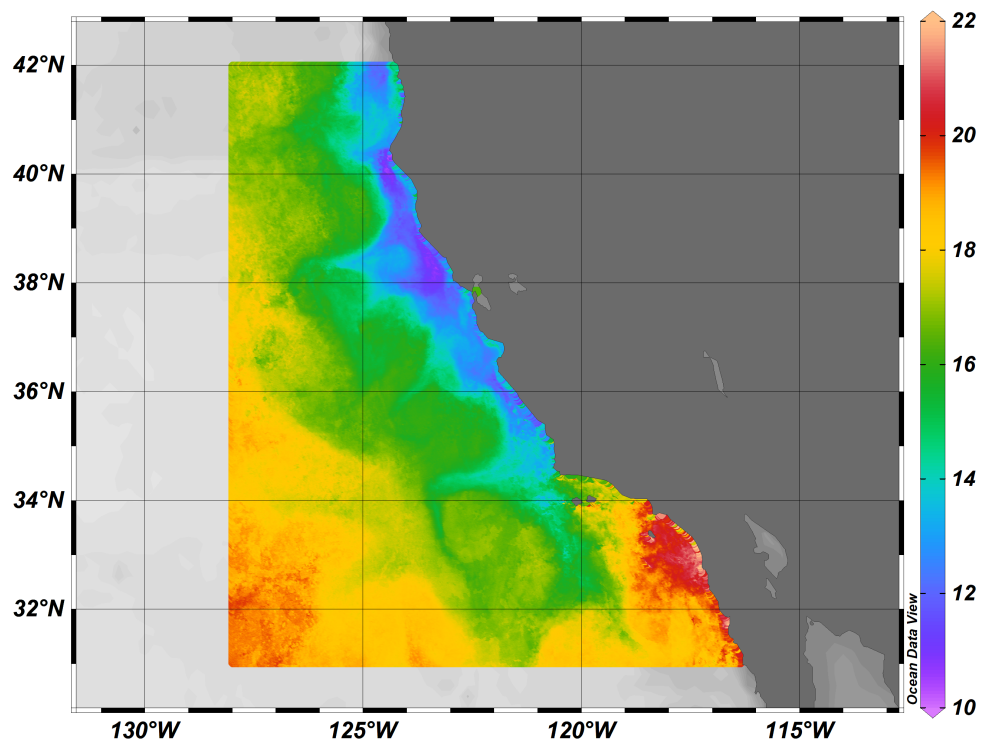


Figure 6.7: July 2003 sea surface temperatures off the coast of California.

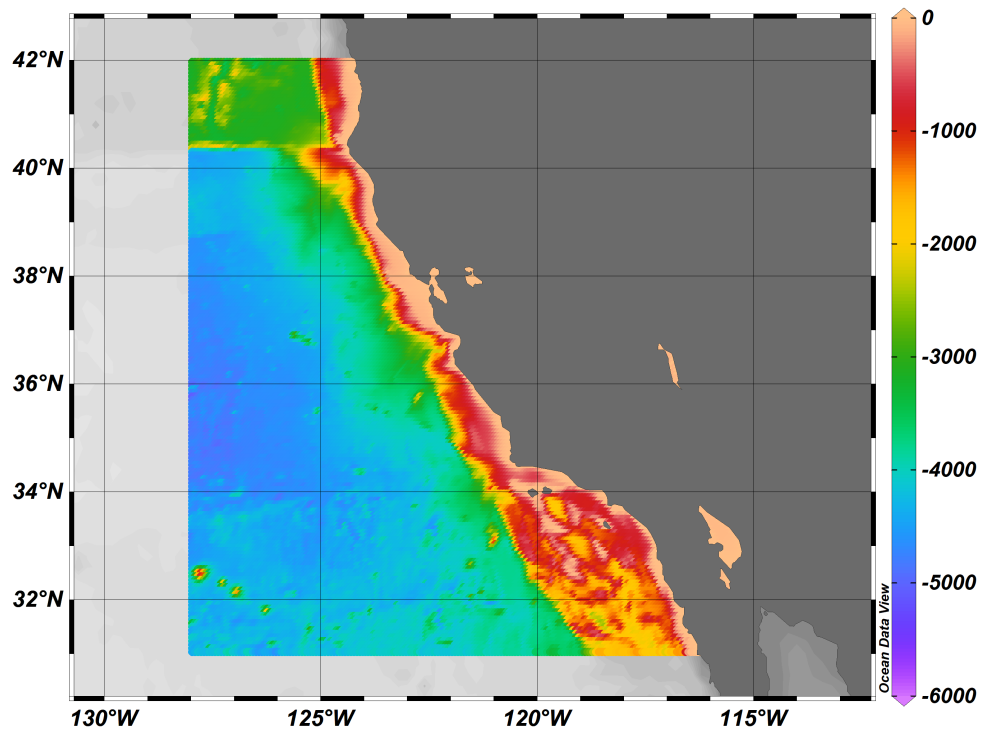
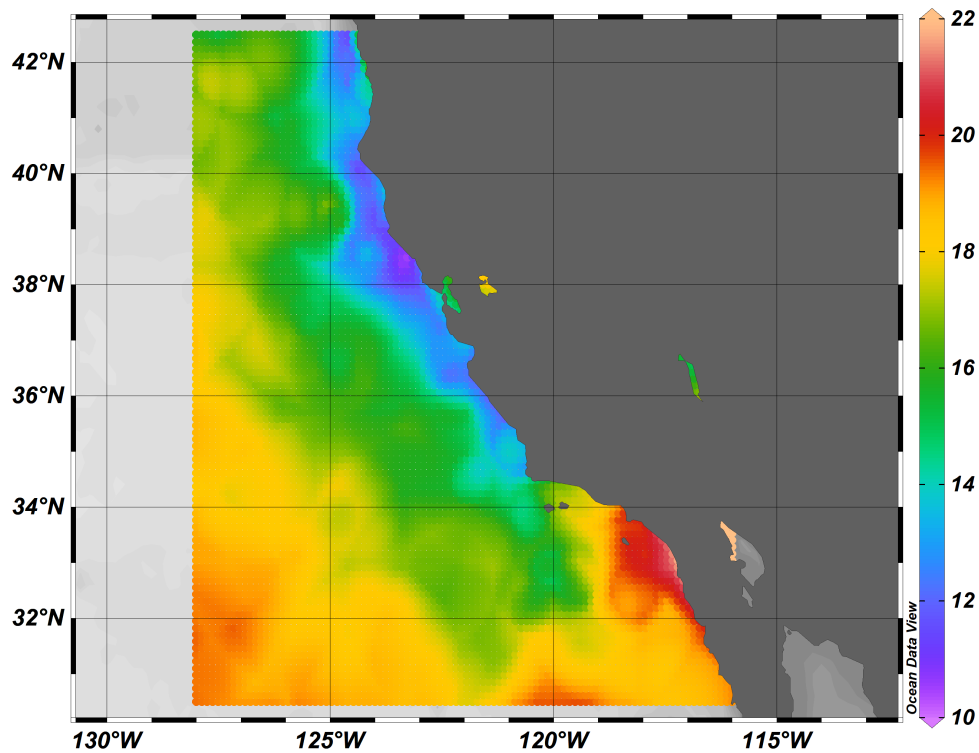


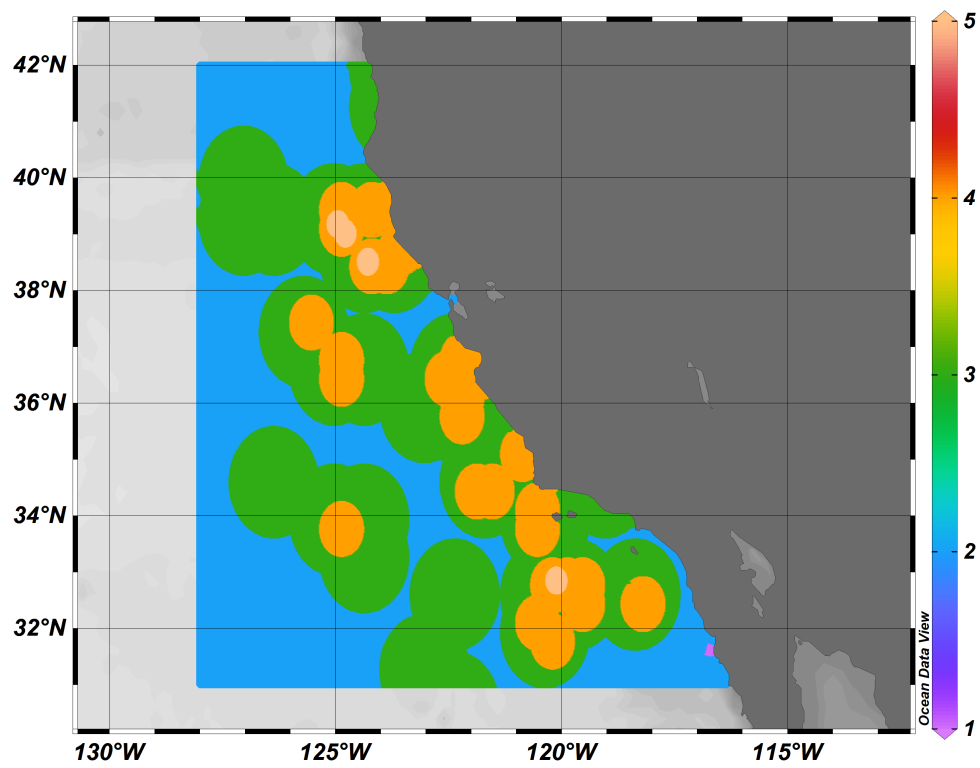
Figure 6.8: Bathymetry off the coast of California.



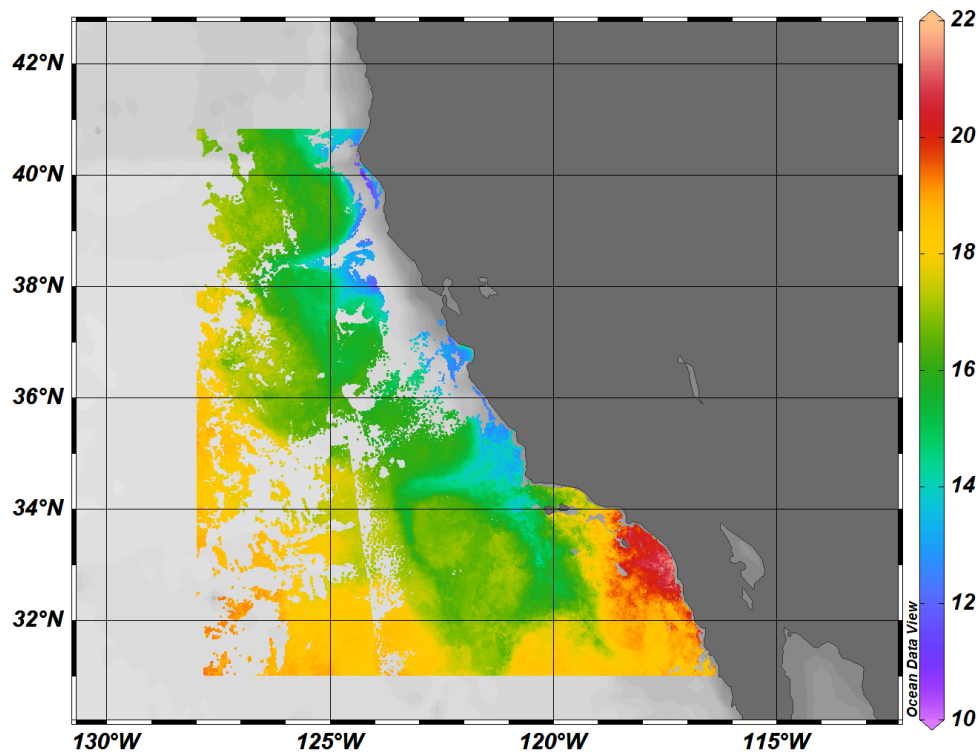
**Figure 6.9:** Fitted values on a grid off the coast of California after 100 iterations of MSSS.

with a large number of resolutions, displayed in figure 6.10, largely follow the border of the shallower region near the shore.

To demonstrate predictive performance and multiresolution structure identification in an example with missing data, the following procedure was followed. First, a three day composite dataset from July third 2003 was selected due to its representative missing data pattern, which is shown in figure 6.11. Northern California was quite foggy during that three day window, so a large amount of missingness was found in that region, but Southern California was mostly spared. MSSS was then fit to that data with the same parameters as in the above analysis. The resulting predictions are displayed in figure 6.12. In it, we can see that the vast majority of the upwelling is still picked up, but there is a small region near Point Arena that is assigned a higher temperature. This is an edge effect brought



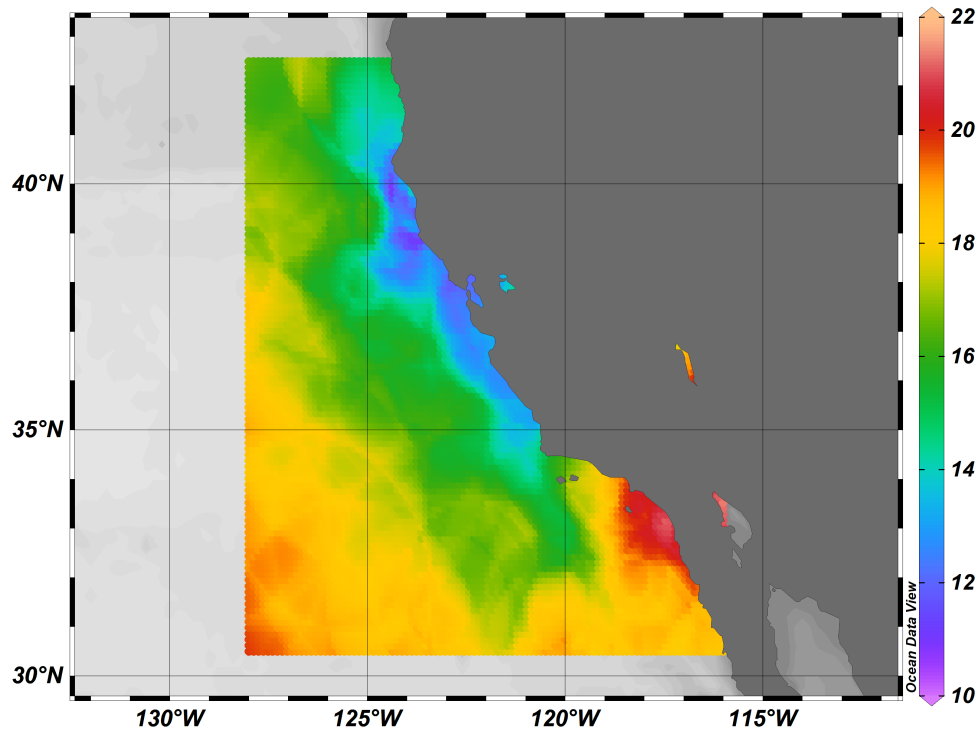
**Figure 6.10:** Posterior expected number of resolutions active off the coast of California after 100 iterations of MSSS.



**Figure 6.11:** Missing pattern off the cost of California

on by the extreme missingness along the shore. The resulting plot of resolutions used, displayed in figure 6.13, demonstrated similar behavior to the full data. The regions with a large number of resolutions followed generally the border of the upwelling. However, the region with little to no data, along the shore between 36 and 40 degrees North, had a smaller number of resolutions. The out of sample mean absolute error was .4, and 95% interval coverage probability was .76.

To provide a basis for comparison, NNGP and `LatticeKrig` were also fit to this data. With NNGP, 10 neighbors, exponential covariance, and the latent response model were used, and for `LatticeKrig`, 3 resolutions and 30 basis functions were used. NNGP resulted in a mean absolute error of .64 and a 95% interval coverage probability of .65, and due to the MCMC based inference, substantially slower performance than MSSS. `LatticeKrig` had similar predictive and intrval

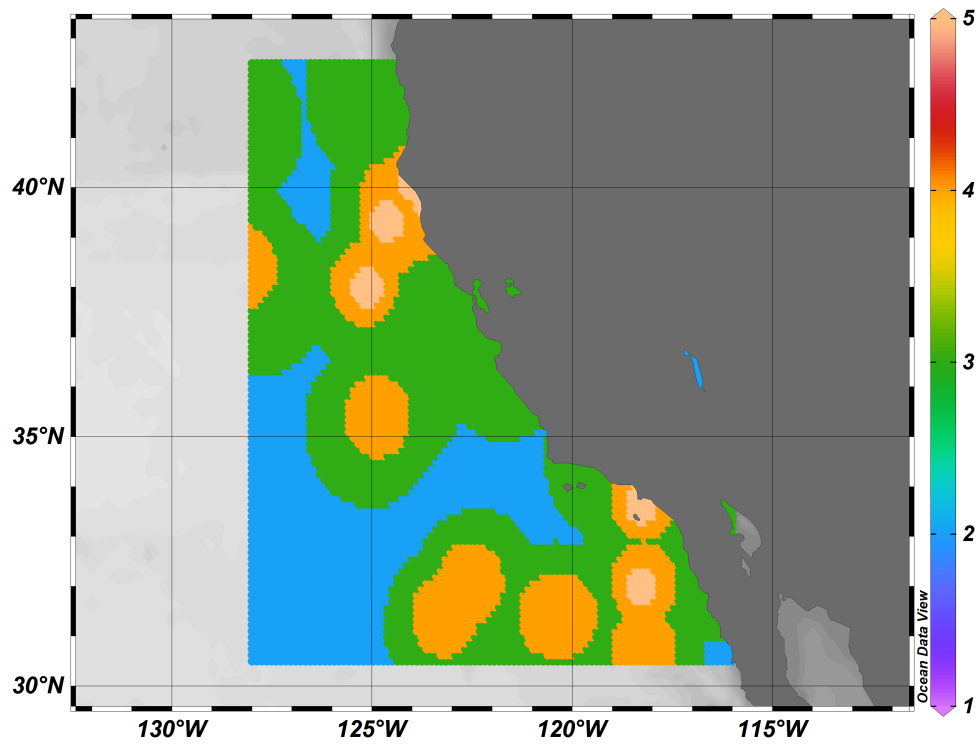


**Figure 6.12:** Predicted temperatures off the cost of California with missing data

performance to MSSS with an out of sample mean absolute error of .41 and a 95% interval coverage probability of .87.

## 6.6 Conclusion

The analyses in this chapter demonstrate the feature identification properties of multi-resolution process convolutions with spatially varying resolution. Specifically, regions that have a high estimated number of resolutions have been identified to have more small scale variation. These regions often can be interpreted in the scientific context of the dataset. Procedures that result in spatially varying resolution provide more than competitive prediction accuracy and uncertainty quantification properties. They also can identify regions of interest in a spatial domain.



**Figure 6.13:** Estimated number of resolutions off the coast of California with missing data

# Chapter 7

## Conclusion

This work has developed methods for a new characterization of non-stationarity in spatial fields through spatially varying resolution. We have performed selection of potential basis function sets with this property via stochastic search, and via a LASSO like penalty, and discussed how to perform prediction and uncertainty quantification via these models. Due to parallelization and modern optimization methodology, these models can be fit to large datasets. We have also developed software to make these models available to practitioners and demonstrated how to use these methods to identify hidden local features in spatial datasets.

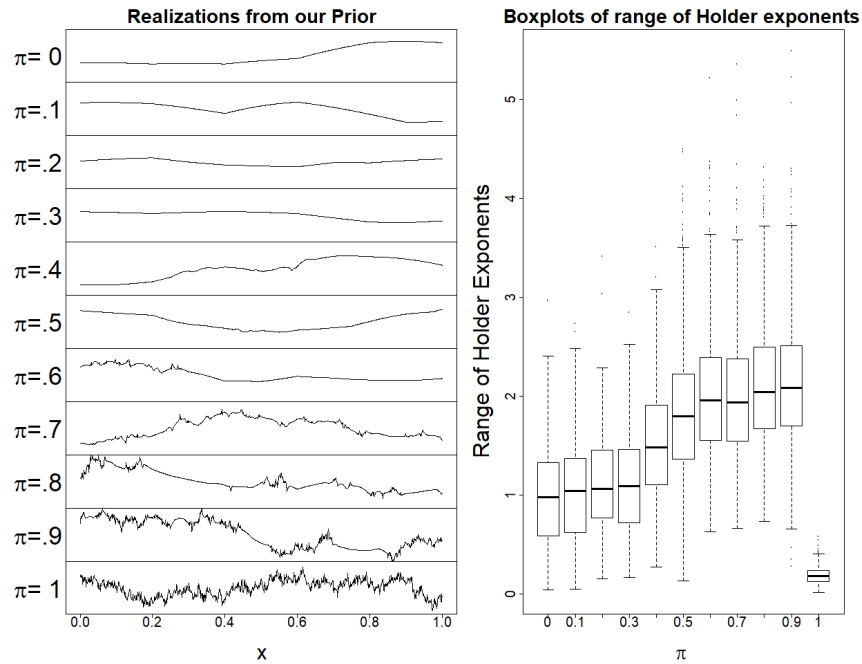
Several extensions and enhancements could be pursued. One of the most obvious is extension to a binomial, or other GLM family response types. This could be pursued through the optimization approach of chapter 3 by altering the loss function. Laplace approximations for the GLM setting have been developed for hyper-g priors (Bové et al., 2011), and these could be applied within the stochastic search approach of chapter 2.

One potential avenue of the exploration of non-stationary models in general is multi-fractal analysis (Jaffard et al., 2006), which uses discrete wavelet techniques to measure local signal regularity via Holder exponents. Some preliminary em-



pirical exploration has been completed in the context of the model of chapter 2. To explore how spatially varying resolution causes the local signal regularity to change, and how the varying degrees of sparseness controlled by  $\pi$  can affect this, 100 trajectories from priors corresponding to grid of values for ranging from 0 to 1, in a one dimensional setting, were simulated. We then perform a multi-fractal analysis by recording the resulting ranges of Holder exponents. In all our simulations we fix  $J(1) = 7$ . Notice that  $\pi = 0$  results in just the 7 knots, and for increasing  $\pi$ , the average number of resolutions and knots will increase. For the  $\pi = 1$  example, we truncate the maximum number of resolutions to 10, but as shown in section 2.2.1, no truncation is necessary for  $\pi < .5$ . Next, conditional on the knots and locations, a design matrix will be generated from our Bezier kernel, with  $\phi_r = 2.5$  and  $\nu = 1$ . Finally, for each knot  $s_j^r$ , we generate  $\beta_j^r \sim N(0, 1/r^2)$ , which makes the coefficients on average smaller at higher resolutions. A wide range of Holder exponents suggests that the fractal behavior varies substantially in the resulting curve, which means that the roughness of the response curve differs at different points in the domain. Results are displayed in figure 7.1, where a clear increasing trend is observed in the range of Holder exponents, save for  $\pi = 1$ , as, in such case, the Holder exponents are virtually unchanged in the space. This makes intuitive sense, since for a dense multi-resolution grid, the resolution is not spatially varying.

Alternatives to the stochastic process prior for the space of potential models developed in section 2.2.1 that encourage different kinds of patterns of sparsity are possible. One potential prior that would knots at a resolution to be accompanied by more knots nearby at the same resolution, if we let  $Nb(\gamma_j^r)$  be the set of neighbors to a knot at the same resolution,



**Figure 7.1:** Left panel: randomly selected realizations from our prior, one for each sparsity level, plotted on the same axes. Smaller values of  $\pi$  correspond to stronger spatial variability of the roughness of the sample paths. For  $\pi = 1$  we observe homogeneous local variability across space. Right panel: distribution of Holder exponents as a function of  $\pi$ . An evident increasing pattern is present, except for  $\pi = 1$ , where Holder exponents typically have a very small range, indicating that the fields with dense multi-resolution grids do not exhibit multi-fractal behavior.

$$Pr(\gamma_j^r = 1 | \boldsymbol{\gamma}^{r-1}, Nb(\gamma_j^r)) = \begin{cases} \pi^{\frac{1}{1+\sum Nb(\gamma_j^r)}} \text{ with } N_j^r = \sum_{j=1}^{J(r-1)} \gamma_j^{r-1}, & \text{if } \gamma_{\lfloor \frac{j-1}{2^d} \rfloor + 1}^{r-1} = 1 \\ 0, & \text{if } \gamma_{\lfloor \frac{j-1}{2^d} \rfloor + 1}^{r-1} = 0. \end{cases}$$

Another possibility that would strongly encourage large portions of a resolution to enter the model at the same time would be

$$Pr(\gamma_j^r = 1 | \boldsymbol{\gamma}^{r-1}) = \begin{cases} \pi^{\frac{1}{G_{r-1}}} \text{ with } G_{r-1} = \sum_{j=1}^{J(r-1)} \gamma_j^{r-1}, & \text{if } \gamma_{\lfloor \frac{j-1}{2^d} \rfloor + 1}^{r-1} = 1 \\ 0, & \text{if } \gamma_{\lfloor \frac{j-1}{2^d} \rfloor + 1}^{r-1} = 0. \end{cases},$$

though evaluation of this would be more challenging.

Another enhancement that might be pursued is the exploration of prior distribution other than the hyper-g prior for the distribution on the slab in chapter 2. Priors with accessible Laplace approximations, such as the non-local prior (Johnson and Rossell, 2012) could be explored. Priors that have the property of spatially varying shrinkage, such as the tree shrinkage prior from (Guhaniyogi and Sansó, 2017), could also be explored. This would break the conjugacy necessary for MSSS, so an approximation or some kind of empirical Bayes procedure would need to be developed.

Computational improvements are potentially possible within the optimization approach. The main computational bottleneck is the initial value. This could potentially be improved via a convex relaxation (Obozinski and Bach, 2012), where the prior is approximated by a convex function, which can then be optimized more quickly. Extending chapter 2 to a spatiotemporal setting will break the conjugacy necessary for computational efficiency and stochastic search. An RJMCMC approach would likely be necessary, but slow. It is possible that the maximization

approach from chapter 3 could be extended to the time domain in a number of ways. If the goal is feature identification at different times, the multi-resolution structure could be allowed to change at different times, but a strong prior could encourage similarity in the coefficients. Something similar to

$$p(\beta_{r,j,t}) = N(\beta_{r,j,t-1}, \sigma_t)$$

with  $\sigma_t$  small could potentially be used within the maximization scheme via the data augmentation strategy discussed in chapter 3. However, this would likely be computationally intensive, so would need to be coupled with computational improvements.

# Bibliography

- Alkama, R. and A. Cescatti (2016). Biophysical climate impacts of recent changes in global forest cover. *Science* 351(6273), 600–604.
- Anderson, C., D. Lee, and N. Dean (2014). Identifying clusters in Bayesian disease mapping. *Biostatistics* 15(3), 457–469.
- Armagan, A., D. B. Dunson, and J. Lee (2013). Generalized double pareto shrinkage. *Statistica Sinica* 23(1), 119.
- Bach, F. R. (2008). Consistency of the group lasso and multiple kernel learning. *Journal of Machine Learning Research* 9(Jun), 1179–1225.
- Bach, F. R. (2009). Exploring large feature spaces with hierarchical multiple kernel learning. In *Advances in neural information processing systems*, pp. 105–112.
- Banerjee, S. (2017, 06). High-dimensional Bayesian geostatistics. *Bayesian Anal.* 12(2), 583–614.
- Banerjee, S., A. E. Gelfand, A. O. Finley, and H. Sang (2008). Gaussian predictive process models for large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70(4), 825–848.
- Bayarri, M. J., J. O. Berger, A. Forte, G. García-Donato, et al. (2012). Criteria for Bayesian model choice with application to variable selection. *The Annals of statistics* 40(3), 1550–1577.
- Beck, A. (2017). *First-order methods in optimization*, Volume 25. SIAM.
- Beck, A. and M. Teboulle (2009). A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences* 2(1), 183–202.
- Benedetti, M., V. Berrocal, and N. Narisetty (2018). Identifying regions of inhomogeneities in spatial processes via an m-ra and mixture priors. Technical report, Technical report, University of Michigan.
- Bickel, P. J., Y. Ritov, A. B. Tsybakov, et al. (2009). Simultaneous analysis of LASSO and Dantzig selector. *The Annals of Statistics* 37(4), 1705–1732.

- Bornn, L., G. Shaddick, and J. V. Zidek (2012). Modeling nonstationary processes through dimension expansion. *Journal of the American Statistical Association* 107(497), 281–289.
- Bottolo, L., S. Richardson, et al. (2010). Evolutionary stochastic search for Bayesian model exploration. *Bayesian Analysis* 5(3), 583–618.
- Bové, D. S., L. Held, et al. (2011). Hyper- $g$  priors for generalized linear models. *Bayesian Analysis* 6(3), 387–410.
- Brenning, A. (2001). *Geostatistics without stationarity assumptions within geographical information systems*. Citeseer.
- Castillo, I., J. Schmidt-Hieber, A. Van der Vaart, et al. (2015). Bayesian linear regression with sparse priors. *The Annals of Statistics* 43(5), 1986–2018.
- Chan, G. and A. T. Wood (1999). Simulation of stationary Gaussian vector fields. *Statistics and Computing* 9(4), 265–268.
- Chen, X., Q. Lin, S. Kim, J. G. Carbonell, E. P. Xing, et al. (2012). Smoothing proximal gradient method for general structured sparse regression. *The Annals of Applied Statistics* 6(2), 719–752.
- Chipman, H. A., E. I. George, and R. E. McCulloch (1998). Bayesian cart model search. *Journal of the American Statistical Association* 93(443), 935–948.
- Chipman, H. A., E. I. George, R. E. McCulloch, et al. (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics* 4(1), 266–298.
- Chung, K. L. (2012). *Elementary probability theory with stochastic processes*. Springer Science & Business Media.
- Clyde, M. A., J. Ghosh, and M. L. Littman (2011). Bayesian adaptive sampling for variable selection and model averaging. *Journal of Computational and Graphical Statistics* 20(1), 80–101.
- Cressie, N. and G. Johannesson (2008). Fixed rank kriging for very large spatial data sets. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70(1), 209–226.
- Datta, A., S. Banerjee, A. O. Finley, and A. E. Gelfand (2016). Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets. *Journal of the American Statistical Association* 111(514), 800–812.
- Eddelbuettel, D. and R. François (2011). Rcpp: Seamless R and C++ integration. *Journal of Statistical Software* 40(8), 1–18.

- Eddelbuettel, D. and C. Sanderson (2014, March). Rcpparmadillo: Accelerating r with high-performance c++ linear algebra. *Computational Statistics and Data Analysis* 71, 1054–1063.
- Efron, B., T. Hastie, I. Johnstone, R. Tibshirani, et al. (2004). Least angle regression. *The Annals of Statistics* 32(2), 407–499.
- Finley, A., A. Datta, and S. Banerjee (2017). spnngp: spatial regression models for large datasets using nearest neighbor Gaussian processes. *R package version 0.1 1*.
- Finley, A. O., H. Sang, S. Banerjee, and A. E. Gelfand (2009). Improving the performance of predictive process modeling for large datasets. *Computational statistics & data analysis* 53(8), 2873–2884.
- Fraley, C. and D. Percival (2015). Model-averaged l1 regularization using Markov chain Monte Carlo model composition. *Journal of Statistical Computation and Simulation* 85(6), 1090–1101.
- Francom, D., B. Sansó, A. Kupresanin, and G. Johannesson (2018). Sensitivity analysis and emulation for functional data using Bayesian adaptive splines. *Statistica Sinica*, 791–816.
- Friedman, J., T. Hastie, H. Höfling, R. Tibshirani, et al. (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics* 1(2), 302–332.
- Fuentes, M. and R. L. Smith (2001). A new class of nonstationary spatial models. Technical report, Technical report, North Carolina State University, Raleigh, NC.
- Furrer, R., M. G. Genton, and D. Nychka (2006). Covariance tapering for interpolation of large spatial datasets. *Journal of Computational and Graphical Statistics* 15(3), 502–523.
- Garcia-Donato, G. and M. A. Martinez-Beneito (2013). On sampling strategies in Bayesian variable selection problems with large model spaces. *Journal of the American Statistical Association* 108(501), 340–352.
- Gelfand, A. E., P. Diggle, P. Guttorp, and M. Fuentes (2010). *Handbook of spatial statistics*. CRC press.
- Gelman, A. and J. Hill (2006). *Data analysis using regression and multilevel/hierarchical models*. Cambridge University Press.
- George, E. and D. P. Foster (2000). Calibration and empirical Bayes variable selection. *Biometrika* 87(4), 731–747.

- George, E. I. and R. E. McCulloch (1993). Variable selection via gibbs sampling. *Journal of the American Statistical Association* 88(423), 881–889.
- George, E. I. and R. E. McCulloch (1997). Approaches for Bayesian variable selection. *Statistica Sinica*, 339–373.
- Gerber, F. et al. Predicting missing values in spatio-temporal satellite data. arxiv 2016. *arXiv preprint arXiv:1605.01038*.
- Gerber, F., R. de Jong, M. E. Schaepman, G. Schaepman-Strub, and R. Furrer (2018). Predicting missing values in spatio-temporal remote sensing data. *IEEE Transactions on Geoscience and Remote Sensing* 56(5), 2841–2853.
- Gneiting, T. and A. E. Raftery (2007). Strictly proper scoring rules, prediction, and estimation. *Journal of the American Statistical Association* 102(477), 359–378.
- Gramacy, R. B. et al. (2016). lagp: large-scale spatial modeling via local approximate Gaussian processes in r. *Journal of Statistical Software* 72(1), 1–46.
- Gramacy, R. B. and H. K. H. Lee (2008). Bayesian treed Gaussian process models with an application to computer modeling. *Journal of the American Statistical Association* 103(483), 1119–1130.
- Green, P. J. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82(4), 711–732.
- Green, P. J., K. Łatuszyński, M. Pereyra, and C. P. Robert (2015). Bayesian computation: a summary of the current state, and samples backwards and forwards. *Statistics and Computing* 25(4), 835–862.
- Guhaniyogi, R. and S. Banerjee (2018). Meta-kriging: Scalable Bayesian modeling and inference for massive spatial datasets. *Technometrics* (just-accepted).
- Guhaniyogi, R. and B. Sansó (2017). Large multi-scale spatial modeling using tree shrinkage priors. *Statistica Sinica*.
- Guinness, J. (2019). Spectral density estimation for random fields via periodic embeddings. *Biometrika* 106(2), 267–286.
- Hahn, P. R. and C. M. Carvalho (2015). Decoupling shrinkage and selection in Bayesian linear models: a posterior summary perspective. *Journal of the American Statistical Association* 110(509), 435–448.
- Hans, C., A. Dobra, and M. West (2007). Shotgun stochastic search for “large p” regression. *Journal of the American Statistical Association* 102(478), 507–516.



- Hansen, M. C., P. V. Potapov, R. Moore, M. Hancher, S. A. Turubanova, A. Tyukavina, D. Thau, S. Stehman, S. J. Goetz, T. R. Loveland, et al. (2013). High-resolution global maps of 21st-century forest cover change. *science* 342(6160), 850–853.
- Hastie, T., R. Tibshirani, and J. Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.
- Heaton, M. J., W. F. Christensen, and M. A. Terres (2017). Nonstationary Gaussian process models using spatial hierarchical clustering from finite differences. *Technometrics* 59(1), 93–101.
- Heaton, M. J., A. Datta, A. O. Finley, R. Furrer, J. Guinness, R. Guhaniyogi, F. Gerber, R. B. Gramacy, D. Hammerling, M. Katzfuss, et al. (2019). A case study competition among methods for analyzing large spatial data. *Journal of Agricultural, Biological and Environmental Statistics* 24(3), 398–425.
- Herbei, R. and L. M. Berliner (2014). Estimating ocean circulation: an MCMC approach with approximated likelihoods via the Bernoulli factory. *Journal of the American Statistical Association* 109(507), 944–954.
- Higdon, D. (1998). A process-convolution approach to modelling temperatures in the north atlantic ocean. *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*, pp. 37–56. Springer.
- Hollister, J. and Tarak Shah (2017). *elevatr: Access Elevation Data from Various APIs*. R package version 0.1.3, doi:10.5281/zenodo.400259.
- Iancu, C., S. Hofmeyr, F. Blagojević, and Y. Zheng (2010). Oversubscription on multicore processors. In *2010 IEEE International Symposium on Parallel & Distributed Processing (IPDPS)*, pp. 1–11. IEEE.
- Ishwaran, H., J. S. Rao, et al. (2005). Spike and slab variable selection: frequentist and Bayesian strategies. *The Annals of Statistics* 33(2), 730–773.
- Jaffard, S., B. Lashermes, and P. Abry (2006). Wavelet leaders in multifractal analysis. In *Wavelet Analysis and Applications*, pp. 201–246. Springer.
- Jenatton, R., J.-Y. Audibert, and F. Bach (2011). Structured variable selection with sparsity-inducing norms. *Journal of Machine Learning Research* 12(Oct), 2777–2824.

- Johnson, V. E. and D. Rossell (2012). Bayesian model selection in high-dimensional settings. *Journal of the American Statistical Association* 107(498), 649–660.
- Karspeck, A. R., A. Kaplan, and S. R. Sain (2012). Bayesian modelling and ensemble reconstruction of mid-scale spatial variability in north atlantic sea-surface temperatures for 1850–2008. *Quarterly Journal of the Royal Meteorological Society* 138(662), 234–248.
- Kass, R. E. and L. Wasserman (1995). A reference Bayesian test for nested hypotheses and its relationship to the schwarz criterion. *Journal of the American Statistical Association* 90(431), 928–934.
- Katzfuss, M. (2017). A multi-resolution approximation for massive spatial datasets. *Journal of the American Statistical Association* 112(517), 201–214.
- Katzfuss, M. and D. Hammerling (2017). Parallel inference for massive distributed spatial data using low-rank models. *Statistics and Computing* 27(2), 363–375.
- Kaufman, C. G., M. J. Schervish, and D. W. Nychka (2008). Covariance tapering for likelihood-based estimation in large spatial data sets. *Journal of the American Statistical Association* 103(484), 1545–1555.
- Kim, H.-M., B. K. Mallick, and C. Holmes (2005). Analyzing nonstationary spatial data using piecewise Gaussian processes. *Journal of the American Statistical Association* 100(470), 653–668.
- Kirsner, D. and B. Sansó (2020). Multiscale shotgun stochastic search for large spatial datasets. *Computational Statistics & Data Analysis*, 106931.
- Koslovsky, M., M. Swartz, L. Leon-Novelo, W. Chan, and A. Wilkinson (2018). Using the em algorithm for Bayesian variable selection in logistic regression models with related covariates. *Journal of Statistical Computation and Simulation* 88(3), 575–596.
- Kyung, M., J. Gill, M. Ghosh, G. Casella, et al. (2010). Penalized regression, standard errors, and Bayesian lassos. *Bayesian Analysis* 5(2), 369–411.
- Lasinio, G. J., G. Mastrantonio, and A. Pollice (2013). Discussing the “big n problem”. *Statistical Methods & Applications* 22(1), 97–112.
- Lemos, R. T. and B. Sansó (2009). A spatio-temporal model for mean, anomaly, and trend fields of north atlantic sea surface temperature. *Journal of the American Statistical Association* 104(485), 5–18.

- Lemos, R. T. and B. Sansó (2012). Conditionally linear models for non-homogeneous spatial random fields. *Statistical Methodology* 9(1), 275 – 284. Special Issue on Astrostatistics + Special Issue on Spatial Statistics.
- Li, Y. and M. A. Clyde (2018). Mixtures of g-priors in generalized linear models. *Journal of the American Statistical Association* 113(524), 1828–1845.
- Liang, F., I. H. Jin, Q. Song, and J. S. Liu (2016). An adaptive exchange algorithm for sampling from distributions with intractable normalizing constants. *Journal of the American Statistical Association* 111(513), 377–393.
- Liang, F., R. Paulo, G. Molina, M. A. Clyde, and J. O. Berger (2008). Mixtures of g priors for Bayesian variable selection. *Journal of the American Statistical Association* 103(481), 410–423.
- Liang, W. W. and H. K. Lee (2011). Bayesian nonstationary Gaussian process models for large datasets via treed process convolutions. Technical report, Tech. rep., UC Santa Cruz, Department of Applied Mathematics and Statistics.
- Liang, W. W. and H. K. Lee (2019). Bayesian nonstationary Gaussian process models via treed process convolutions. *Advances in Data Analysis and Classification* 13(3), 797–818.
- Lindgren, F., H. Rue, and J. Lindström (2011). An explicit link between Gaussian fields and Gaussian Markov random fields: the stochastic partial differential equation approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 73(4), 423–498.
- Liu, Z. (2017). Bayesian model-averaged regularization for Gaussian graphical models. *Communications in Statistics-Simulation and Computation* 46(4), 3213–3223.
- Madigan, D., J. York, and D. Allard (1995). Bayesian graphical models for discrete data. *International Statistical Review/Revue Internationale de Statistique*, 215–232.
- Mairal, J., R. Jenatton, F. R. Bach, and G. R. Obozinski (2010). Network flow algorithms for structured sparsity. In *Advances in Neural Information Processing Systems*, pp. 1558–1566.
- Maruyama, Y., E. I. George, et al. (2011). Fully Bayes factors with a generalized g-prior. *The Annals of Statistics* 39(5), 2740–2765.
- Meng, X.-L. and D. B. Rubin (1993). Maximum likelihood estimation via the ecm algorithm: A general framework. *Biometrika* 80(2), 267–278.

- Mitchell, T. J. and J. J. Beauchamp (1988). Bayesian variable selection in linear regression. *Journal of the American Statistical Association* 83(404), 1023–1032.
- Nychka, D., S. Bandyopadhyay, D. Hammerling, F. Lindgren, and S. Sain (2015). A multiresolution Gaussian process model for the analysis of large spatial datasets. *Journal of Computational and Graphical Statistics* 24(2), 579–599.
- Nychka, D., D. Hammerling, S. Sain, and N. Lenssen (2016). Latticekrig: Multiresolution kriging based on Markov random fields. R package version 7.0.
- Nychka, D., C. Wikle, and J. A. Royle (2002). Multiresolution models for nonstationary spatial covariance functions. *Statistical Modelling* 2(4), 315–331.
- Obozinski, G. and F. Bach (2012). Convex relaxation for combinatorial penalties. *arXiv preprint arXiv:1205.1240*.
- Paciorek, C. J. and M. J. Schervish (2006). Spatial modelling using a new class of nonstationary covariance functions. *Environmetrics* 17(5), 483–506.
- Park, T. and G. Casella (2008). The Bayesian lasso. *Journal of the American Statistical Association* 103(482), 681–686.
- Qian, H. et al. (2017). Big data Bayesian linear regression and variable selection by normal-inverse-gamma summation. *Bayesian Analysis*.
- Raftery, A. E., D. Madigan, and J. A. Hoeting (1997). Bayesian model averaging for linear regression models. *Journal of the American Statistical Association* 92(437), 179–191.
- Ročková, V. and E. I. George (2014). Emvs: The em approach to Bayesian variable selection. *Journal of the American Statistical Association* 109(506), 828–846.
- Ročková, V. and E. I. George (2016). The spike-and-slab lasso. *Journal of the American Statistical Association* (just-accepted).
- Rossell, D. and F. J. Rubio (2018). Tractable Bayesian variable selection: beyond normality. *Journal of the American Statistical Association* 113(524), 1742–1758.
- Rue, H. and L. Held (2005). *Gaussian Markov random fields: theory and applications*. CRC press.
- Sampson, P. D. and P. Guttorp (1992). Nonparametric estimation of nonstationary spatial covariance structure. *Journal of the American Statistical Association* 87(417), 108–119.

- Sanderson, C. and R. Curtin (2016). Armadillo: a template-based c++ library for linear algebra. *Journal of Open Source Software* 1(2), 26.
- Sang, H., M. Jun, and J. Z. Huang (2011). Covariance approximation for large multivariate spatial data sets with an application to multiple climate model errors. *The Annals of Applied Statistics*, 2519–2548.
- Schlather, M., A. Malinowski, P. J. Menck, M. Oesting, and K. Strokorb (2015). Analysis, simulation and prediction of multivariate random fields with package randomfields. *Journal of Statistical Software* 63(8).
- Schlather, M., A. Malinowski, M. Oesting, D. Boecker, K. Strokorb, S. Engelke, J. Martini, F. Ballani, O. Moreva, J. Auel, P. J. Menck, S. Gross, U. Ober, Christoph Berreth, K. Burmeister, J. Manitz, P. Ribeiro, R. Singleton, B. Pfaff, and R Core Team (2017a). *RandomFields: Simulation and Analysis of Random Fields*. R package version 3.1.50.
- Schlather, M., A. Malinowski, M. Oesting, D. Boecker, K. Strokorb, S. Engelke, J. Martini, F. Ballani, O. Moreva, J. Auel, P. J. Menck, S. Gross, U. Ober, Christoph Berreth, K. Burmeister, J. Manitz, P. Ribeiro, R. Singleton, B. Pfaff, and R Core Team (2017b). *RandomFields: Simulation and Analysis of Random Fields*. R package version 3.1.50.
- Schmidt, A. M. and A. O’Hagan (2003). Bayesian inference for non-stationary spatial covariance structure via spatial deformations. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 65(3), 743–758.
- Scott, J. G. and J. O. Berger (2010). Bayes and empirical-Bayes multiplicity adjustment in the variable-selection problem. *The Annals of Statistics*, 2587–2619.
- Shin, M., A. Bhattacharya, and V. E. Johnson (2015). Scalable Bayesian variable selection using nonlocal prior densities in ultrahigh-dimensional settings. Technical report, Texas A&M University.
- Smith, M. and R. Kohn (1996). Nonparametric regression using Bayesian variable selection. *Journal of Econometrics* 75(2), 317–343.
- Stein, M. L. (2007). Spatial variation of total column ozone on a global scale. *The Annals of Applied Statistics*, 191–210.
- Stroud, J. R., M. L. Stein, and S. Lysen (2017). Bayesian and maximum likelihood estimation for Gaussian processes on an incomplete lattice. *Journal of computational and Graphical Statistics* 26(1), 108–120.

- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1), 267–288.
- Tibshirani, R. J., J. Taylor, et al. (2011). The solution path of the generalized lasso. *The Annals of Statistics* 39(3), 1335–1371.
- Tretto, C. (2014). Measuring Bias of Seas Surface Temperature Measurement Devices in the Mediterranean Sea. Master’s thesis, UC Santa Cruz, Santa Cruz, CA.
- Wendland, H. (2004). *Scattered data approximation*, Volume 17. Cambridge university press.
- Yuan, M. and Y. Lin (2006). Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68(1), 49–67.
- Zellner, A. (1986). On assessing prior distributions and Bayesian regression analysis with g-prior distributions. *Bayesian Inference and Decision Techniques*.
- Zellner, A. and A. Siow (1980). Posterior odds ratios for selected regression hypotheses. *Trabajos de estadística y de investigación operativa* 31(1), 585–603.
- Zhao, P., G. Rocha, B. Yu, et al. (2009). The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics* 37(6A), 3468–3497.
- Zhao, P. and B. Yu (2006). On model selection consistency of lasso. *Journal of Machine Learning Research* 7(Nov), 2541–2563.
- Zhou, H. and Y. Wu (2014). A generic path algorithm for regularized statistical estimation. *Journal of the American Statistical Association* 109(506), 686–699.
- Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 67(2), 301–320.