Spatial Regression and Gaussian Process BART

by

Xuetao Lu

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

Approved November 2020 by the
Graduate Supervisory Committee:

Robert McCulloch, Chair
Steven Saul
Paul Hahn
Shiwei Lan
Shuang Zhou

ARIZONA STATE UNIVERSITY

December 2020

ABSTRACT

Spatial regression is one of the central topics in spatial statistics. Based on the goals, interpretation or prediction, spatial regression models can be classified into two categories, linear mixed regression models and nonlinear regression models. This dissertation explored these models and their real world applications. New methods and models were proposed to overcome the challenges in practice. There are three major parts in the dissertation.

In the first part, nonlinear regression models were embedded into a multistage workflow to predict the spatial abundance of reef fish species in the Gulf of Mexico. There were two challenges, zero-inflated data and out of sample prediction. The methods and models in the workflow could effectively handle the zero-inflated sampling data without strong assumptions. Three strategies were proposed to solve the out of sample prediction problem. The results and discussions showed that the nonlinear prediction had the advantages of high accuracy, low bias and well-performed in multi-resolution.

In the second part, a two-stage spatial regression model was proposed for analyzing soil carbon stock (SOC) data. In the first stage, there was a spatial linear mixed model that captured the linear and stationary effects. In the second stage, a generalized additive model was used to explain the nonlinear and nonstationary effects. The results illustrated that the two-stage model had good interpretability in understanding the effect of covariates, meanwhile, it kept high prediction accuracy which is competitive to the popular machine learning models, like, random forest, xgboost and support vector machine.

A new nonlinear regression model, Gaussian process BART (Bayesian additive regression tree), was proposed in the third part. Combining advantages in both BART and Gaussian process, the model could capture the nonlinear effects of both observed and latent covariates. To develop the model, first, the traditional BART was generalized to accommodate correlated errors. Then, the failure of likelihood based Markov chain Monte Carlo (MCMC) in parameter estimating was discussed. Based on the idea of analysis of variation, back comparing and tuning range, were proposed to tackle this failure. Finally, effectiveness of the new model was examined by experiments on both simulation and real data.

ACKNOWLEDGMENTS

The present dissertation was undertaken under the joint supervision of Dr. Robert McCulloch and Dr. Steven Saul. I would like to express my deepest gratitude and appreciation to them. During my research, Robert always trusted me with his many great ideas and, at the same time, encouraged me to confidently further develop and explore them in my own ways, always while offering invaluable advice. Steven is my co-advisor as well as a good friend. Four years ago he introduced me into spatial statistics which was a total new area for me. Without his inspirational guidance, constant support, and patient encouragement this dissertation would not be completed. Overall, they have inspired me both as an academic and as a person and, therefore, I feel very privileged to have worked under their supervision.

I am very grateful to my mentors, Dr. Julie Bessac and Dr. Umakant Mishra, in Argonne National Laboratory. My research presented in Chapter 3 was accomplished under their supervision when I took an internship at Argonne this summer. They all made this an unforgettable experience that has evolved in an ongoing collaboration and friendship.

I also would like to extend my appreciation to my committee members, Dr. Paul Hahn, Dr. Shiwei Lan, and Dr. Shuang Zhou for their helpful support on my dissertation. Their efforts make this dissertation better.

I am grateful to the funding bodies that enabled me to pursue my studies: Gulf of Mexico Research Initiative (GoMRI), National Oceanic Atmospheric Administrations (NOAA) National Marine Fisheries Service (NMFS) through the University of Miamis Cooperative Institute for Marine and Atmospheric Studies (CIMAS), and National Science Foundation.

Finally my deepest thanks go to my parents in law, Wenhua and Wenjiang, for helping us take care of my daughter. And to my parents, Junjiang and Hongxia, my sister, Chao, for sending their love and support from thousands miles away. A special thanks to my wife Dr. Yuxia Shen for her endless support, love and patience. And to my daughter Julie for reminding me that happiness is in the simple things.

This dissertation is dedicated to my wife, my daughter, Yuxia and Julie, my love for the rest of my life.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Chapter 1

INTRODUCTION

Spatial statistics is a branch of statistics that be developed specifically for geographic data. These data are prevalent in many scientific disciplines such as meteorology, oceanography, soil science, agriculture, geology, natural resources, epidemiology, etc. With the use of spatial statistics becoming more popular across different disciplines, it is currently one of the most active research areas in statistics. Gelfand *et al.* (2010) viewed spatial statistics as being comprised of three major categories: continuous spatial variation, discrete spatial variation and spatial point patterns. Continuous spatial variation that focus on the study of continuous spatial processes includes the topics, like, geostatistical modeling and inference, likelihood-based approaches, spectral methods, hierarchical modeling, spatial design, etc. Spatial regression stays at the center of this category and connects with all other topics.

1.1 Spatial Regression

Regression is a technique used to examine the relation of a dependent variable to specified covariates. When the data has a spatial component, the regression model has to recognise and adapt to this change. In this case, we call it spatial regression model. A general form of the spatial regression model that we studied in the dissertation is as follows:

$$y(s) = f(s; X(s)) + w(s) + \epsilon(s) \tag{1.1}$$

where $s := \{s_1, ...s_n\}$ is the set of spatial locations; $y(s)$ is the observed dependent variable at s; $X(s)$ are the observed covariates at s; $f(\cdot)$ is an arbitrary function;

$w(s)$ is a stochastic process; $\epsilon(s)$ are the i.i.d. errors.

According the general form (1.1), spatial regression models can be divided into two categories.

(1) Spatial Linear Mixed Regression Model

In this case, $w(s) \neq 0$, the effect of unobserved covariates is exhibited as spatial dependence that be modeled by a stochastic process $w(s)$. The function $f(\cdot)$ that models the effect of observed covariates has a linear form. The classical Kriging models (Cressie, 1993) which focus on estimating the first-order (large-scale or global trend) and second-order (small-scale or local) structure of $y(s)$ falls in this category. For example, at any new spatial location s_0, the stochastic term in Kriging models is $w(s_0) = \sum\limits_{i=1}^{n} \lambda(s_i)y(s_i)$. Then,

- if $f(s_0; X(s_0)) = 0$, (1.1) is a simple kriging model.
- if $f(s_0; X(s_0)) = \alpha_0 + \alpha_1 s_{0x} + \alpha_2 s_{0y}$, (1.1) is an universal kriging model.
- if $f(s_0; X(s_0)) = X(s_0)\beta$, (1.1) is a regression kriging model.

Spatial linear mixed regression model has a long history in spatial statistics (Cressie, 1993). With the advantages of solid theoretical foundation, simple mathematicial formula and good interpretability, they are widely applied in different disciplines, such as geography (Haining *et al.*, 2010), ecology (Robertson, 1987), meteorology (Dobesch *et al.*, 2007), etc.

(2) Spatial Nonlinear Regression Model

In this case, $w(s) = 0$, it means that only the effect of observed covariates be considered in the model. The spatial dependence effect is modeled by the function $f(\cdot)$. Since the linear regression model is trival, we are interested in the nonlinear ones, e.g. the popular machine learning models, like, ensemble models (Random

Forest, XGBoost), kernel based models (Support Vector Machine), Neural Networks, etc. As the rising of machine learning, their application in spatial analysis grows rapidly, especially in the direction of deriving spatial predictions for spatial regression (Appelhans *et al.*, 2015) (Li *et al.*, 2011) and detecting spatial patterns (Williamson *et al.*, 2020).

Similar to the ordinary statistical regression, there are two major goals in spatial regression, prediction and interpretation. Figure 1.1 illustrates the relative positions of the two categories models in the coordinate system of prediction and interpretation. In real applications, if our goal is to get a good prediction then spatial nonlinear regression models are good choices. While, if understanding the relationships between Y and X is a top priority, we prefer the spatial linear Mixed regression models which are much easier to be explained than the nonlinear ones. However, there is a trade-off between the prediction and interpretation. We will discuss it in chapter 3.

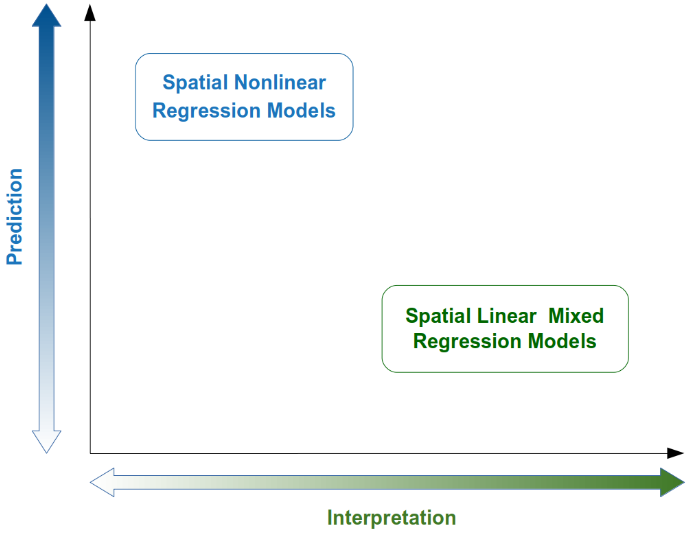

Spatial Nonlinear Regression Models
Prediction
Spatial Linear Mixed Regression Models
Interpretation

Figure 1.1: Prediction Vs Interpretation

1.2 Gaussian Process

In spatial process regression model (1.1), the term $w(s)$ usually is a Gaussian process. Gaussian process is as well known as the extension of multivariate Gaussian to infinite-sized collections of real-valued variables. This extension can be used to infer the distribution over functions. First, using Gaussian process defines a prior over functions. Then, convert it into a Gaussian process posterior after obtaining some data.

Suppose we choose a particular finite subset of these random function variables $\boldsymbol{f} = \{f_1, ..., f_N\}$ and the data $\{\boldsymbol{Y} = \{y_1, ..., y_N\}, \boldsymbol{X^Y} = \{\boldsymbol{X^Y}_1, ..., \boldsymbol{X^Y}_N\}\}$ as the prior distribution. By the property of Gaussian Process, f follows a multivariate Gaussian distribution:

$$p(\boldsymbol{f}|\boldsymbol{Y}, \boldsymbol{X^Y}) \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{K^Y})$$

where $\boldsymbol{K^Y}_{ij} = C(\boldsymbol{X^Y}_i, \boldsymbol{X^Y}_j)$, $C(\cdot, \cdot)$ is a covariance function. In spatial regression models, we always assume the mean of Gaussian process is zero.

If some new data $\{\boldsymbol{Z} = \{z_1, ..., z_M\}, \boldsymbol{X^Z} = \{\boldsymbol{X^Z}_1, ..., \boldsymbol{X^Z}_M\}\}$ be observed, we can get the posterior distribution by the property of conditional multivariate Gaussian distribution:

$$p(\boldsymbol{f}|\boldsymbol{Z}, \boldsymbol{X^Z}) \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$$

where $\boldsymbol{\mu} = \boldsymbol{K^{ZY}}(\boldsymbol{K^Y})^{-1}\boldsymbol{Y}$ and $\boldsymbol{\Sigma} = \boldsymbol{K^Z} - \boldsymbol{K^{ZY}}(\boldsymbol{K^Y})^{-1}\boldsymbol{K^{YZ}}$. And $\boldsymbol{K^{ZY}} = C(\boldsymbol{X^Z}, \boldsymbol{X^Y}) = (\boldsymbol{K^{YZ}})^T$ is $M \times N$ and $\boldsymbol{K^Z} = C(\boldsymbol{X^Z}, \boldsymbol{X^Z})$ is $M \times M$.

Figure 1.2 (a) shows 5 samples from a Guassian process prior distribution, while (b) illustrates 5 samples from its posterior distribution after obtaining 8 new observations.

The zero mean Gaussian process can be denoted by $GP(0, C(\cdot, \cdot|\boldsymbol{\theta}))$,where $\boldsymbol{\theta}$ is the parameters of covariance function. It is completely determined by its covariance

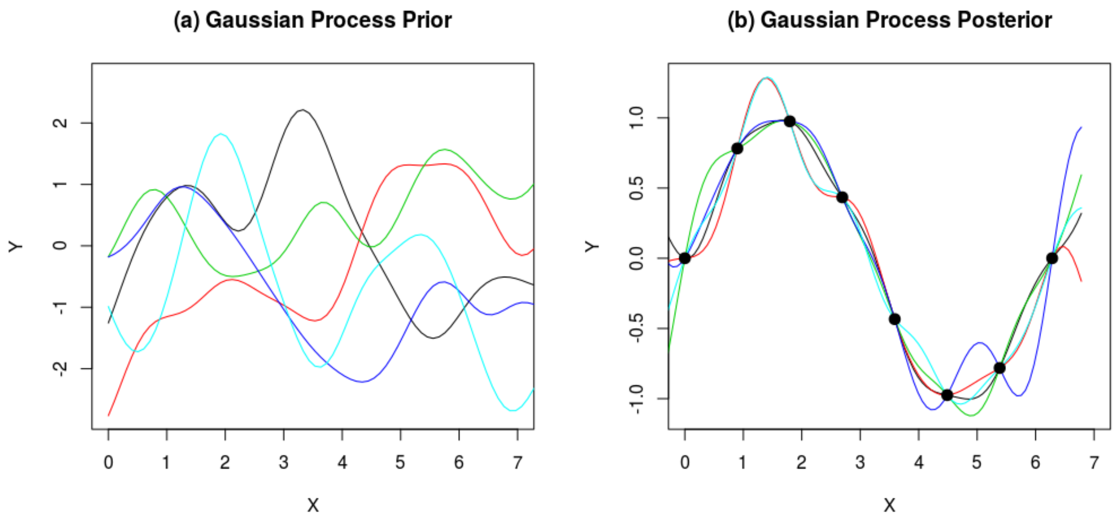


Figure 1.2: Sampling from Gaussian Process Prior and Posterior Distributions

function $C(\cdot, \cdot|\boldsymbol{\theta})$. In order to model the spatial dependence, we assume the covariance function following some spatial correlation structure. For example, a low dimensional parametric correlation structure can be specified by Matérn covariance function family (Stein, 1999) as following.

$$C(s_i, s_j) = C_\nu(||s_i - s_j||) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{||s_i - s_j||}{\rho} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{||s_i - s_j||}{\rho} \right)$$

where $\Gamma(\cdot)$ is the gamma function; $K_\nu(\cdot)$ is the modified Bessel function of the second kind; $||s_i - s_j||$ is the Euclidean distance between spatial point s_i and s_j; The set of parameters is $\boldsymbol{\theta} = \{\sigma^2, \rho, \nu\}$ and ρ, ν are positive real numbers. ρ and ν control the decay rate and smoothness in spatial correlation respectively. There are two popular candidates from Matérn covariance function family in spatial statistics.

- $\nu = \frac{1}{2}$, $C_\nu(||s_i - s_j||) = \sigma^2 exp\{-\frac{||s_i - s_j||}{\rho}\}$, it's an exponential covariance function.

- $\nu \rightarrow \infty$, $C_\nu(||s_i - s_j||) = \sigma^2 exp\{-\frac{||s_i - s_j||^2}{2\rho^2}\}$, it's a gaussian covariance function which is also called squared-exponential covariance function.

With advantages mentioned above, Gaussian process is prevalent in spatial statistical modeling today. However, the computational issue will rise when data becomes big. It's because the likelihood computation of a Gaussian process observed at n spatial locations has to calculate the inverse and determinant of covariance matrix whose exact calculation requires $O(n^3)$ operations and $O(n^2)$ storage. In recent year, due to the advance in technology, massive spatial data are collected in various disciplines, we do require novel methods to overcome this challenge. Fortunately, there is a rich literature on this problem. Basically, the studies are going on two tracks, low-rank and sparsity. Low-rank approximation is a very active field in numerical linear algebra. Hackbusch (2015) developed the theory of hierarchical matrices which can provide a low-rank approximation requiring only $O(nklog(n))$ units of storage and $O(nk^{\alpha}log(n)^{\beta})$ operations for matrix multiplication, inversion or determinant, where k is the rank parameter controlling the accuracy of the approximation, $\alpha, \beta \in \{1, 2, 3\}$. Geoga *et al.* (2019) presented a kernel-independent method that applies hierarchical matrices to the problem of maximum likelihood estimation for Gaussian processes. There were also low-rank modeling methods in spatial statistics community, e.g. Fixed Rank Kriging (Cressie and Johannesson, 2008), predictive process model (Banerjee *et al.*, 2008) and stochastic partial differential equations (Lindgren *et al.*, 2011). On the other track, studies seek to introduce sparseness into the covariance or precision matrix. For example, Furrer *et al.* (2006) applied covariance tapering to create a sparse approximate linear system that can then be solved using sparse matrix algorithms, or Datta *et al.* (2016) extends the Vecchias approximation (Vecchia, 1988) to a Gaussian process for creating the sparse precision matrices by using conditional independence given information from neighboring locations.

1.3 Overview

Figure 1.3 shows the two categories of spatial regression models and my study Roadmap. My research involves both application and methodology problems. They are organized in the rest of the dissertation as follows.

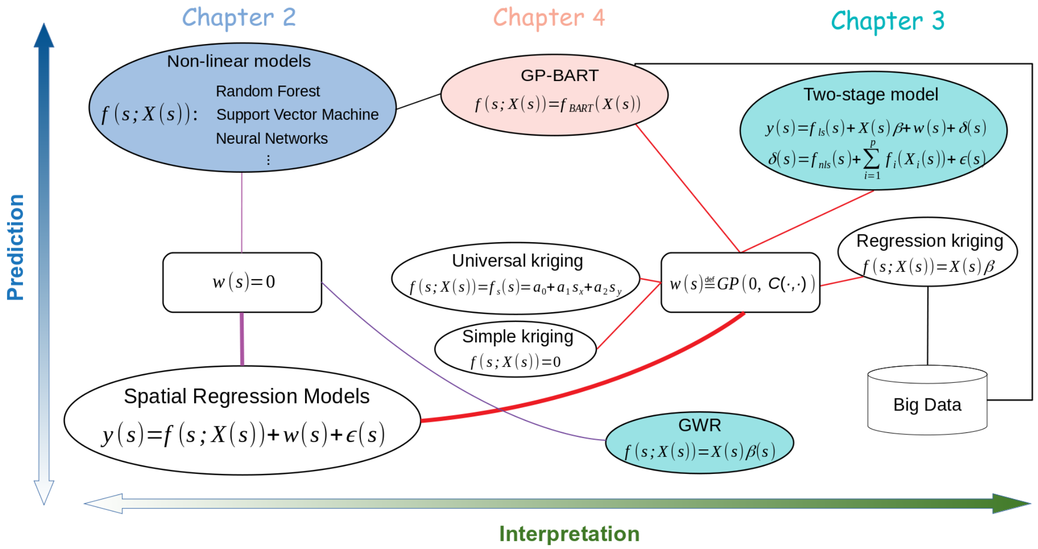


Figure 1.3: Study Roadmap

- Chapter 2, introduces a real world application that applied the nonlinear regression models to predict the spatial distributions of reef fish abundance in the Gulf of Mexico. A multistage workflow is proposed to overcome the challenges of zero-inflated data and out of sample prediction. The nonlinear predictions are compared to the predictions of other methods.

- Chapter 3, aims to develop a spatial regression model that can break the trade-off between prediction and interpretation. A two-stage universal regression Kriging and generalized additive model is built to achieve this goal. Model's interpretability and prediction accuracy are tested on both the real and simulation data.

- Chapter 4, proposes a new BART model and combines it with the Gaussian process to build a nonlinear spatial regression model, named Gaussian process BART. The methods for parameter estimation is discussed. Two experiments and a real data testing are given for examining the effectiveness of Gaussian process BART model.

- Chapter 5, summarizes the key contributions of the dissertation and discusses ideas for future work.

Chapter 2

REAL DATA CHALLENGES AND NONLINEAR MODELS

Understanding the spatial distribution of abundance is fundamental to assessing and managing organism population. However, the task becomes difficult to marine species due to the low detection rates because of sampling underwater. In this chapter, we proposed a multistage statistical workflow which applied the non-linear spatial regression models to tackle this problem. In Section 2.1, we introduced the problem, data and challenges. We proposed a multistage statistical workflow to overcome the challenges in Section 2.2. The methods to solve the zero-inflated data problem were discussed in this section as well. In Section 2.3, we applied nonlinear spatial regression models to predict the spatial distributions of reef fish abundance. Three strategies were proposed to handle the out of sample problem. In Section 2.4, the nonlinear prediction results were compared to the ones of linear regression and catch per unit effort models.

2.1 Introduction

The ability to map the abundance of organisms across space is a critical precursor for many applied research applications that support sustainable environmental resources management. This includes understanding linkages between species and habitat use (Mateo-Sánchez *et al.*, 2015), establishing protected areas (Lin *et al.*, 2017), building population and ecosystem models (Stratford *et al.*, 2016), using such ecological analyses to develop biologically and economically sustainable management policies (Guisan *et al.*, 2013), etc. However, in most cases, the real data needed to develop such maps either does not exist or is zero-inflated or unevenly distributed across space (Prosser *et al.*, 2018). This is because field sampling efforts can be expensive or often must collect information for multiple applications. For example, collecting data from the marine environment can be particularly difficult due to, cost and logistical considerations accessing remote locations, the inability to transmit radio or satellite signals from underwater, and visual limitations associated with water clarity and water column light attenuation. It results in fewer observations of species from the marine environment with which to develop distribution maps.

Independent of the ecosystem, a variety of techniques have been employed over the years to maximize the use of the zero-inflated sampling data, such as variogram estimation and random field simulation (Saul and Purkis, 2015), generalized additive models (Drexler and Ainsworth, 2013), additive beta regression (Ros-Pena *et al.*, 2018), etc. However, most of them assume samples evenly distributed in the study region or their values follow specified distributions, conditional normal distribution or beta distribution. In practice, these assumptions may be violated, because most organisms usually distribute in patchy pattern across the landscape or seascape (Ainsworth *et al.*, 2016). The underlying reason is that organism abundance directly depends on their habitat environment and most of environmental covariates follow patchy spatial patterns. Although, for some organisms, approaches work on commercial data, like catch per unit effort data (McDonald *et al.*, 2001), can extract useful maps of abundance, models with environmental covariates are more promising credit to their high interpretability and low bias (Streich *et al.*, 2017). Many methods to map organism spatial abundance use the relationship between abundance and environmental covariates. Models can be divided into two categories, linear models and nonlinear models. Linear models had been extensively studied in predicting organism abundance (Guisan *et al.*, 2002). They have many advantages, such as, model is easy to construct, parameters and results are highly interpretable, computation is efficient in both time and resources, etc. Linear models perform well on large spatial scales identifying the overall trend or gradient of abundance (Guisan *et al.*, 2002). However, most relationships in real world are intrinsically nonlinear rather than linear in nature.In the few years, as therising of machine learning, nonlinear models have been widely applied in organism population prediction (Ye *et al.*, 2019). They are good at capturing the subtle nonlinear relations between abundance and environmental covariates and the complex interactions among covariates. So, nonlinear models also have the ability to identify the local trend in fine spatial scale. Compare to linear models, nonlinear ones could normally provide more accurate predictions at the cost of sacrificing interpretability.

2.1.1 Data

The study region was the area whose depth is within 1 200 meters in north Gulf of Mexico (Figure 2.1). Two types of datasets, video surveys data and interpolated bottom habitat data, were used in the study.

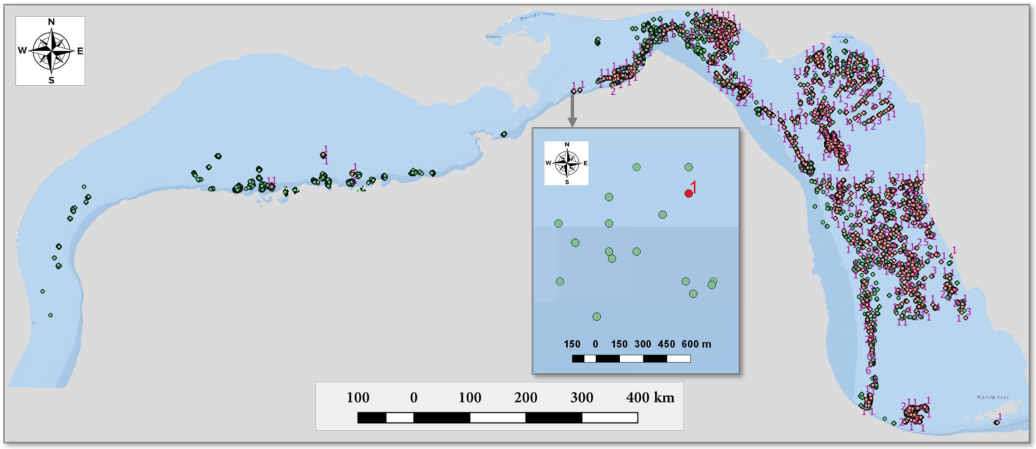


Figure 2.1: Study region and video survey data. The zoomed area shows zero-inflated feature of the sampling data. The red dot represents positive sampling, the value on it numbers the fishes been sampled.

Three independent fishery video surveys are carried out annually to collect information on the abundance of shallow water reef fish species throughout the Gulf of Mexico. The first one is sponsored by the National Marine Fisheries Service (NMFS) Panama City Florida laboratory, the second is part of the Southeast Area Monitoring and Assessment Program (SEAMAP), and the third is sponsored by the State of Florida Fish and Wildlife Commission (FWC). Each survey is methodologically standardized to others. It allows us to merge them to a single dataset with trivial effort. The video surveys target the commercially important species, such as red grouper (Epinephelus morio), red snapper (Lutjanus campechanus), gag grouper (Mycteroperca microlepis), mutton snapper (Lutjanus analis), etc. Video surveys were carried out by following a two-stage sampling design. The first-stage or primary sampling units (PSUs) which located in the most possible habitat area were spatial blocks with 10 minutes of latitude by 10 minutes of longitude. The second stage or ultimate sampling sites (USS) were point locations that were randomly choose in the ultimate sampling sites (USS). The sampling gear consisted of four cameras mounted orthogonally with each other. Cameras was deployed at each location for 20 minutes and record every species encountered. The camera sampling protocol included the use of bait at the center of the four-camera array to increase the positive detection rate. The video footage was read by several technicians to identified and enumerated the species observed. In order to avoid double counting, the count value was set as the maximum number of the species recorded in a frame during the 20-minute sampling period (Somerton and Glendhill, 2005). The sampling design wasnt optimal due to budget constraint. For example, the short sampling time may be the prime reason for the zero-inflated sampling result (Figure 2.1). Another defect is that the sampling sites were not evenly distributed across the study region. In the prediction stage, it will cause the out of sample prediction problem for the blank areas (Figure 2.1).

Bottom habitat information was offered by dbSEABED database. dbSEABED project produced detailed mappings of the sea floor in various locations by interpolating from all available point datasets. Individual raw data points were screened for quality control before being used for interpolation. Isotropic, binned semivariograms were used to interpolate point data to raster map (Goff *et al.*, 2008). Maps can be generated respectively to describe single benthic environmental variable. In this study, the environmental covariate maps (percentage content) to be used for prediction includes sand, gravel, mud, sediment grain size, carbonate, clay, and rock. One defect of dbSEABED database was that benthic samples collected over the years were more concentrated in nearshore areas than offshore ones. It makes the data has lower variation thus higher accuracy in nearshore area than in offshore area. Despite this, the dbSEABED dataset is the most spatially comprehensive habitat data publicly available for the Gulf of Mexico at this time.

2.2 Multistage Workflow

The multistage workflow (Figure 2.2) start from simulating the video survey process to generate simulated sampling outcomes under different settings. In the second stage, a method named empirical maximum likelihood analysis worked with simulated sampling data to find a relationship between the video survey data (catch ratio) and fish abundance (empirical maximum likelihood density). The relation represented by an empirical maximum likelihood density function which was the key to address the zero-inflated issue. Then, inputting with real video survey data and the empirical maximum likelihood density function, a two-step random smoothing method was employed in the third stage. In step one, spatial abundance was estimated in sampling areas. In step two, uncertainties in abundance estimation were effectively removed to produce the block spatial abundance that will work as training data in next stage models. In the final stage, working on the training data and environmental covariates (habitat data), nonlinear spatial regression (Machine learning) models coming from 3 different families, support vector machine, neural networks and random forest, were assembled to generate a high accuracy and low bias prediction of abundance spatial distribution.

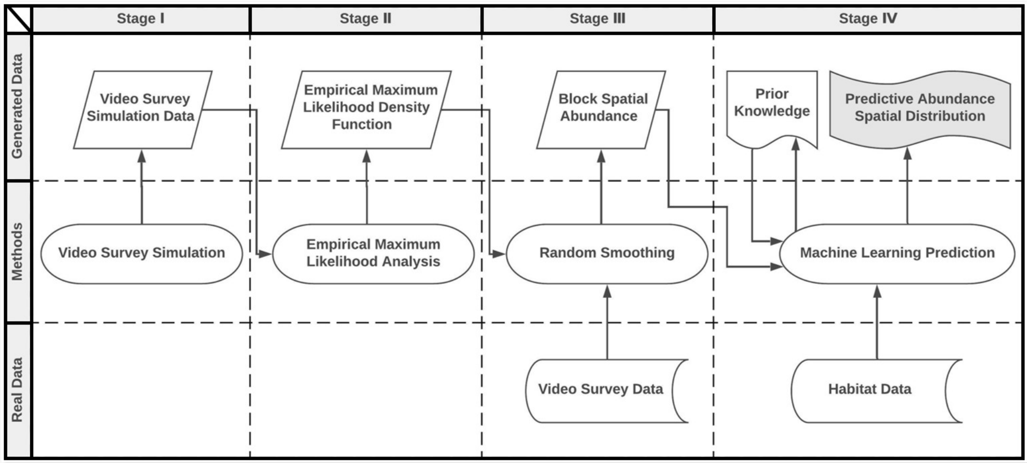

Stage I
Stage II
Stage III
Stage IV
Generated Data
Methods
Real Data
Video Survey Simulation Data
Empirical Maximum Likelihood Density Function
Block Spatial Abundance
Prior Knowledge
Predictive Abundance Spatial Distribution
Video Survey Simulation
Empirical Maximum Likelihood Analysis
Random Smoothing
Machine Learning Prediction
Video Survey Data
Habitat Data

Figure 2.2: There are four stages with different methods/models and data in each of them. Generated data means that the data was generated by the model or prior knowledge. In contrast, real data is collected from real world.

In the rest of this section, we will go through the StageI to Stage III. The methods/models in these stages work together to tackle the zero-inflated problem of the video survey data.

2.2.1 Video Survey Simulation

To make the most use of the video survey data, an individual-based discrete event simulation was developed to model the video survey process.(Pfeffermann, 2013) This was developed using the PyGame library in the Python programming language (Kelly, 2016). The following assumptions were made for the simulation:

(1) Site fidelity

Red grouper excavates benthic material to create nests or pits in which they live, and from which they exhibit high site fidelity (Harter *et al.*, 2017). Red snapper exhibits less site fidelity than red grouper but spends continuous periods of time at one site, on the order of months or years, before moving to another habitat location. Thus, at short time intervals, such as the length of the camera sampling protocol, individuals were assumed to have strong site fidelity in the simulation.

(2) Fish home and behavior

In order to model site fidelity, we assumed the fish move around nearby its home. Fish was able to explore surrounding places by wandering in a random fashion. In the simulation, wandering implemented by a Markov Chain Monte Carlo. Its random fashion followed an isotropic bi-normal spatial distribution around the home (Figure 2.3). It was meant to represent activities such as food foraging similar to central place foraging theory (Schoener, 1987). Fish had a 68% probability to move within one standard deviation of the isotropic bi-normal spatial distribution, and 95% probability of moving within two standard deviations. Home range was defined as the spatial distance of two standard deviations. For two neighboring homes, the average percentage of overlapping is less than 50% (Farmer and Ault, 2011). The setting of parameters, such as, home range, fishs movement frequency, speed, turning angle, followed Farmers papers (Farmer and Ault, 2011),(Farmer and Ault, 2014) which conducted a thorough investigation to the movement of reef fish species in the Gulf of Mexico.

(3) Camera and bait

In video survey, the vulnerability of fish to be sampled by the camera gear was enhanced by placing bait at the location of the camera array to attract fish. We modeled the bait effect on video sampling. The chance that a fish would detect the bait and come to it is determined by two factors: the diffusion rate of the scent of the bait, and the probability that a fish in the vicinity of bait detectability, could detect the bait (Stoner, 2004). We assumed that the bait odor had highest intensity at the

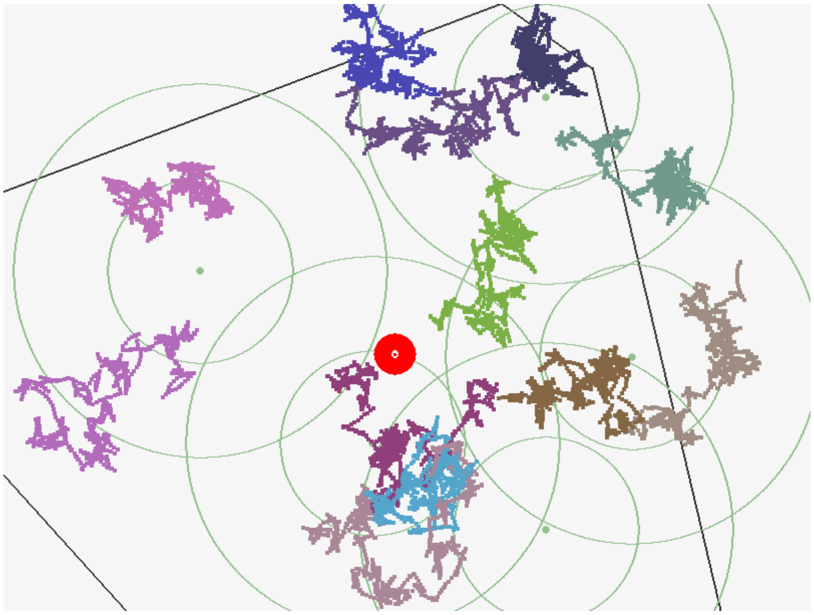


Figure 2.3: Video survey simulation. The green dots represent the location of a fish home, green concentric circles around each dot represents the first and second standard deviations, the red dot represents the location of the camera, and the red circle represents the distance to bait detectability, which expands throughout 20 minutes, the winding trails represents the trace of fishes, each color corresponding one fish.

camera sampling gear, and spread with intensity diminishing exponentially as moving away from the camera. The attenuation of bait odor through the water column is an understudied complex process, as is the probability that a fish nearby will detect it. The few studies that have been done suggest a wide range of distances from which fish can detect bait, and the research suggests it is species dependent (Sainte-Marie and Hargrave, 1987). As a result, we made two assumptions: (a) that the radius of detectability from the camera array, meaning the maximum distance of odor spread was 50 meters in 20 mins, and (b) that the value for the shape parameter of the exponential bait detectability distribution was 0.05.

(4) Interactions

In each simulation step, program checks the location of fish in relation to the location of the bait and the range of bait odor dissipation. If the fish entered the range of bait odor, then it was assigned a probability of being sampled by the camera.

Once a fish was sampled by the camera, it was removed from the simulation to avoid double counting.

The most important parameter in simulation model was the number of fish homes. Since simulation area was invariant and the number of fishes in each home followed a known uniform discrete distribution, the number of fish homes scaled the abundance of fish at each sampling station. We tested a range of numbers fish homes in the simulation. It was increased by three and up to a big enough number which was constrained by the rule of less than 50% habitat overlapping and the size of simulation area. For each number, the simulation run 5000 times. One simulation step corresponds to 3 seconds in real time, and camera works 20 minutes. The data generated by the video survey simulation will be used in the empirical maximum analysis.

2.2.2 Empirical Maximum Likelihood Analysis

In statistics, traditional maximum likelihood analysis produces the maximum likelihood estimator (MLE) for unknown parameter. This method typically includes three components: an analytic mathematic model, the target parameter and the observed outcomes. Although our workflow contained a model (video simulation model), the target parameters (fish density) and the observed outcomes (real video survey data), we are unable to apply traditional maximum likelihood methodology because the video survey model is a programing simulation model rather than an analytic mathematic model. As a result, we developed a novel method called empirical maximum likelihood analysis to tackle this issue. The method includes four parts: re-sampling, empirical probability mass functions, empirical likelihood function, and empirical maximum likelihood density function.

The procedures of re-sampling and creating empirical probability mass functions is as following.

(1) Initialize the number of the fish homes as n=3.

(2) Sample 100 outcomes without replacement from the results of video survey simulation. Calculate the ratio between number of detected fishes and the number of fish homes. We mane this ratio as catch ratio (CR).

(3) Calculate empirical probability mass function (pmf) of discretized catch ratio values for each value of home numbers.

(4) Increase home number by n=n+3 and repeat steps (2) and (3).

Once the probability mass functions were obtained, we can calculate the empirical likelihood function for each catch ratio. Figure 2.4 gives an example that how to build an empirical likelihood function from probability mass functions. Then, the empirical maximum likelihood estimator of home number under each catch ratio will be equal to the globe maximum of empirical likelihood function. Since the number of fishes in each home followed a uniform discrete distribution, the maximum likelihood estimator of fish homes can be easily converted to maximum likelihood estimator of density by the ratio of maximum likelihood estimator of fishes and the size of area. This allowed us to build a function between the empirical maximum likelihood density and catch ratio (Figure 2.5). This function handles the spatial sparsity and zero-inflated characteristics of video survey data. Even if the catch ratio is close to zero, a non-zero maximum likelihood density could be calculated.

There is a linear relation between the empirical maximum likelihood density and catch ratio. The changing of parameters value, like, the parameters value of camera, bait and fish behavior, in video simulation model only affects the coefficient (slope) of this linear relation. However, all the coefficient will be cancelled when we transform the absolute value of abundance to the relative value of abundance - the spatial

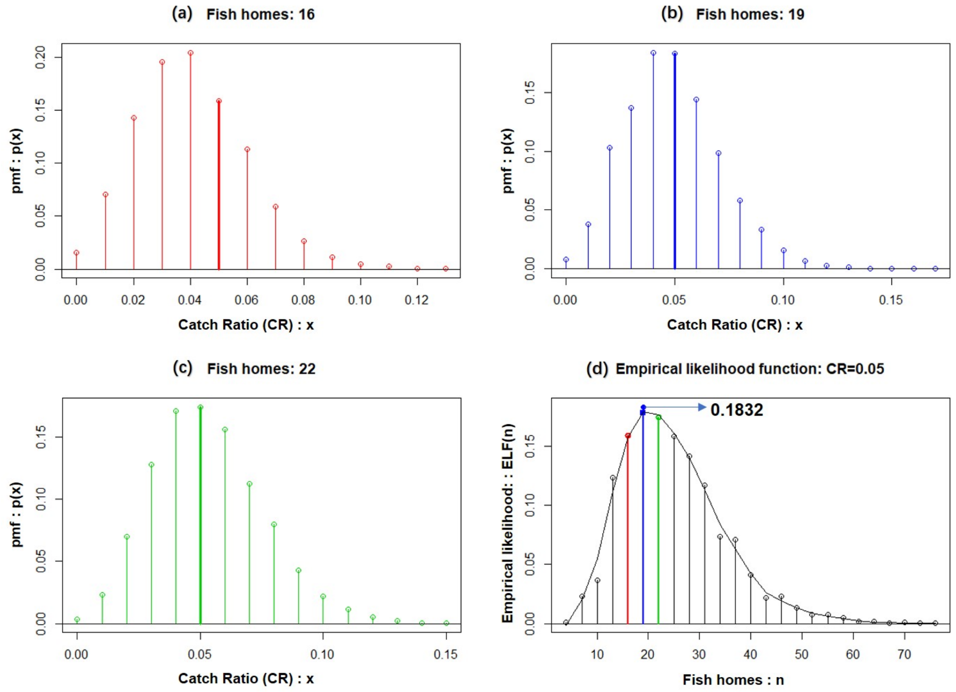


Figure 2.4: For each home number, there is a corresponding probability mass function for discrete catch ratios. (a), (b), and (c) are 3 examples of pmf. Given a value of catch ratio, the discrete empirical likelihood function can be obtained (d). Since the gap of home number was 3, we can fit a curve (d) to get the discrete empirical likelihood function with gap one. Finally, the empirical maximum likelihood estimator can be found from this discrete empirical likelihood function. In this example, when the catch ratio was 0.05, the empirical MLE of home number was 19.

distribution. For purposes of this study, we were only interested in the abundance spatial distribution. So, we didnt really care about the setting of parameters in video simulation model because they did influence the linear coefficients rather than the final spatial distribution. But when you are interest in estimating the real abundance, the parameter setting is essential.

2.2.3 Random Smoothing Estimation

The spatial abundance or empirical maximum likelihood density was estimated by random smoothing (Figure 2.6). First, the sampling area needs to be rasterized into grid cells, each approximately 0.25 square kilometers. Then, random smoothing

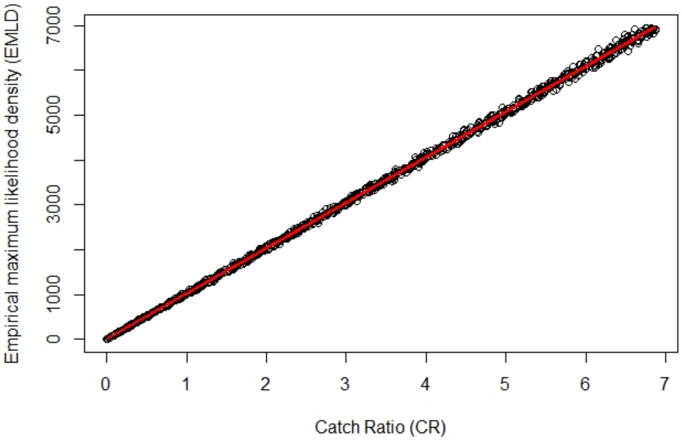


Figure 2.5: Empirical Maximum Likelihood Density Function

was carried out by randomly drawing circle windows in the area (Figure 2.6). In each smoothing window, the catch ratio was calculated from the video survey data. Working with empirical maximum likelihood density (EMLD) function, we can assign the empirical maximum likelihood density (abundance) to all the grid cells in the smoothing window.

The smoothing windows may overlap with each other. Therefore, a grid cell could be covered by different windows. Hence different empirical maximum likelihood density may be assigned to the same grid cell. In order to combine all the different values of a grid cell, a weighted mean empirical maximum likelihood density was taken. Weights were determined by calculating a credibility statistic for each window. The credibility was defined as follows:

$$c(x) = \frac{x}{N} \tag{2.1}$$

where $c(x)$ is the credibility; x is the number of samples in the smoothing window; N is sample size.

In order to penalize the windows with low credibility, we calculate the weight of each window:

$$w_i = \frac{c_i^2}{\sum_{i=1}^{n} c_j^2}, \qquad i = 1, 2, ..., n \tag{2.2}$$

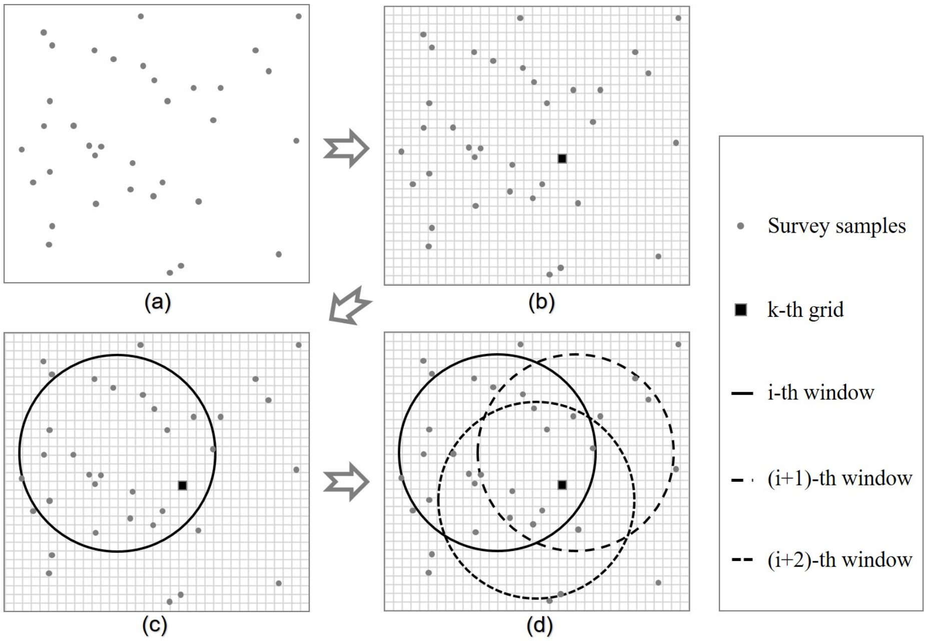


Figure 2.6: The procedure of random smoothing estimation starts from rasterizing, (a) to (b). Then, randomly draw windows in the sampling area up to a large enough number, (c) to (d). This number can be determined by checking the convergency of gemld.

where w_i is the weight of i^{th} window; c_i is the credibility of i^{th} window; n is the number of random smoothing windows.

Thus, the weighted mean empirical maximum likelihood density of a grid cell can be denoted as follows.

$$gemld_k = \sum_{i=1}^{n} w_{ik} * wemld_i, \qquad i = 1, 2, ..., N \tag{2.3}$$

where $gemld_k$ is the weighted mean empirical maximum likelihood density of k-th grid; w_k is the weight of i^{th} random smoothing window covering k^{th} grid; $wemld_i$ is the empirical maximum likelihood density of i^{th} random smoothing window; n is the number of random smoothing windows covering the k^{th} grid; N is the number of grid cells.

2.2.4 Reducing Uncertainty

In random smoothing, uncertainties were introduced by randomized smoothing windows. We consider two concepts to identify uncertainties: credibility and variance. Credibility measured uncertainty from a Bayesian perspective, while variance measured uncertainty from a frequentist perspective.

For each grid cell, we can calculate its credibility mean as follows.

$$gmc_k = \frac{1}{n}\sum_{i=1}^{n} gc_{ik}, \qquad k = 1, 2, ..., N \tag{2.4}$$

where gmc_k is the mean of credibility of k^{th} grid; gc_{ik} is the credibility of i^{th} window covering k^{th} grid; n is the number of random smoothing windows covering k^{th} grid; N is the number of grid cells.

The variance can be calculated:

$$gv_k = var(S_k), \qquad k = 1, 2, ..., N \tag{2.5}$$

where gv_k is the variance of empirical maximum likelihood density of k^{th} grid; $S_k = \{gemld_1, ..., gemld_n\}$; n is the number of random smoothing windows covering k^{th} grid; N is the number of grid cells.

Based on above definitions, we developed a method hereafter referred to as Bayesian and Frequentist scissors to eliminate uncertainties by using a priori determined threshold.

$$Scissors_B = quantile_{gmc}\{gmc_i, \quad i = 1, ..., N\} \tag{2.6}$$

$$Scissors_F = quantile_{gv}\{gv_i, \quad i = 1, ..., N\} \tag{2.7}$$

where N is the number of grid cells.

The scissors worked as follows.

$$G_{BS} = \{gmc_k : gmc_k \geq Scissors_B \quad k = 1, ..., N\} \tag{2.8}$$

$$G_{FS} = \{gv_k : gv_k \leq Scissors_F \quad k = 1, ..., N\} \tag{2.9}$$

where G_{BS} is the set of grid cells after Bayesian scissors cutting; G_{FS} is the set of grid cells after Frequentist scissors cutting; N is the number of grid cells.

The final set of grid cells, G_L, is obtained by the intersection of G_{BS} and G_{FS}.

$$G_L = G_{BS} \cap G_{FS} \tag{2.10}$$

G_L is the block spatial abundance with low uncertainty. Figure 2.7 shows the process that how to apply random smoothing and Bayesian/frequentist scissors to get the block spatial abundance.

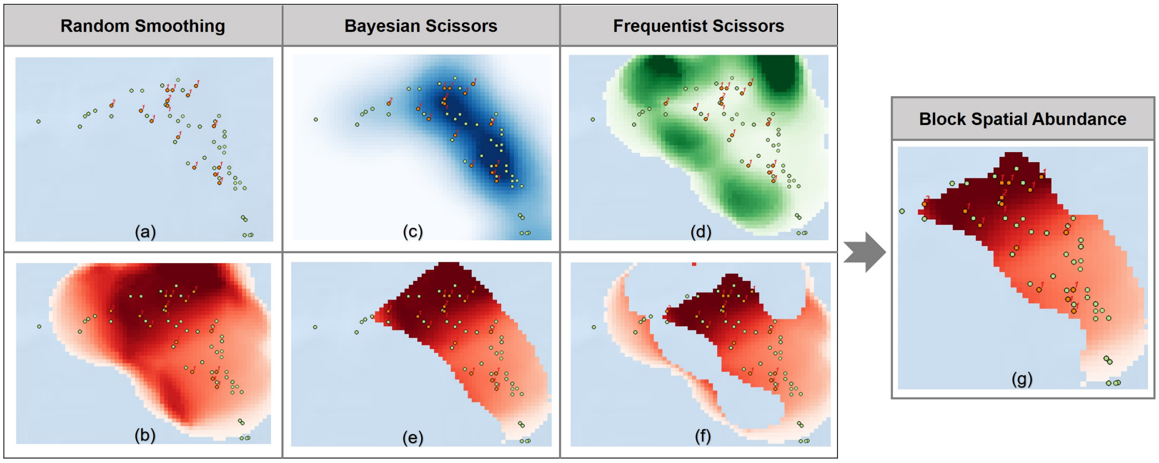


Figure 2.7: An example applying random smoothing and Bayesian/frequentist scissors to get a low uncertainty set of label data. Panel (a) shows the video survey data of red grouper in a small region. The red dots with small numbers identify the number of red groupers were found in the survey. The result of the random smoothing is shown in panel (b). The spatial mean of credibility and spatial sample variance are shown in panels (c) and (d) respectively. Panels (e), (f) and (g) show the results of G_{BS}, G_{FS} and G_L respectively.

Up to now we successfully convert the zero-inflated video survey data to high credibility low variance block spatial abundance data. In next section, the later will be used as training data of the spatial nonlinear regression models.

2.3 Non-linear Models and out of Sample Prediction

As discussed in Section 2.1, our final goal is to get good prediction of abundance spatial distribution. Nonlinear spatial regression (machine learning) models are good choices for this task. With the ability of capturing the subtle nonlinear relations between abundance and environmental covariates and the complex interactions among covariates, nonlinear models are able to produce high accuracy and low bias predictions which reflect the near-real patchy patterns in both large and fine scales. In addition, there is no precondition on the distribution assumptions of the data. This feature greatly enhances their adaptability to adapt different datasets. We choose the nonlinear models from three families, multilayer perceptron, random forest, and support vector machine. Multilayer perceptron is a type of artificial neural network that carry out supervised learning via a back-propagation training algorithm (Ramchoun *et al.*, 2016). With bootstrapping the input data, random forest models use a combination of multiple decision trees to implement prediction (Breiman, 2001). Support vector machine makes prediction by classifying the input dataset into discrete classes across a separating hyperplane. Moreover, it can incorporate multiple variables to map correlations in non-linear space to improve predictions (Cristianini and Shawe-Taylor, 2000). Machine learning models took the block spatial abundance as its training dataset. Predictors were habitat environmental covariates that include location (latitude and longitude), depth, rugosity, sand, gravel, mud, sediment grain size, carbonate, clay, and rock. As discussed in Section 2.1, video survey data is clustered rather than evenly distributed in the study region. It caused some blank areas in which block spatial abundance (training data) was absent. Predicting in these blank areas will encounter the out of sample prediction problem. This problem can introduce great uncertainty and produce erroneous predictions in high risk

(Wenger and Olden, 2012). Therefore, we proposed three strategies, prior knowledge, aggregation, and iteration to overcome this problem.

2.3.1 Prior Knowledge

The goal of involving prior knowledge is to extend the set of training data in the areas that video survey was missing. First, we need to identify the areas that prior knowledge can be engaged confidently. For example, it is well known that red grouper is predominately spatially distributed with high abundance in the eastern portion and very low abundance in the western portion of the Gulf of Mexico. Second, we created several initial predictive maps by running different machine learning models (Figure 2.8 b,c,d). Then, the prior knowledge about spatial distribution was represented by assigning weights to each initial prediction (Table 2.1). Finally, the new training data can be calculated from the weighted combination.

Table 2.1: An example shows the prior knowledge (weights) applied in Figure 2.8.

Area	Prior Weights		
	LR	MLP	SVM
Area 1	0.1	0	0.9
Area 2	0.1	0.45	0.45
Area 3	1/3	1/3	1/3
Area 4	0.1	0.45	0.45
Area 5	0.3	0.4	0.4

The limitation of this strategy comes from the lack of prior knowledge. For example, we can't use it for red grouper in the gap areas of east gulf of Mexico, because the red groupers live there and any inaccurate prior knowledge will cause bias. The same thing happend on red snappers who live in the entire gulf of Mexico.

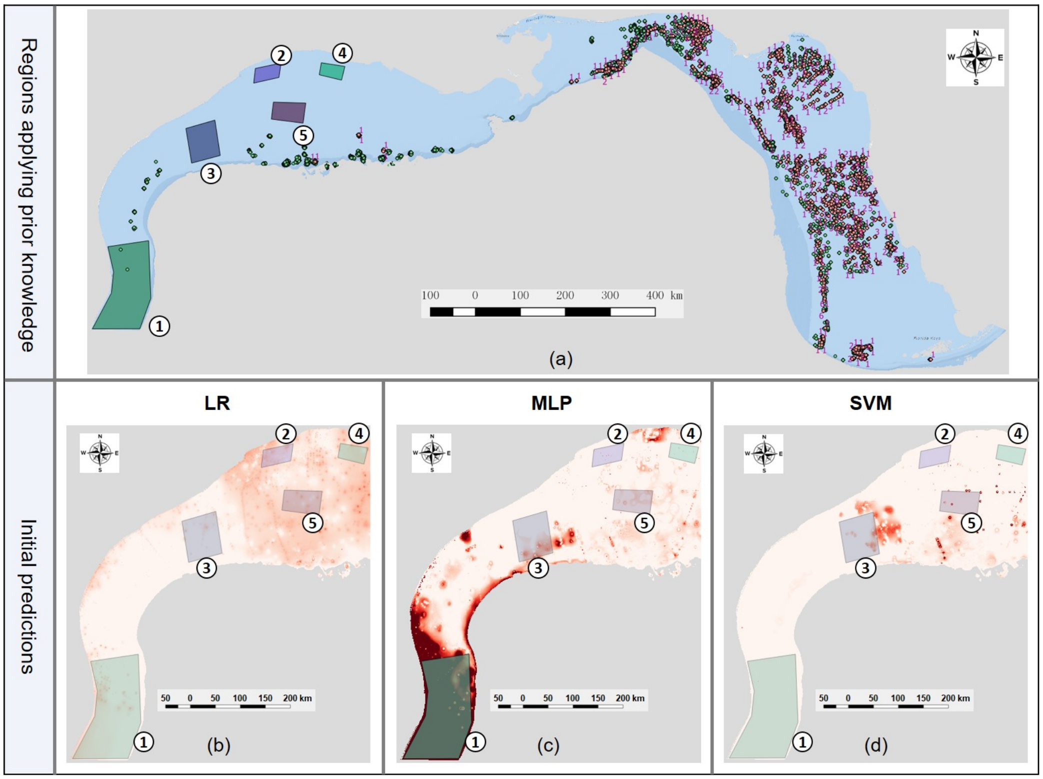


Figure 2.8: The top panel shows the video survey of red grouper throughout the Gulf of Mexico. Red points indicate positive samples and green points indicate negative (zero). The three panels at bottom show initial predictions of spatial abundance in the western portion of the Gulf. They are generated from linear regression, multilayer perceptron and support vector machine respectively. Numbered polygons are the areas applied prior knowledge (weights). The weights are shown in Table 1 below.

2.3.2 Aggregation

Aggregation is a model ensemble approach (Diesing and Stephens, 2015)(Stohlgren *et al.*, 2010) that combines predictions from different models to stabilize the final prediction. In this study, three popular machine learning algorithms were used: multilayer perceptron, random forest and support vector machine. In each category, a range of tuning parameters for that algorithm were tested in a range of possible values. With multilayer perceptron algorithm, the number of hidden layers and the number of neurons in each hidden layer was tried. With random forest algorithm, the maximum number of features allowed in each decision tree, the number of trees to build before averaging for prediction, the number of levels in each decision tree (maximum depth), and the minimum sample leaf size (size of the end node) were tested. With support vector machine algorithm, the kernel parameters that include the type of hyperplane and the shape of the hyperplane were tested. Based on the mean test score obtained from cross-validation, we narrowed down the number of candidate models by a criterion that the mean test scores fell in the range between 0.65 and 0.95. This criterion worked well with model combination in keeping the balance between underfitting and overfitting. Table 2.2 shows the selected models for aggregative prediction of red grouper spatial abundance.

With the selected 33 models in table 2.2, we fit them with training data then make predicitons. The final prediciton is estimated by a suitable statistic of the 33 predicitons, like, mean, weighted mean, median, etc. In the study, we choose median.

Table 2.2: Parameter settings and mean test scores of the selected models for aggregative prediction of red groupers abundance spatial distribution.

Modle	Tuning parameters		Mean test score
MLP	Number of neurons in each hidden layer	(20,)	0.70046559
		(30,)	0.74232753
		(50,)	0.76654593
		(80,)	0.79432473
		(250,)	0.84556754
		(300,)	0.8656241
		(8, 8)	0.74759073
		(10, 10)	0.75592852
		(15, 15)	0.80448259
		(30, 30)	0.87254421
		(50, 50)	0.91866864
		(5, 5, 5)	0.70332429
		(7, 7, 7)	0.7580669
		(10, 10, 10)	0.80493967
		(13, 13, 13)	0.85251525
		(20, 20, 20)	0.89251978
		(30, 30, 30)	0.92922169
Random Forest	max features, number of estimators max depth, min samples per leaf	(0.3, 1000, 7, 150)	0.75829703
		(0.5, 1000, 7, 150)	0.78100404
		(0.5, 1000, 8, 150)	0.81256017
		(0.7, 1000, 8, 100)	0.84390548
		(0.5, 1000, 9, 100)	0.85742639
		(0.7, 1000, 9, 100)	0.87075123
		(0.7, 1000, 5, 50)	0.71127043
		(0.7, 1000, 5, 150)	0.70476749
		(0.7, 1000, 15, 100)	0.91193176
Support Vector Machine	C, γ	(30, 0.5)	0.92571115
		(50, 0.05)	0.74480628
		(50, 0.1)	0.79185051
		(50, 0.15)	0.8290371
		(50, 0.25)	0.88165398
		(70, 0.5)	0.93602774
		(100, 0.1)	0.80533548

2.3.3 Iteration

For each grid cell, the selected models generated predictions. Based on these predicitons, mean, standard deviation, and coefficient of variation (CV) can be calculated. By choosing a threshold for CV (CV0.5), we can filter all the grid cells to get the ones with low variance. Then, new selected cells can be added into the original training data (Figure 2.9). We can iterate this process to expand the training data. However, in order to avoid systematic bias, iterations are better to be less than three. Figure 2.10 shows the distributions of the coefficient of variation (CV) before and after the additional training data was added. It indicates that the one-time iteration can greatly reduce the overall CV, thus stabilize the final prediction.

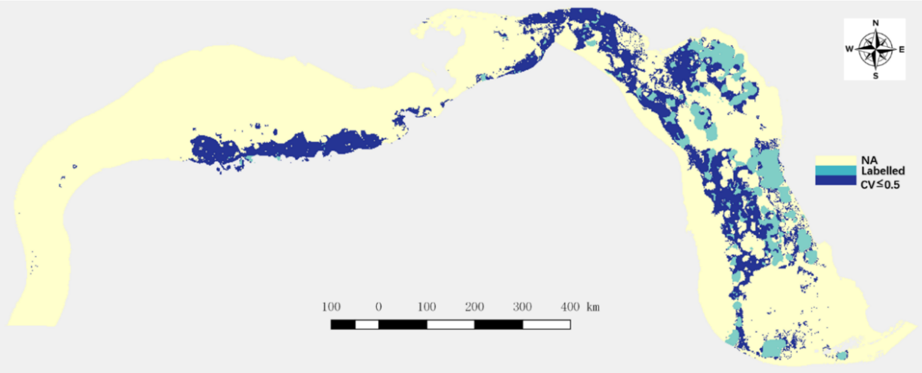


Figure 2.9: Map of original training data (light blue) and new training data (dark blue). The new training data was extended by one-time iteration with criterion $CV \leq 0.5$.

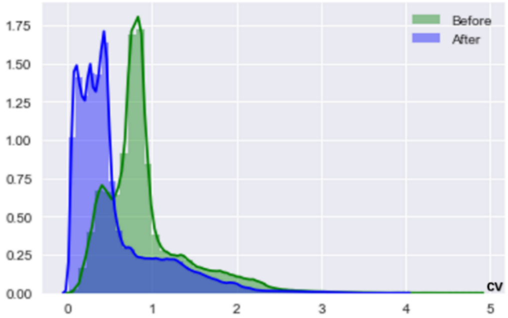


Figure 2.10: The distributions of coefficient of variation (CV) before and after one-time iteration.

2.4 Results and Discussion

Our workflow worked well not only in large scale but also in fine scale. To demonstrate this, there is a comparison with the linear regression model in Figure 2.11. The figure shows that linear regression model can capture the overall trend across the entire Gulf. However, in fine scale, linear regression prediction was too smooth to capture the patchy pattern. In comparison, the prediction of our workflow, hereinafter referred to as non-linear prediction, was able to capture the overall trend as well as the patchy pattern under high resolution. In addition, at some locations, the linear regression prediction shown contradictories to the non-linear prediction. For example, it is well known by biologist that red grouper distributes higher abundance in areas with higher level rugosity and gravel over sea bottoms. Its because their affinity for structure and their role as ecosystem engineers excavating pits in which to live (Harter *et al.*, 2017) (Coleman *et al.*, 2011). Circle number one in Figure 2.11 (panels b-1 and b-2) shows that non-linear prediction correctly gave higher abundance in areas of higher rugosity (panel c), while the linear regression prediction shown the opposite. Similarly, when considering the locations of gravel habitat, circles numbered two, three, and four in Figure 2.11 (panels b-1 and b-2) shows that non-linear prediction correctly demonstrates the positive relation between abundance and the level of gravel. But the linear regression prediction shows inconsistent results. Especially, linear regression prediction at circle number two illustrates a self-contradictory.

The nonlinear prediciton abundance maps of red grouper and red snapper were shown in Figure 2.12(b) and 2.13(b). To validate the goodness of predicting, we choose the maps of spatial catch per unit effort (CPUE) obtained from fishery data (Figures 2.12(a) and 2.13(a)) (McDonald *et al.*, 2001). Figure 2.12(c) and 2.13(c) show the prediciton that corrected by considering the impact of pollution and over-

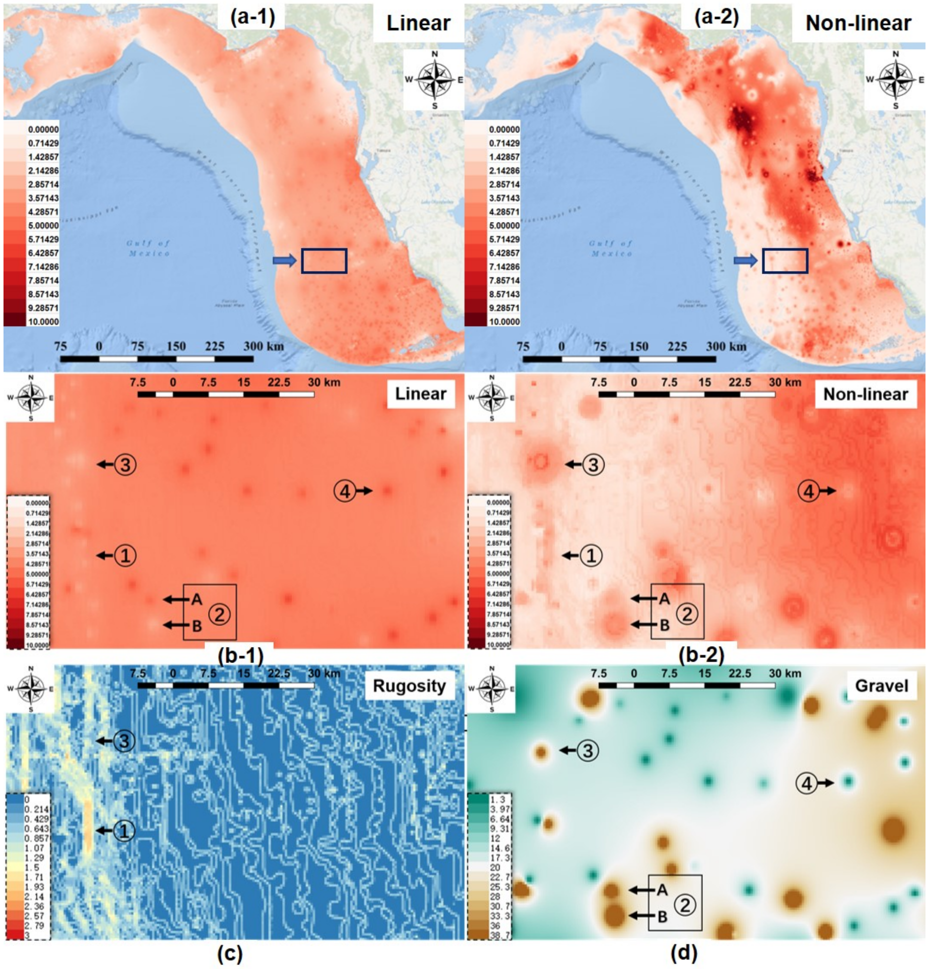


Figure 2.11: Comparison between linear regression prediction and non-linear prediction. Panels a-1 and a-2 show the two predictions in a large scale. Panels b-1 and b-2 show the two predictions in a small scale which is 10 times finer than the large scale (area in blue rectangle in panels a-1 and a-2). Panels c and d show the spatial distributions of rugosity and gravel respectively corresponding to the area in panels b-1 and b-2.

fishing. Overall, the non-linear prediction is consistent with CPUE map. However, it is important to acknowledge that they do notcoincide each othertotally. Because the quality, quantity, and location of fishery-dependent data were influenced by the decision-making behaviors of commercial fishers (Saul *et al.*, 2013). It made the CPUE map biased from real abundance distribution.

In catch per unit efforts (CPUE) map, bias can be introduced from the amount

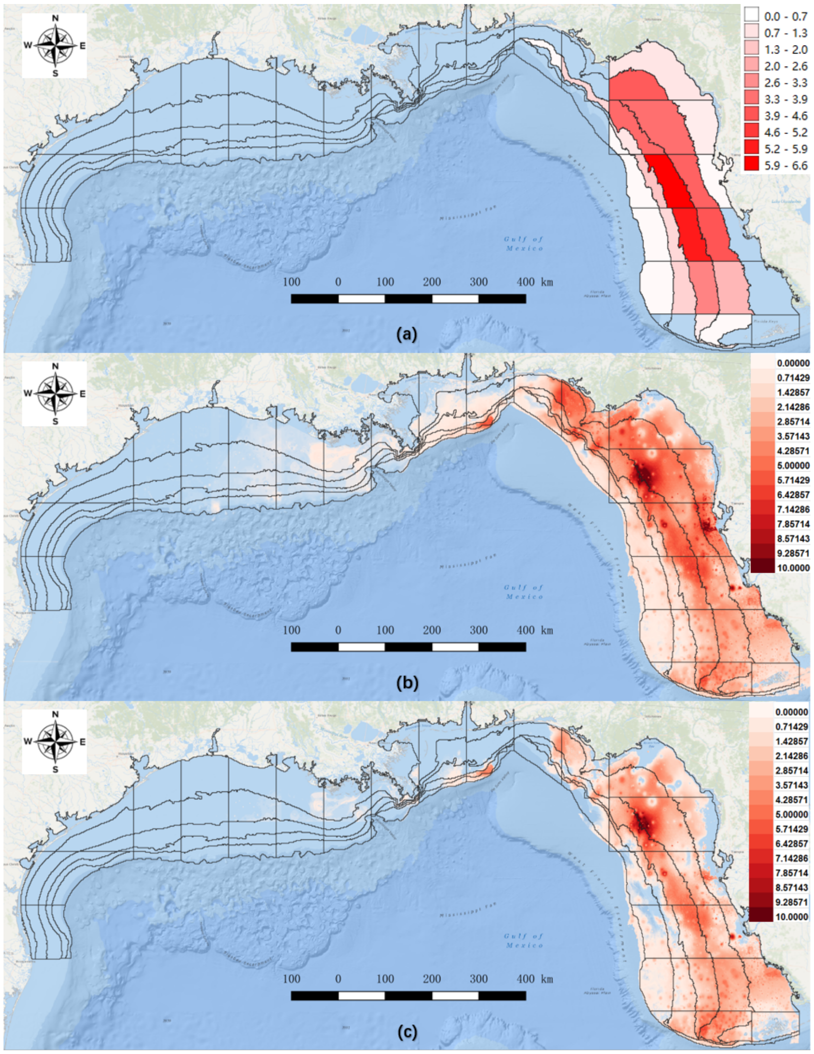


Figure 2.12: Abundance spatial distribution of red grouper. The top panel represents spatial catch per unit effort map obtained from logbook data. The middle panel represents the non-linear prediction. The bottom panel shows the prediciton that corrected by considering the impact of pollution and overfishing.

of catch. For example, Figure 2.14(b) shows relatively low abundance in the area circles numbered 3 and 4. In fact, there is a zone in the middle of these two areas (Figure 2.14(a)). The sea bottom of this zone has high-level rugosity and covered by hard and soft corals. The environment condition is desirable for the living of red grouper (Coleman *et al.*, 2011). As a part of Florida middle grounds habitat of particular concern project, this zone is protected from some fishing gear types

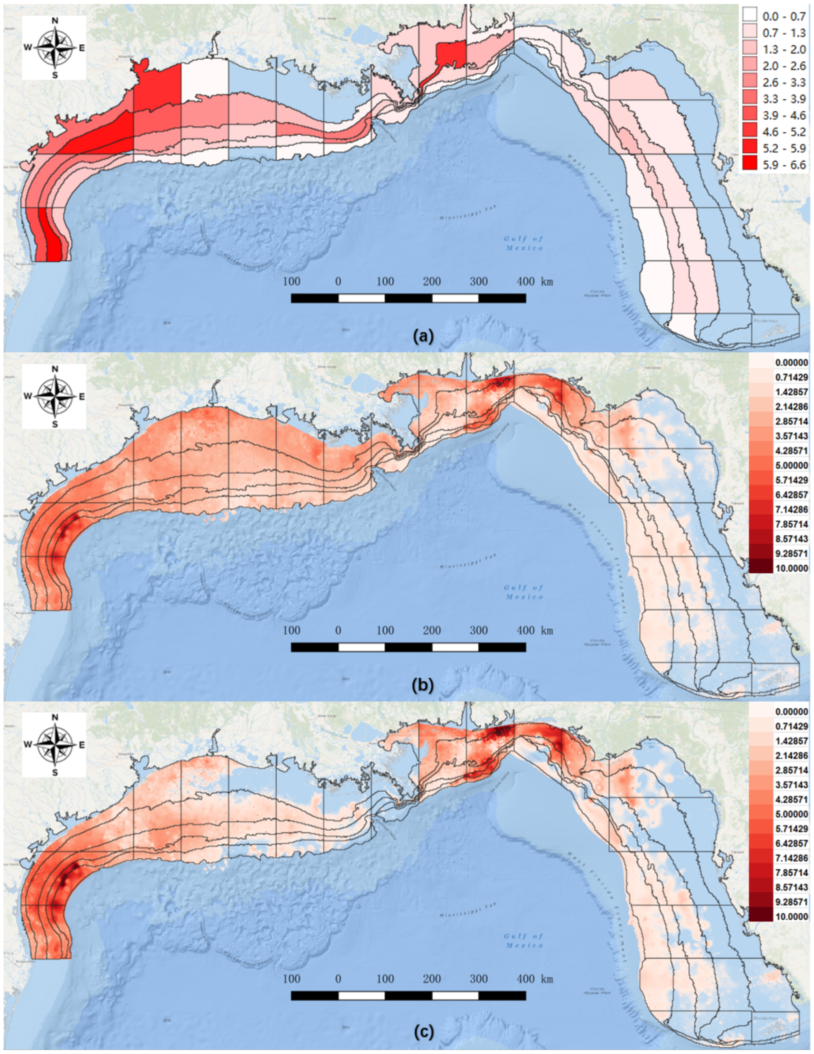


Figure 2.13: Abundance spatial distribution of red snapper. The top panel represents spatial catch per unit effort map obtained from logbook data. The middle panel represents the non-linear prediction. The bottom panel shows the prediciton that corrected by considering the impact of pollution and overfishing.

including bottom longlines, trawls, dredges, pots and traps (Lembke *et al.*, 2017). CPUE prediction in this zone is highly biased, since it depends on the amount actually fished. Contrarily, the nonlinear models which directly employed habitat information was able to appropriately predict near-real abundance based on real environment conditions. Figure 2.14(a) illustrates that nonlinear prediction was able to capture the high abundance in the protected zone.

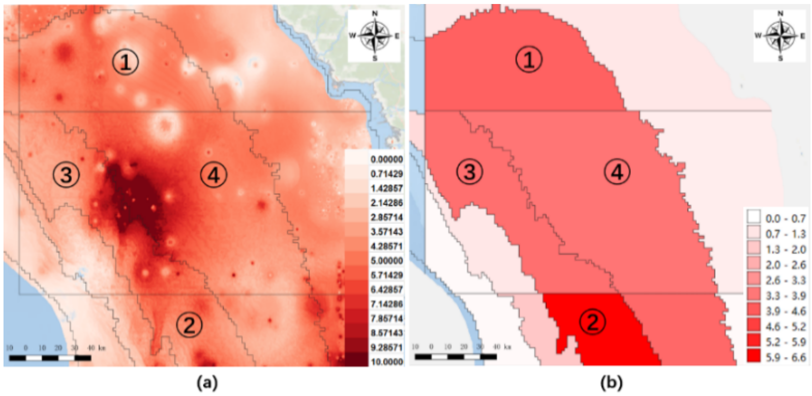


Figure 2.14: Comparison of red grouper abundance maps between non-linear prediction and CPUE. Non-linear prediction can catch the high abundance patchy area (between circles numbered 3 and 4). This area is highly suitable for the living of red grouper because it is covered by hard and soft corals and protected from some fishing gear types including bottom longlines, trawls, dredges, pots and traps. However, CPUE map shows highly biased prediction due to it can be distorted by fishery policy.

Catch per unit efforts (CPUE) map may also introduce bias from the amount of efforts. For example, in Figure 2.15(b), the non-linear prediction of red snappers abundance in western Gulf of Mexico shows that the abundance in region circle numbered 1 was higher than regions circles numbered 2 and 3. It is reasonable because region circle numbered 1 in Figure 2.15(a) has higher level mud on sea bottom. And its well known by biologist that red snappers occupy mud bottom during much of their life history. However, CPUE map in Figure 15 (c) gave opposite answer. We can see there are only two seaports (in blue circles) in western of Gulf of Mexico. For both of them, the cost of fishing in region circle numbered 1 is higher than regions circle numbered 2 and 3. The high cost or effort distorts CPUE prediction in this area far from the real abundance. To this end, depending on fishery-independent environmental data nonlinear prediction maps could be less bias than CPUE maps.

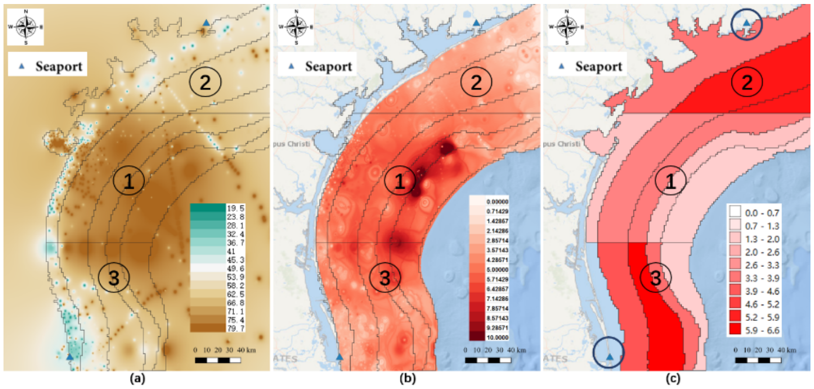


Figure 2.15: Maps representing the spatial distribution of mud levels (panel a), the non-linear prediction (panel b), and CPUE map (panel c). Blue triangles in blue circles on panel c indicate the locations of fishing port.

The generalizability of the multistage workflow can be explained as follows.

First, the distribution of many organisms is patchy across the landscape or seascape. A strength of our workflow and nonlinear predicitons (Figure 2.2) is the ability to capture patch dynamics from sparse data, which will render them applicable to many organisms.

Second, video survey methodology is commonly used to capture presence and abundance data, both in marine and terrestrial ecosystems. As a result, our workflow is well suited for applications of many already existing video survey datasets collected in a variety of ecosystems. Actually, different video simulations can flexibly be adapted to this workflow without having to make any changes to the rest stages.

Third, as mentioned in the end of Section 2.2.2, if the matter is spatial distribution rather than absolute abundance. The only thing you need to know there is the relationship (function) between the catch ratio and population density. If the relationship is a linear function, the first two stages of workflow can be omitted. Even the exact form of the linear function is not required. You can assign an arbitrary value to the coefficient for the linear function and continue subsequentstages of the workflow. The effect of coefficient will be canceled automatically. Fortunately, in most cases, the relationship between catch ratio and population density is a linear function. Otherwise, you have to find the exact form of this function. If it is possible, you can still get rid of the first two stages of the workflow. Final, the flexibility of the workflow also comes from the loose coupling relations among its stages. For examples, a moving window smoothing can take the place of random window smoothing. Another example, you can add more nonlinear (machine learning) models or different parameterizations. Note that if you change the set of more nonlinear (machine learning) models, the final prediction will change as well. However, when you apply the techniques in Section 2.3, such as, controlling the level of predictive accuracy, aggregation and iteration, the final prediction will go stable.

In this chapter, we proposed a generalizable multistage workflow for the nonlinear regression models to predict maps of abundance spatial distribution for reef fish species. This workflow can effectively handle zero-inflated sampling data without strong assumptions. The nonlinear prediction has the advantages, high accuracy, low bias and well-performed in multi-resolution. Moreover, high adaptivity of the workflow makes it suitable to different applications and datasets.

Chapter 3

A TWO-STAGE MODEL

The purpose of study in this chapter is to develop a spatial regression model for analyzing the soil carbon stock (SOC) data. Different from the application in Chapter 2, the desired model should perform well in both prediction and interpretation. Unfortunately, as mentioned in Chapter 1 there is trade-off between the two goals. For example, generally speaking, the linear regression model has good interpretability but bad prediction accuracy. In contrast, the nonlinear models are good at predicting but the black-box property harms their interpretability. In this chapter, we proposed a two stage model trying to break the trade-off between prediction and interpretation. Section 3.1 introduces the data we used in this study. In Section 3.2, a two-stage model is proposed. The model's abilities in interpretation and prediction are discuss from a conceptual view. The results and discussion are presented in Section 3.3.

3.1 Data

The soil carbon stock (SOC) data comes from the rapid carbon assessment study initiated by the Natural Resources Conservation Services Soil Science Division of the U.S. Department of Agriculture (USDA) Staff and Loecke (2016). More than 6200 sites across the conterminous United States were established according to a multilevel stratified random sampling scheme. SOC stock for a fixed soil depth (0 - 30 cm) was calculated using (3.1) (Adhikari *et al.*, 2020). Figure 3.1 shows the map of SOC data in a log transformed scale.

$$SOC_{stk} = SOC \times BD \times D \times (1 - \frac{CF}{100}) \tag{3.1}$$

where SOC_{stk} is the SOC stock ($Mg\ ha^{-1}$), SOC is the SOC content (g 100 g^{-1}), BD is the soil bulk density ($Mg\ m^{-3}$), D is the given soil layer thickness (cm), and CF is the volumetric fraction of the coarse fragments.

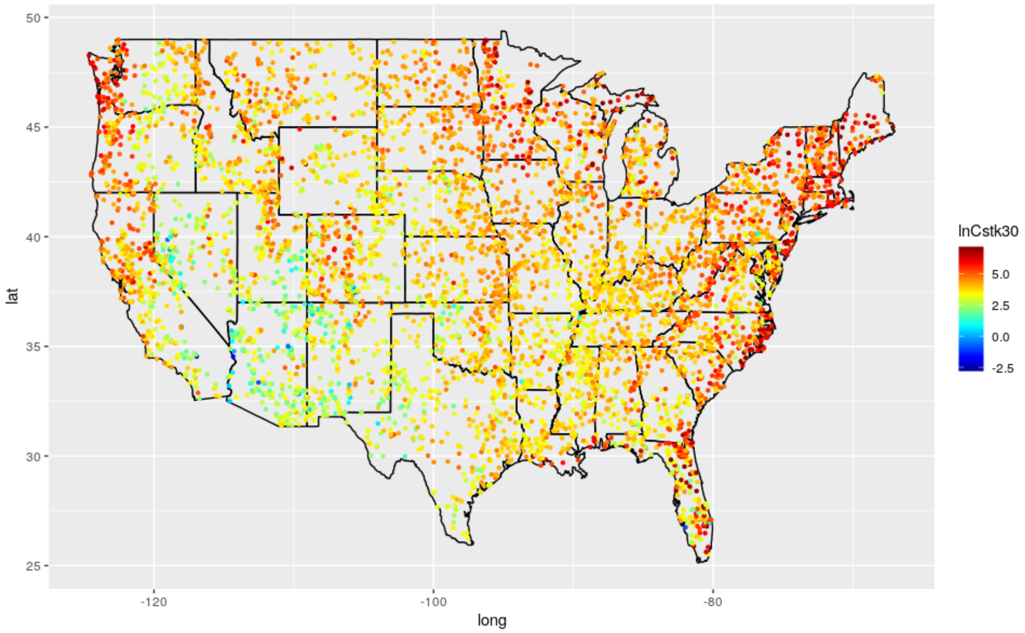


Figure 3.1: Soil carbon stock (SOC) data. The scale of SOC data was transformed by the nature log function. This transformation normalized the SOC data (Figue 3.2) for the convenance of modeling.

A wide range of environmental covariates (31 variables) were collected and evaluated as SOC predictors. Table 3.1 lists their name, a brief description and their source. Figure 3.2 shows a summary of SOC data and eight environmental covariates.

Different from the application in Chapter 2, this data is neat and tidy. As Figue 3.1 showing, the samples are well scattered across the whole area. It's good for prediction. And also, there are 31 covariates which is relatively sufficient for the study of interpretation.

Table 3.1: Environmental Variables Description and Data Source

Environmental variable	Brief description	Data source
Precipitation (PPT)	30-yr (1981 to 2010) annual average	http://www.prism.oregonstate.edu/normals
Precipitation of the driest season (PDRY)	30-yr (19712000) annual average precipitation of the driest month	http://worldclim.org/bioclim
Potential evapotranspiration (PET)	30-yr (19712000) potential evapotranspiration	https://doi.org/10.6084/m9.figshare.7504448.v3
Precipitation of the wettest season (PWET)	30-yr (19712000) annual average precipitation of the wettest month	http://worldclim.org/bioclim
Dew point temperature (TD)	30-yr (19812010) annual average dew point temperature	http://www.prism.oregonstate.edu/normals
Minimum temperature (TMIN)	30-yr (19812010) annual average minimum temperature	http://www.prism.oregonstate.edu/normals
Mean temperature (TMEAN)	30-yr (19812010) annual average temperature	http://www.prism.oregonstate.edu/normals
Maximum temperature (TMAX)	30-yr (19812010) annual average maximum temperature	http://www.prism.oregonstate.edu/normals
Ecological region (ECOL3)	Ecological zone map at level 3 legend	DerivedfromgSSURGO
Net primary production (NETPP)	Annual terrestrial primary production	DerivedfromLandsat
Landsat Band 3 (RED)	Landsat Band 3 for 2014	http://earthenginepartners.appspot.com/science-2013-global-forest/download_v1.6.html
Landsat Band 5 (SW1)	Landsat Band 5 for 2014	http://earthenginepartners.appspot.com/science-2013-global-forest/download_v1.6.html
Landsat Band 7 (SW2)	Landsat Band 7 for 2014	http://earthenginepartners.appspot.com/science-2013-global-forest/download_v1.6.html
National land cover database (NLCD)	Land cover of the United States for 2011	
Potential vegetation (PVEG)	U.S. Potential natural vegetation	Original Kuchler Types, v2.0
Normalized difference vegetation index (NDVI)	Calculated as (NIR RED)/(NIR + RED), where, NIR is near-infrared band (Landsat Band 4)	http://earthenginepartners.appspot.com/science-2013-global-forest/download_v1.6.html
Elevation (DEM)	Land surface elevation	Derived from the national digital elevation dataset (NDEM) from U.S. Geological Survey
Slope aspect (ASPECT)	Direction of the steepest slope from the north	Derived from the DEM
Slope length factor (LSFACTOR)	Slope length factor calculated as in the USLE (universal soil-loss equation)	Derived from the DEM
Multi-resolution valley bottom flatness index (MRVBF)	Potential depositional areas	Derived from the DEM
Melton ruggedness number (MRN)	Melton ruggedness number	Derived from the DEM
Mid-slope position (MSPOS)	Covers the warmer zones of slopes	Derived from the DEM
Wetness index (SAGAWI)	Topographic wetness index with modified catchment area	Derived from the DEM
Slope height (SLOPEHT)	Height of the local slope	Derived from the DEM
Slope gradient (SLOPE)	Local slope gradient in percent	Derived from the DEM
Valley depth (VALDEP)	Calculates the extent of valley depth	Derived from the DEM
Drainage class (DRNG)	Natural soil drainage class	Derived from gSSURGO
Surface geology (GEOSUR)	Surficial geology class	Derived from gSSURGO
Hydrological group (HYDRO)	Hydrologic soil group class	Derived from gSSURGO
Soil order class (SOIL)	Taxonomy soil order class	Derived from gSSURGO
Soil temperature regime (SOILTR)	Soil temperature regime class	Derived from gSSURGO

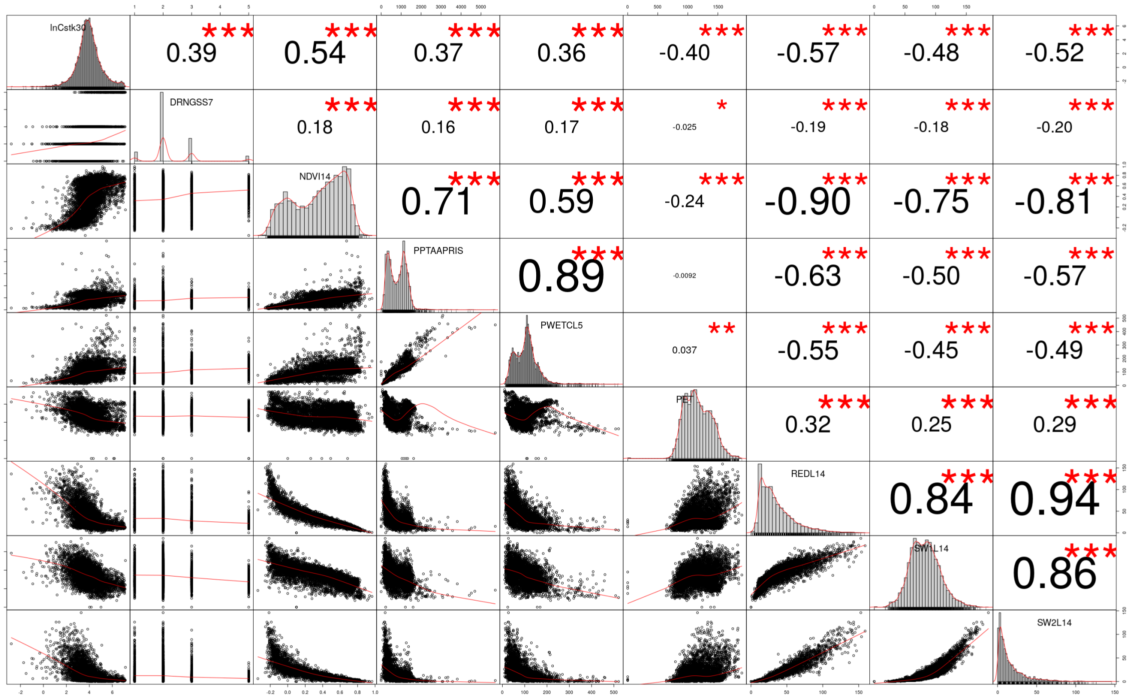


Figure 3.2: The summary of SOC data and 8 environmental covariates. The numbers are correlation values between variables.

3.2 Methods

As mentioned in the beginning of this chapter, the challenge of this study is the trade-off between prediction and interpretation. To break this trade-off, we propose a novel two-stage statistical method that combines global mostly-linear effects (Stage-1) and with non-linear effects (Stage-2). In particular, Stage-1 relies on the universal regression kriging whereas Stage-2 is based in a Generalized Additive Model with splines to capture non-linear effects.

3.2.1 The Two-stage Model

The two-stage model is built on the basis of two well studied statistical models, universal regression Kriging and generalized additive model.

1) Universal regression kriging (URK)

The Universal regression kriging relies on the expression of the quantity of interest Y as follows.

$$Y(s) = f(s) + X(s)\beta + \lambda(s) + \epsilon(s) \tag{3.2}$$

where s is the spatial location, $f(s)$ is a low degree polynormial function that can capture the deterministic spatial trend of dependent variable; $X(s)\beta$ is a regression part to capture the global linear relationship between the dependent variable $Y(s)$ and the explanatory covariates $X(s)$; $\lambda(s)$ is a stochastic part that captures the spatial structure of the variable $Y(s)$, $\lambda(s)$ is generally assumed to be a zero-mean stationary Gaussian process; $\epsilon(s)$ is the nugget effect, usually assumed independent from Y and independent across locations, and identically distributed.

2) Generalized additive model (GAM)

Generalized additive models were originally invented by Hastie and Tibshirani in 1986 (Hastie and Tibshirani (1986), Hastie and Tibshirani (1990)). GAM assumes the relationships between the individual predictors X and the dependent variable Y follow smooth functionals that can be linear or nonlinear. These smooth functional relationships can be estimated and added up as the predictors of $E(Y|X)$ and is expressed as following.

$$Y = \beta_0 + f_1(X_1) + ... + f_p(X_p) + \epsilon \tag{3.3}$$

where $f_i(X_i)$ is an arbitrary smooth univariate function of X_i, usually based on basis decomposition such as splines; ϵ are i.i.d errors. Meanwhile, the predictor function has constraint equal to zero.

$$E(f_i(X_i)) = 0, i = 1, ..., p$$

3) The two-stage model

The proposed two-stage universal regression kriging generalized additive model is a workflow in which we apply the universal regression kriging in the first stage and the generalized additive model in the second stage.

[Stage-1:] Universal regression kriging model

$$Y(s) = f_{spl}(s) + X(s)\beta + \lambda(s) + \delta(s) \tag{3.4}$$

where $f_{spl}(s)$ is a linear function of spatial coordinates capturing the global linear spatial trend; $X(x)\beta$ is the linear regression of covariates representing the global linear effects of covariates $X(s)$ on $Y(s)$; $\lambda(s)$ is a zero mean stationary Gaussian process which explains the global stationary spatial dependence of the process Y; $\delta(s)$ are residuals of the first stage.

[Stage-2:] Generalized additive model

$$\delta(s) = f_{sps}(s) + \sum_{i=1}^{p} f_i(X_i(s)) + \epsilon(s) \tag{3.5}$$

where $f_{sps}(s)$ is a spatial smoother that handle the nonlinear and nonstationary spatial dependence; $\sum_{i=1}^{p} f_i(X_i(s))$ are the additive nonlinear univariate functions for each covariate; $\epsilon(s)$ is a pure white-noise error.

3.2.2 *Model Interpretability and Analysis Flow*

There is no mathematical definition of model's interpretability. However, we can consider it as the degree to which a human can understand/explain the model. In this section, we discuss 3 levels of interpretability.

The concept that corresponds to the first-level (model-level) interpretability of a model is the R^2 coefficient expressed as the ratio of model explained variation and total variation.

$$R^2 = 1 - \frac{SSE}{SST} = \frac{Explained\ Variation}{Total\ Variation}$$

In the two-stage model (3.4) and (3.5), all the components are additive. It's a elegant and powerful assumption that offers a natural way to decompose the interpretability of model into its components. The key idea in the definition of R^2 is the explained variation by the model. Similarly, we can generalized this idea to the second-level (component-level) interpretability (Figure 3.3). In the first stage, the Universal Regression Kriging (URK) models the global variations of the data. In particular, the URK decomposes the total variation into 4 parts: the variations explained by a global linear spatial trend, the variations explained by the linear regression of covariates, the variations explained by a zero-mean stationary spatial random process and the remaining unresolved variations. In the second stage, the residuals of URK model become the input of Generalized Additive Model (GAM). The variations that can'tnot captured by URK will be handled in GAM. Similarly, GAM decomposes the variations into a non-stationary or non-linear spatial component explained by a spatial smoother and nonlinear covariates component explained by spline smoothers and a pure errors component which can't be explained by both URK and GAM.

The third-level (element-level) interpretability is the explanation of relationship between elements and the response variable Y in each component cited above in the second-level. For example, the global linear relationship between the covariates $X(s)$ and response variable $Y(s)$ can be explained by the coefficients β, the global stationary zero mean Gaussian process $\lambda(s)$ can be characterized and explained by the parameters of its covariance function. Since the element-level interpretability of URK is simple and straightforward, we put more effort on GAM.

n general, GAM has the interpretability advantages of multiple linear regression

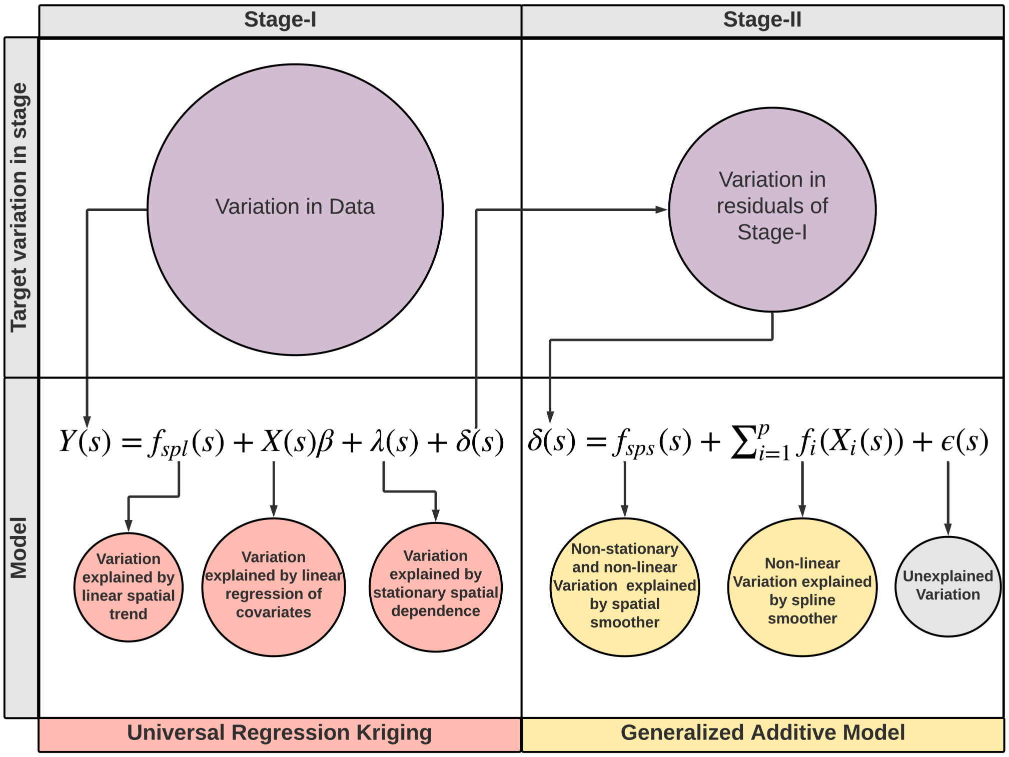

Stage-I
Stage-II
Target variation in stage
Model
Variation in Data
Variation in residuals of Stage-I
$Y(s) = f_{spl}(s) + X(s)\beta + \lambda(s) + \delta(s)$
$\delta(s) = f_{sps}(s) + \sum_{i=1}^{p} f_i(X_i(s)) + \epsilon(s)$
Variation explained by linear spatial trend
Variation explained by linear regression of covariates
Variation explained by stationary spatial dependence
Non-stationary and non-linear Variation explained by spatial smoother
Non-linear Variation explained by spline smoother
Unexplained Variation
Universal Regression Kriging
Generalized Additive Model

Figure 3.3: Two-stage Universal Regression Kriging Generalized Additive Model

model where the contribution of each covariate to the response variable is clearly encoded. In addition, GAM is substantially more flexibility since the relationships between covariates and dependent variables are not assumed to be linear. Since the marginal impact of a single covariate X_i, does not depend on the values of the other covariates in GAM, we can simply interpret its relationship to the response variable by exhibiting the univariate function $f_i(X_i)$. For example, the synthetic example of Figure 3.4, we can say that the expected value of first stage residuals $\delta(s)$ increases exponentially as $X_1(s)$ increases, holding everything else constant. Another important feature of GAM ,which also plays an important role in model interpretation, is the ability to control the smoothness of the predictor functions. With GAMs, we can impose the prior belief that predictive function is inherently smooth in nature, even though the dataset may suggest a more noisy function.

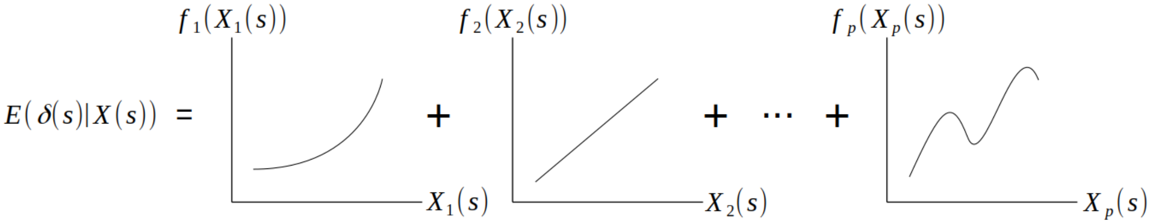


Figure 3.4: Generalized Additive Model Demo

As Figure 3.3 showing, the model is fitted in two stages, which leads to an analysis conducted in two stages bringing the following advantages.

(1) Layers of analysis

In spatial data analysis, extracting global linear trend (with covariates) and stationary spatial dependence is the first interest of geostatistical studies, which we consider as the first layer analysis corresponding to the first-stage of the proposed model. The second layer analysis that aims at revealing the nonlinear relationships between response variable $Y(s)$ and covariates $X(s)$ coincides with the second stage of the analysis flow. The second layer analysis is subtle and on the basis of first layer analysis. The order of these two layers is meaningful since it is challenging to separate the effects of global stationary from nonlinear relationships between $Y(s)$ and $X(s)$ in second layer, if this order is not followed. For example, the existing machine learning models that be applied in spatial context do not include a independent stochastic process, like the $\lambda(s)$ in first stage, to capture the spatial dependence.

(2) Simplicity and flexibility

The universal regression kriging model and generalized additive model are well studied in the statistical community. In the proposed two-stage model, we connect them by following the simple rule that the former's output will be the latter's input. Moreover, there are existing R packages that implement these two models respectively. In this paper, we use the R packages "fields" Nychka *et al.* (2017) for URK and "mgcv" Simon Wood (2019) for GAM. The differences among packages mainly come from the target problems they want to solve and the algorithms that used by the model. For example, fields and FRK Andrew Zammit-Mangion (2020) are two R packages implementing the universal regression kriging model. The main issue that FRK focuses on is computational intensity of large/big spatial data, while fields be designed as a versatile tool for spatial analysis with moderate size data. The algorithms they utilized to estimate parameters are different as well. FRK uses EM algorithm, while fields uses REML and GCV algorithms. Various choices of packages offer great flexibility for analyzing data. We can select the most suitable packages according to the requirements in practise.

3.2.3 Prediction Setup

In predictive modelling and especially with the increasing use of machine learning techniques, a trade-off emerges between interpretability and accuracy of prediction. One of the major goals of this paper is to find an optimal framework to balance this trade-off. As Figure 3.3 shows, the two-stage model can accommodates linear and nonlinear, stationary and non-stationary variations. In the following section, we assess the predictive accuracy of the proposed two-stage model by comparing it with popular machine learning (nonlinear) models for simulated data and soil organic carbon data.

Figure 3.5 illustrates the framework of model comparison. Five models are evaluated and compared: an ordinary linear regression model, the proposed two-stage model, a random forest model, a gradient boost model and a support vector machine model. Since some models, like gradient boost and support vector machine,

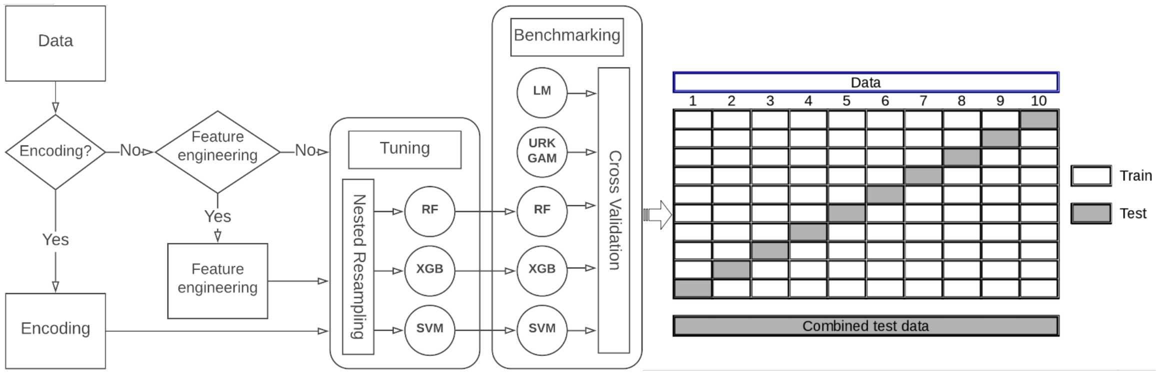


Figure 3.5: Framework of Model Comparison

can not handle categorical data, thus encoding process or feature engineering can be performed as discussed in Section 3.3.2. In order to avoid data linkage, nested resampling was applied to tune the hyperparameters of machine learning models. Then, all five models were compared through a shared cross-validation scheme, as illustrated in Figure 3.5 (right). Statistics of combined test data, like the predictive root mean square error (RMSE) or predictive R^2, were used to evaluate and compare the accuracy of predictions.

3.3 Results and Discussion

In this section, we exhibit the results in terms of interpretability and prediction of the proposed two-stage model fitted on the SOC data and its covariates described in Section 3.2.

3.3.1 Interpretation of Fitted Model

(1) Variable selection

The principle of Occams Razor states that among several plausible explanations for a phenomenon, the simplest is best. Simplicity plays a important role in model's interpretability. We want to explain the data in the simplest way redundant predictors should be removed. Moreover, unnecessary predictors will add noise to the estimation of other quantities that we are interested in. So, the first thing we need to do is variable selection. Since there are several categorical variables in the data, we choose group lasso Yuan and Lin (2006) to select the important variables to be included in the regression model.

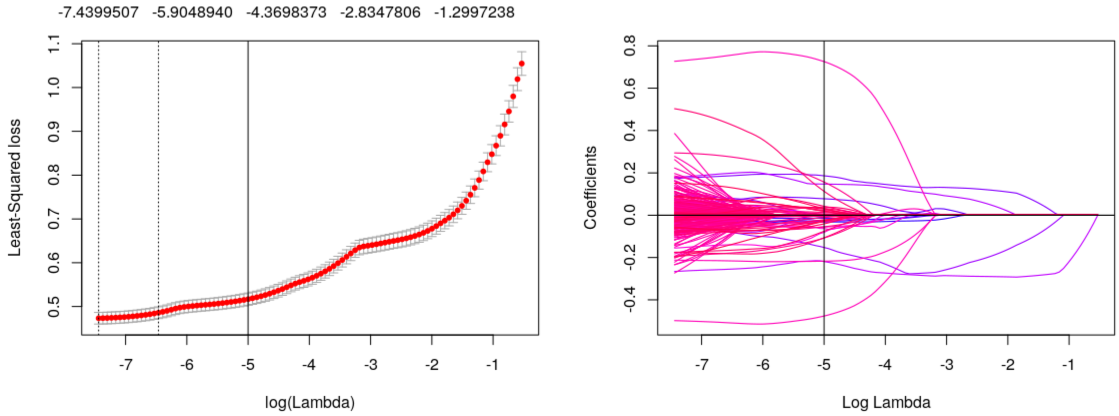


Figure 3.6: Group Lasso Variable Selection

The covariates selected by group lasso are as follows.

ASPECT REDL14 TMEANAA30 PWETCL5 NDVI14 MIDSLPPOS
LSFACTOR DRNGSS7 NLCD2011 SoilOrder SOILMREGIM DEMNED6
Landsat_NPP PET SLOPEHT TMAXAA30 VALDEP

(2) Components of the fitted model on data

With the previously selected covariates, the two-stage model is fitted to the organic soil carbon data. In each stage, we summarize the model information and elucidate the structure of data interpreted by the model.

Stage-1: Universal regression kriging model

The estimated regression coefficients are exhibited in Table 3.2. Because all the covariates were scaled before model fitting, the coefficients βs are comparable with each other and provide the relative importance of each of them to the soil carbon stock.. The coefficients α_0, α_{long} and α_{lat} belong to $f_{spl}(s)$ which is a linear surface trend function of spatial coordinates (s_{long}, s_{lat}).

$$f_{spl}(s) = \alpha_0 + \alpha_{long} * s_{long} + \alpha_{lat} * s_{lat}$$

Table 3.2: Estimated Coefficients by URK

α_0	α_{long}	α_{lat}	β_{ASPECT}	β_{REDL14}
4.3597	0.0025	-0.0025	-0.0284	-0.1703
$\beta_{TMEANAA30}$	$\beta_{PWETCL5}$	β_{NDVI14}	$\beta_{MIDSLPPOS}$	$\beta_{LSFACTOR}$
-0.2100	0.1360	0.1439	0.0621	-0.0373
$\beta_{DEMNED6}$	$\beta_{DRNGSS7}$	$\beta_{NLCD2011}$	$\beta_{SoilOrder}$	$\beta_{SOILMREGIM}$
0.0429	0.2361	0.1180	0.1172	0.0629
$\beta_{Landsat\_NPP}$	β_{PET}	$\beta_{SLOPEHT}$	$\beta_{TMAXAA30}$	β_{VALDEP}
-0.0248	-0.0940	0.0015	0.0218	-0.0256

Each component of the URK model (3.4) can be visualized in Figure 3.7 which provides further interpretation. The fitted R^2 of first stage model URK is about 67.3%. In other words, it means there is approximately 22.7% variation of the data left in the residuals $\delta(s)$ and will be dealt with GAM in second stage.

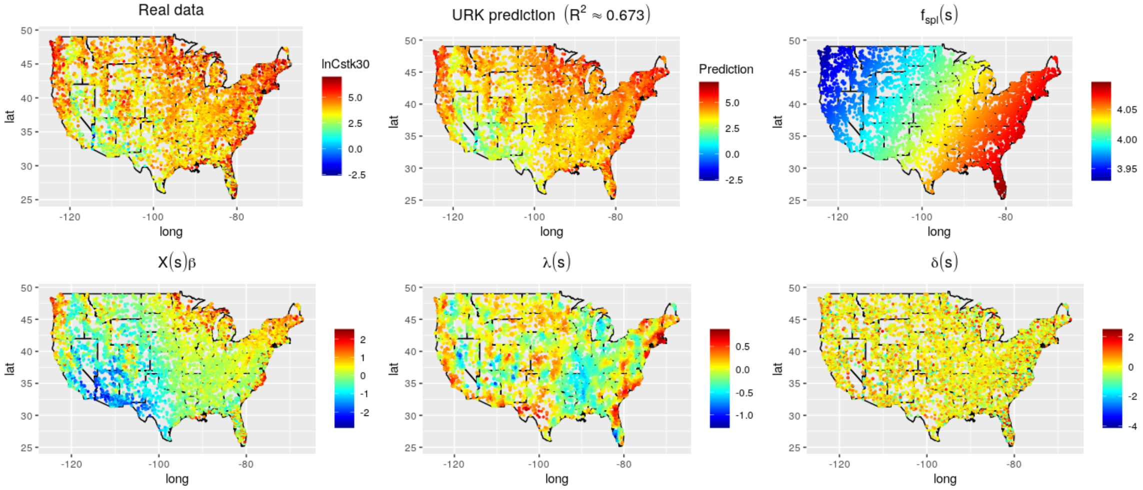


Figure 3.7: Fitted Universal Regression Kriging Model

Stage-2: Generalized additive model

The fitted information of GAM can be found in Table 3.3. Under the significant level 0.01, there are three covariates have non-zero effect on the response variable. They are REDL14, NDVI14 and SoilOrder. All of other covariates are not significant, in other words, they have no effects on the response variable. Figure 3.8 shows the significant fitted predictor functions (smoothers).

Table 3.3: Importance of Smoothers Fitted by GAM in Second Stage

Smoother	edf	Ref.df	F	p-value	
s(Long,Lat)	2.000	2.000	0.530	0.588885	
s(ASPECT)	2.617	2.892	2.661	0.028600	
s(REDL14)	2.733	2.950	8.553	4.20e-05	***
s(TMEANAA30)	1.000	1.000	0.060	0.806594	
s(PWETCL5)	2.084	2.488	1.267	0.197143	
s(NDVI14)	3.000	3.000	20.432	3.62e-13	***
s(MIDSLPPOS)	1.560	1.909	0.562	0.524011	
s(LSFACTOR)	1.658	2.037	0.631	0.508336	
s(DEMNED6)	1.000	1.000	0.078	0.780581	
s(Landsat_NPP)	2.300	2.688	3.391	0.012992	
s(PET)	1.000	1.000	0.151	0.697162	
s(SLOPEHT)	1.000	1.000	0.149	0.699483	
s(TMAXAA30)	1.000	1.000	0.039	0.843379	
s(VALDEP)	1.000	1.000	0.061	0.804975	
s(DRNGSS7)	2.461	2.774	3.776	0.028268	
s(NLCD2011)	1.905	2.283	2.966	0.047557	
s(SoilOrder)	2.330	2.700	6.752	0.000277	***
s(SOILMREGIM)	1.105	1.202	0.450	0.472465	

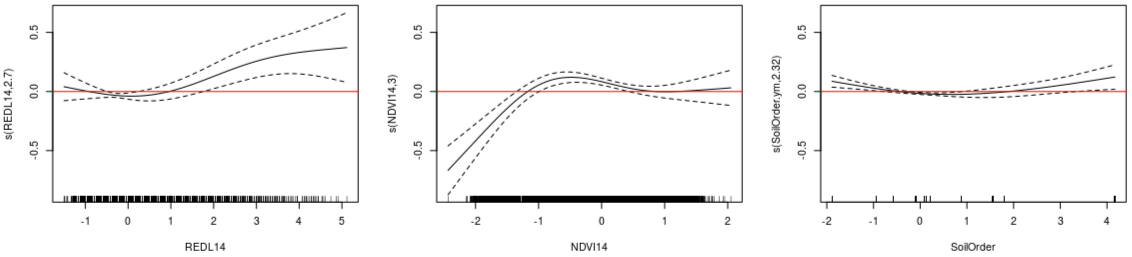


Figure 3.8: Estimated GAM Predictor Functions (Smoothers)

(3) Analysis of spatial patterns of significant predictors

The most obvious benefit of interpretability is understanding the underlying mechanisms of the system. The first stage URK model reveals the global linear relationships between covariates and response variable, then, the second stage GAM corrects the first stage understanding with more subtle non-linear details. Finally, we obtain the overall understanding by adding up results from the two stages. The following Equation (3.6) shows the estimated relationship ($\hat{f}(X_i)$) between covariate X_i and response variable Y.

$$\hat{f}(X_i) = X_i\hat{\beta}_i + \hat{f}_i(X_i) \tag{3.6}$$

where $\hat{\beta}_i$ was the linear coefficient estimated by URK and $\hat{f}_i(X_i)$ was the nonlinear smoothing function fitted by GAM. These functional relationships bring to light the underlying dependencies between the covariates and soil carbon.

In Figure 3.9, the top row plots show the nonlinear fitted functions for covariates, REDL14, NDVI14 and SoilOrder. The visualization of predictor function leads us to check the part of function where the 95% confidence interval is away from zero (segment between the blue vertical lines), which indicates a significant contribution of the covariate to the soil carbon prediction. After representing the significant contribution in the context spatial context (the bottom row of Figure 3.9), one can vizualise the spatial reparttion of the significant predictors. First, significant covariate data tend to exhibit some spatial clustering patterns. Second, the spatial regions of 3 clusters are overlapped and located in the southwest of United States. A reasonable hypothesis may be like that, there is a latent variable which influences the 3 covariates, but it's not included in the data and need further investigation. The clusters provides the information of location where the further investigation should be conducted.

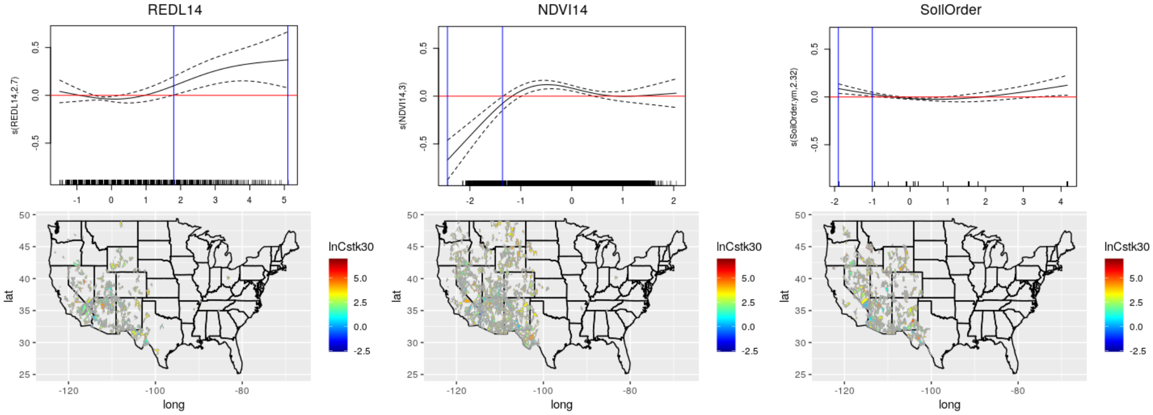


Figure 3.9: Spatial Clusters of Significant Non-linear Predictors

(4) Comparison to GWR model

Geographically weighted regression (GWR) model Brunsdon *et al.* (1998) is an extension of the traditional regression framework and allows the regression coefficients to vary across space. GWR is a very popular geostatistical tool to explore possible spatial patterns of the covariates effects (regression coefficients) and acquire valuable information for further analysis, such as clusters detection. In the following, we compare the interpretation of the components of each GWR and Two-Stage models fitted on the carbon soil data.

First, there are some covariates' effect claimed to be global linear in Two-Stage model but are spatially varying in GWR model. For example, in Figure 3.9 (top row), the effects of covariate NDVI14 shows non-linearity. By comparing the two GWR coefficients maps (Figure 3.10), NDVI14 and TMEANAA30, the spatial variability of TMEANAA30 is larger than NDVI14. So, if we assume the coefficient of NDVI14 is spatially varying (non-linear), GWR model tell us that the coefficient of

TMEANAA30 will be spatially varying as well. In other word, the effect of covariate TMEANAA30 is not globally linear. While Table 3.3 shows that the covariate TMEANAA30 in the second stage GAM has no significant effects on response variable (p-value = 0.806594). It indicates that TMEANAA30 only has the global linear (constant coefficient) relationship with response variable in the first stage URK model. In sum, GWR model and Two-Stage model provide contradicting explanation to covariate TMEANAA30. The reason hides in the stochastic process term $\lambda(s)$ in the first stage URK model. As Figure 3.10 (right) showing, the value $\lambda(s)$ varies spatially compensating for errors in the linear global term. Since the coefficients of GWR are estimated locally, $\lambda(s)$ will cause their estimated values varying across space.

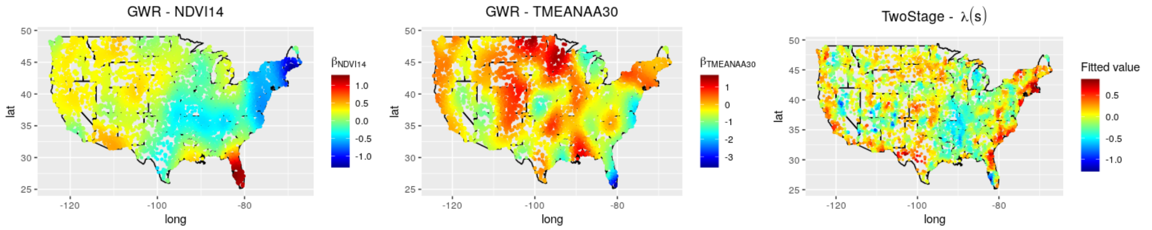


Figure 3.10: Constant or Varying Coefficients

Second, the spatial clustering patterns of the covariate effects differ from GWR to the Two-Stage model. In Figure 3.11, the repartition of REDL14 significance (left) shows non-negligable differences between GWR and Two-Stage model. The maps of NDVI14 (middle) present some similarities but also differences, for example, the junction region between Arizona and Utah, the Northeast states and Florida. For SoilOrder (right), the two maps show similarity in the west of United States but differences in other regions. The reason causing the differences is similar to the analysis of Figure 3.10. In addition, the insignificant parts that we get rid of in GAM plots may also cause the GWR coefficients spatially varying.

In summary, the power of interpretability of Two-Stage model comes from its ability of decomposition. First, Two-Stage model conducts analysis hierarchically and

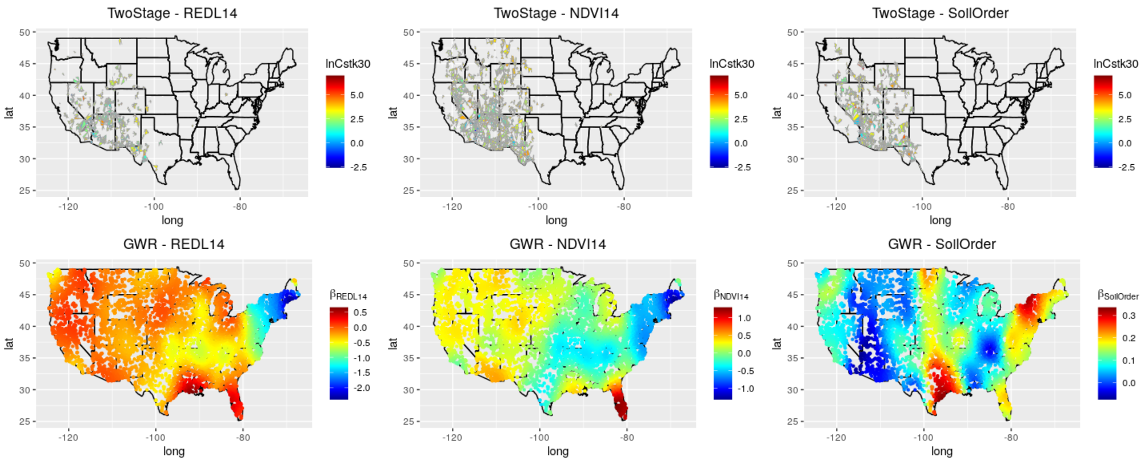


Figure 3.11: Comparing Two-stage model and GWR

the two analysis layers can be easily and clearly decoupled. The second layer (GAM) analysis relies on the basis of extracting out all the influences of first layer (URK). Second, in each layer, the additivity of components ensures decomposition of interpretable components. For instance, the spatial cluster patterns of REDL14 (Figure 3.11), the analysis is base on the condition of extracting out all of other influences, such as the influences from global linearity, global stationary spatial dependence, non-stationary spatial dependence and other covariates. While GWR model mixes up all of those influences, which makes the interpretation ambiguous.

3.3.2 Prediction Results

In this section, we compare the prediction results of different models. The framework for comparison (Figure 3.5) was introduced in Section 3.2.3.

(1) Real data

Different from interpretability, the goal of prediction is accuracy and ability to predict the data characteristics. In order to compare the model capabilities of prediction, we employ all predictors (covariates) into the models.

To work with categorical data, we can use variable encoding approach. However, this approach didn't work well on this real dataset. It caused the predictive R^2 of SVM model ($\approx$ 49%) even lower than the ordinary linear regression model ($\approx$ 54%). To solve this issue, we adopted feature engineering on categorical variables. For example, SoilOrder is a nominal categorical variable. Feature engineering can be applied by using the median value of response variable to instead the nominal number of category (Table 3.4). Now, we transferred the nominal categorical variable to a numerical variable which can be accommodated by any model. After feature engineering (all categorical covariates), we found that the predictive R^2 increased from 49% to 58% with SVM model but changed negligibly with other models.

Table 3.4: Example of Feature Engineering for the Covariate Soilorder

SoilOrder	1	2	3	4	5	6	7	8	9	10	11
y.median	3.9276	3.5199	4.0103	4.0338	3.6944	5.9676	4.0752	4.3900	4.7165	3.0662	4.8314

Table 3.5 lists the predictive rmse and predictive R^2 for the compared models. The Two-Stage model stays competitive to the popular machine learning models, Random forest, XGBoost and SVM. However, comparing to URK model, Two-Stage model only improves the predictive R^2 by 0.5% which is negligible in some circumstances. The reason came from the nature of data. The purely random variation takes a large proportion (approximately 40%) in the total variation of data. It made all the models, except linear regression model, obtaining similar predictive R^2. Regardless of only 0.5% improvement in predictive R^2, Two-Stage model discovered much more useful information comparing to URK model (see Section 3.3.1).

(2) Simulation data

In order to demonstrate the abilities of the two-stage model to compete with popular machine learning models and improve its capabilities compared to URK and linear regression models, we simulate a dataset and conduct the comparison described

Table 3.5: Prediction Comparison on Real Data

Model	Predictive RMSE	Predictive R^2
LM	0.6978181	0.5392069
URK	0.6729866	0.5732470
Two-Stage	0.6687347	0.5786224
RF	0.6534186	0.5958378
XGB	0.6689105	0.5765102
SVM	0.6693818	0.5757998

in Figure 3.5. The response variable $Y(s)$ is simulated as

$$Y(s) = Y_x(s) + \lambda(s) + p(s) + \epsilon(s) \tag{3.7}$$

where the components are generated as follows and their visualizations can be found in Figure 3.12.

$Y_x(s)$ represents the nonlinear relationship between two covariates $X_1(s)$ and $X_2(s)$ and the response variable.

$$Y_x(s) = 0.1 * X_1(s)^3 + 10 * sin(X_2(s) + 3)$$

$$\begin{cases} X_1(s) \sim unif(1, 4) \\ X_2(s) \sim unif(0, 2\pi) \end{cases}$$

$\lambda(s)$ is a zero mean Gaussian process capturing the isotropic stationary spatial dependence. $\lambda(s)$ entirely characterized by a Matrn covariance function as follows.

$$C_\nu(|s_1 - s_2|) = \sigma^2 \frac{2^{1-\nu}}{\Gamma(\nu)} \left(\sqrt{2\nu} \frac{|s_1 - s_2|}{\rho} \right)^\nu K_\nu \left(\sqrt{2\nu} \frac{|s_1 - s_2|}{\rho} \right)$$

where $\Gamma(\cdot)$ is the Gamma function, $K_\nu(\cdot)$ is the modified Bessel function of the second kind, $|s_1 - s_2|$ is the Euclidean distance between spatial point s_1 and s_2, and $\sigma^2 = 1$, $\rho = 0.5$ and $\nu = 1.5$.

$p(s) = 0.005 * s_x * s_y$ represents non-stationary dependence in the spatial coordinates s_x and s_y.

$\epsilon(s) \overset{i.i.d.}{\sim} N(0, 10.24)$ is the pure error also called nugget effect in Geostatistics.

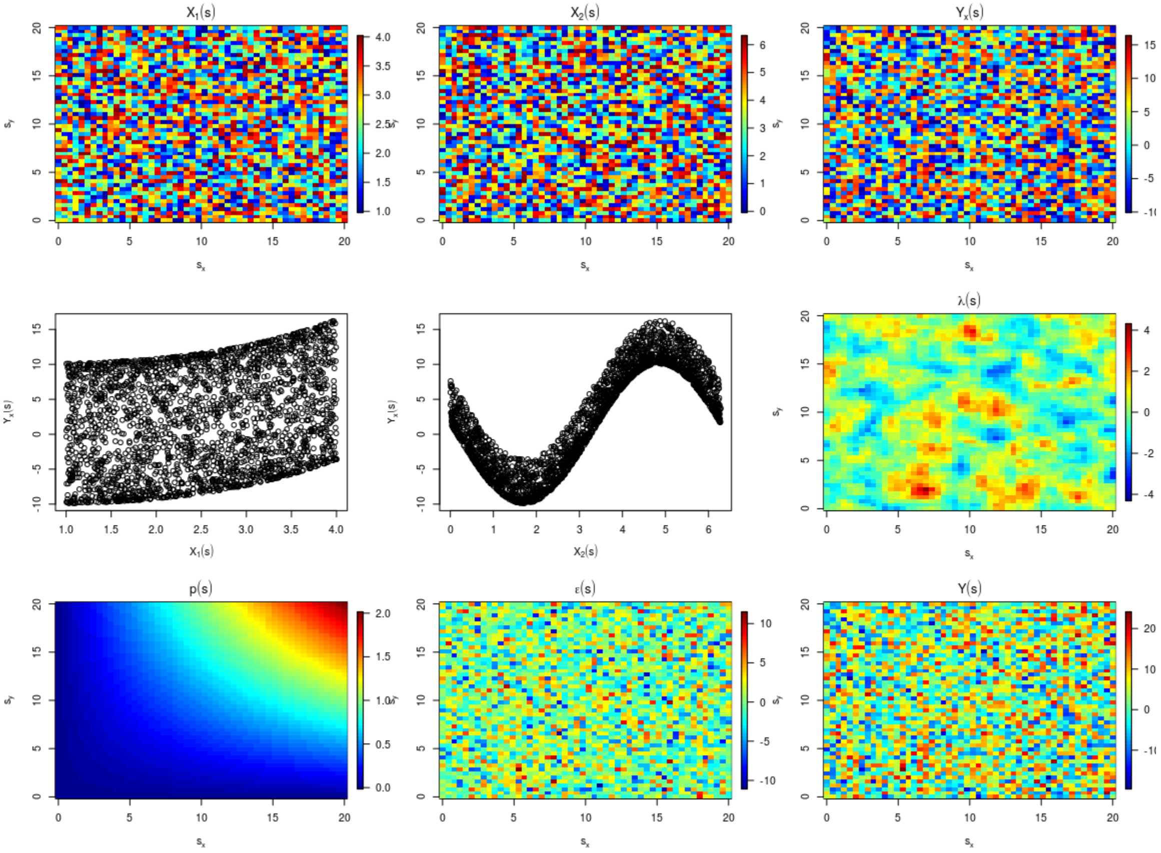


Figure 3.12: Simulated Data

The results of model comparison are shown in Table 3.6. URK model shows a predictive $R^2 \approx 49.3\%$ which is closed to linear regression model 48.4% but far away from Two-Stage model 80.7%. By relationship of URK and Two-Stage model, we know the predictive R^2 contributed by the second stage GAM is 31.4% which is a great improvement. Moreover, comparing to Random forest, XGBoost and SVM, Two-Stage model has higher predictive R^2 and similar results to the random forest. The comparison on synthesized data again proves the predictive ability of Two-Stage model is competitive to popular machine learning models.

Table 3.6: Prediction Comparison on Simulated Data

Model	Predictive RMSE	Predictive R^2
LM	5.695840	0.4843744
URK	5.667649	0.4932645
Two-Stage	3.499716	0.8067849
RF	3.522183	0.8026377
XGB	3.645922	0.7884789
SVM	3.564608	0.7981013

In summary, the two-stage model has good interpretability which is close to the linear regression. Meanwhile, it keeps high prediction accuracy that is competitive to the nonlinear (machine learning) models, like random forest, xgboost and support vector machine. It makes the two-stage model stand out from the rest (Figure 3.13).

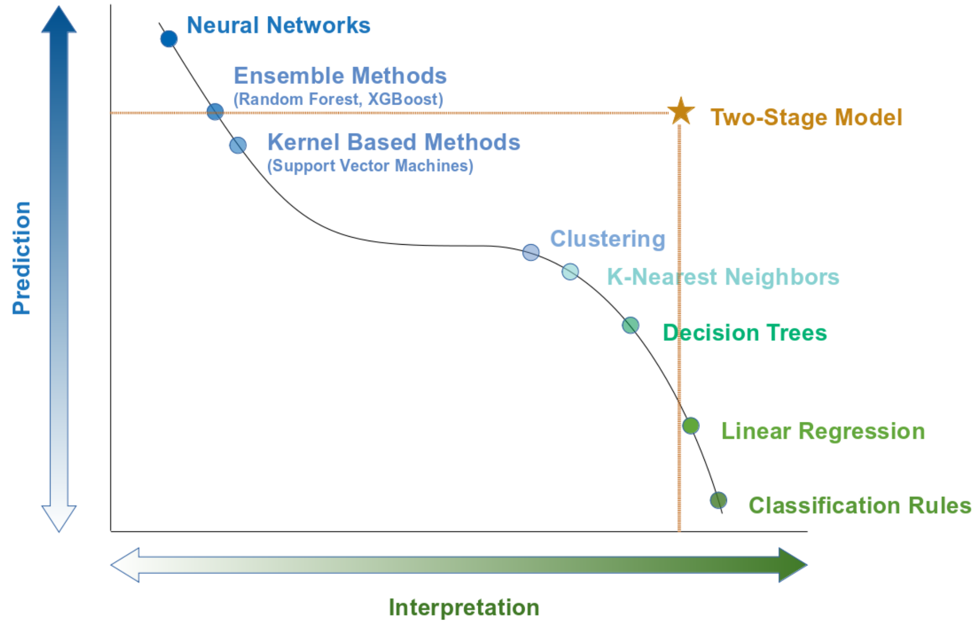


Figure 3.13: Breaking the Trade-off Between Prediction and Interpretation

Chapter 4

GAUSSIAN PROCESS BART

The Bayesian Additive Regression Trees (BART) model is rarely used in spatial applications. One of the reasons is that the error term in BART model is restricted to be independently distributed which is unusual in spatial problems. In this chapter, we get rid of this constraint and propose a Gaussian process BART model for spatial regression problems. First, the traditional BART model is introduced in Section 4.1. Then, in Section 4.2, we develop a new BART model that can accommodate the correlated errors. In section 4.3, the Gaussian process BART model is studied. Section 4.4 shows two experiments and a testing on real data.

4.1 Introduction

Bayesian Additive Regression Trees (BART), proposed by Chipman *et al.* (2010), can be viewed as a sum-of-trees model as follows.

$$y = g(X; T_1, M_1) + ... + g(X; T_m, M_m) + \epsilon, \quad \epsilon \overset{\text{i.i.d}}{\sim} N(0, \sigma^2) \tag{4.1}$$

where y, X are observed dependent and independent variables; ϵ is independent and identically distributed random error; T denotes a tree, consisting of a set of interior nodes with decision rules and a set of terminal nodes; $M = \{\mu_1, ..., \mu_b\}$ where b is the number of terminal nodes of T; $g(X; T_i, M_i),\ i = 1, ..., m$ denotes a single binary regression tree that assigns $\mu_j, j = 1, ..., b$ in M to the observations through T. A example of single binary regression tree is illustrated in Figure 4.1.

BART is inspired by the idea of boosting that sums the contribution of sequential weak learners (trees) to get a much more accurate prediction. Different from

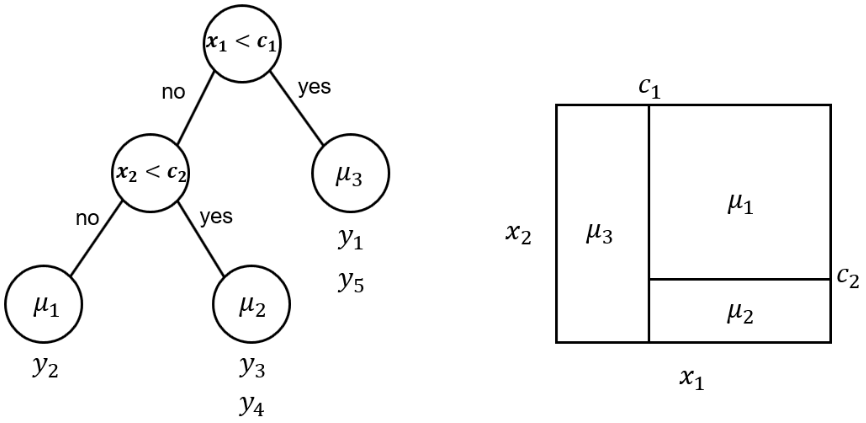


Figure 4.1: (Left) An example of single binary tree, with internal nodes labelled by their splitting rules, terminal nodes labelled with the corresponding parameters μ_i and the observations associated with it. (Right) The corresponding partition of the sample space and the step function.

other boosting methods, like, gradient boosting trees, BART works in a Bayesian framework using prior and likelihood to generate a posterior distribution of the prediction. The posterior distribution provides much richer information than the point estimation of classical regression models. In addition, the Bayesian framework has a built-in complexity penalty mechanism that automatically initializes the model's hyperparameters, like, max tree size, which normally be tuned via cross-validation in other models.

Experiments study (Chipman *et al.*, 2010) showed that BART outperforms other popular machine learning methods, including Neural Nets, Gradient Boosting Trees and Random Forest. Recall the spatial nonlinear regression model which excludes the stochastic process term $w(s)$ in model (1.1):

$$y(s) = f(s; X(s)) + \epsilon(s), \qquad \epsilon(s) \overset{\text{i.i.d}}{\sim} N(0, \sigma^2) \tag{4.2}$$

The BART model (4.1), of course, is a good candidate in this category. But we want to be more ambitious. Since the term $w(s)$ in model 1.1 models the effects of unobserved independent variables. Keeping it in the spatial nonlinear regression model can benefit us in both prediction and interpretation (see Section 4.3.1).

4.2 Bart for Correlated Errors

In BART model (4.1) the error term ϵ is assumed independent and identically distributed, $\epsilon(s) \overset{\text{i.i.d}}{\sim} N(0, \sigma^2)$. We can generalize this assumption and allow the error term has a general correlated structure, $\epsilon \sim N(0, \Sigma)$.

$$y = g(X; T_1, M_1) + ... + g(X; T_m, M_m) + \epsilon, \quad \epsilon \sim N(0, \Sigma) \tag{4.3}$$

We will build the new model (4.3) and illustrate how it works in this section. But, first of all, the question can be simplifie to a single tree model by taking advantage of the reductions $R_j = y - \sum_{k \neq j} g(X; T_k, M_k)$.

$$R_j = g(X; T_j, M_j) + \epsilon, \qquad \epsilon \sim N(0, \Sigma)$$

Hereafter, we remove the subscripts and discuss on the single tree model (4.4).

$$R = g(X; T, M) + \epsilon, \qquad \epsilon \sim N(0, \Sigma) \tag{4.4}$$

4.2.1 Dummy Representation

To understand model (4.4), the first and most impotant step is dummy representation. Simply speaking, dummy representation provides a matrix form to the single tree model (4.4). The tree $g(x; T, M)$ can be denoted as follows.

$$g(X; T, M) = D\mu \tag{4.5}$$

where

$$\mu = [\mu_1, \mu_2, ..., \mu_b]^T$$

and

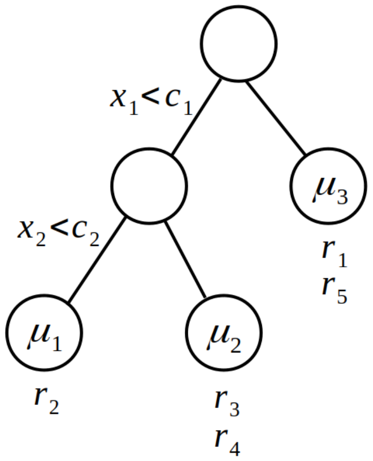
$$D = \begin{bmatrix} d_{11} & d_{12} & ... & d_{1b} \\ d_{21} & d_{22} & ... & d_{2b} \\ \vdots & \vdots & \ddots & \\ d_{n1} & d_{n2} & ... & d_{nb} \end{bmatrix}$$

D is called dummy matrix which is a $n \times b$ matrix. n is the number of observations and b is the number of bottom nodes. For each row in D, there is only one entry equal to 1 and the rest are equal to 0. For example,

$$[d_{i1}, ..., d_{i,j-1}, d_{i,j}, d_{i,j+1}, ..., d_{in}] = [0, ..., 0, 1, 0, ...0] \tag{4.6}$$

is the i^{th} row in D and its j^{th} column is 1. Matrix D can be viewed as a map that maps the observations to the bottom nodes of the tree. The row (4.6) works as mapping the i^{th} observation to the j^{th} bottom node. An example is as following. The dummy matrix D mapped r_2 to node 1, r_3 and r_4 to node 2, r_1 and r_5 to node 3.

$$R = g(X; T, M) = D\mu = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix}$$



Based on the dummy representation, the tree model (4.4) can be re-denoted as a matrix form.

$$R = g(X; T, M) = D\mu + \varepsilon, \qquad \varepsilon \sim N(0, \Sigma) \tag{4.7}$$

The matrix form makes mathematical derivation possible. Moreover, given X this representation perfectly decoupled the components T and M in the tree model. It means if X and T are fixed a dummy matrix D is uniquely determined regardless the value of μ in M. This decoupling will benefit the calculation of marginal likelihood $p(R|X, T)$ which is the pivot of MCMC transitions. The details will be discussed in next section.

4.2.2 Metropolis-Hastings Search

In BART, each tree be updated at every MCMC iteration. Recall (4.4), obviously, to update a tree we need to update its components T and M. Naturally, the structure of tree which is T should be updated first. **?** proposed Metropolis-Hastings algorithm to draw a sequence of trees,

$$T^0, T^1, T^2, ...$$

The sequence starting with an initial tree T^0, iteratively simulate the transitions from T^i to T^{i+1}, $i = 0, 1, 2, ...$, by the following two steps:

(1) Generate a candidate value T^* with probability distribution $q(T^i, T^*)$.

(2) Set $T^{i+1} = T^*$ with probability

$$\alpha(T^i = T^*) = min\{\frac{q(T^*, T^i)}{q(T^i, T^*)}\frac{p(R|X, T^*)p(T^*)}{p(R|X, T^i)p(T^i)}, 1\} \tag{4.8}$$

Otherwise, set $T^{i+1} = T^i$.

In (4.8) the transition kernel $q(\cdot, \cdot)$ and the prior $p(T)$ are same in both traditional BART and new BART. So, (4.9) doesn't change as well.

$$\frac{q(T^*, T^i)}{p(T^i)}\frac{p(T^*)}{q(T^i, T^*)} \tag{4.9}$$

On the other hand, the correlated data goes into (4.10) which is a marginal likelihood ratio. And this ratio is the difference between the traditional BART and the new BART.

$$\frac{p(R|X, T^{i+1})}{p(R|X, T^i)} \tag{4.10}$$

In the discussion of dummy representation, we know that given X and T a dummy matrix D can be uniquely determined. So, the marginal likelihood $p(R|X, T)$ is equal to $p(R|D)$. Then, (4.10) is equal to (4.11) as well.

$$\frac{p(R|D^{i+1})}{p(R|D^i)} \tag{4.11}$$

Now the question is converted to calculate $p(R|D)$. By (4.7), we can get the joint likelihood (4.12).

$$p(R|D,\mu) \sim N(D\mu, \Sigma) \tag{4.12}$$

Then the marginal likelihood can be got by integrated out the μ. The only thing we need is a prior distribution $\pi(\mu)$.

$$p(R|D) = \int p(R|D,\mu)\pi(\mu)\, d\mu \tag{4.13}$$

A Gaussian prior $\pi(\mu) \sim N(\bar{\mu}, Q^{-1})$ is preferred, because it conjugates to (4.12). $\bar{\mu}$ and Q are the mean and precision matrix of the Gaussian prior distribution respectively. (4.14) shows the result of the integration (4.13). The proof can be found in Appendix A.1.1;

$$p(R|D) = \frac{(2\pi)^{-\frac{n}{2}}|\Sigma|^{-\frac{1}{2}}|Q|^{\frac{1}{2}}}{|Q + D^T\Sigma^{-1}D|^{\frac{1}{2}}} exp\{-\frac{1}{2}(-v^T(Q + D^T\Sigma^{-1}D)v + \bar{\mu}^T Q\bar{\mu} + R^T\Sigma^{-1}R)\} \tag{4.14}$$

where, $v = (Q + D^T\Sigma^{-1}D)^{-1}(Q\bar{\mu} + D^T\Sigma^{-1}R)$.

Let $\bar{\mu} = 0$, (4.14) can be simplified to (4.15).

$$p(R|D) = \frac{(2\pi)^{-\frac{n}{2}}|\Sigma|^{-\frac{1}{2}}|Q|^{\frac{1}{2}}}{|Q + D^T\Sigma^{-1}D|^{\frac{1}{2}}} exp\{\frac{1}{2}[R^T\Sigma^{-1}D(Q + D^T\Sigma^{-1}D)^{-1}D^T\Sigma^{-1}R - R^T\Sigma^{-1}R]\} \tag{4.15}$$

Finally, we plug (4.15) into the marginal likelihood ratio (4.11) and get (4.16). The computational complexity of (4.16) will be studied in Section 4.2.4. And the details of calculation can be found in Appendix A.3.

$$\begin{aligned}\frac{p(R|D^{i+1})}{p(R|D^i)} = \frac{|Q^{i+1}|^{\frac{1}{2}}}{|Q^i|^{\frac{1}{2}}} \frac{|Q^i + (D^i)^T\Sigma^{-1}D^i|^{\frac{1}{2}}}{|Q^{i+1} + (D^{i+1})^T\Sigma^{-1}D^{i+1}|^{\frac{1}{2}}} \cdot exp\{\frac{1}{2}R^T\Sigma^{-1} \\ [D^{i+1}(Q^{i+1} + (D^{i+1})^T\Sigma^{-1}D^{i+1})^{-1}(D^{i+1})^T - D^i(Q^i + (D^i)^T\Sigma^{-1}D^i)^{-1}(D^i)^T]\Sigma^{-1}R\}\end{aligned} \tag{4.16}$$

4.2.3 Posterior Distribution of μ

In Section 4.2.2, the tree structure T was updated. Given the new T, we can update M which is the set of μ in the bottom nodes. Since X and new T are known, the likelihood of μ is easily obtained from (4.12).

$$p(R|\mu) = p(R|D,\mu) \sim N(D\mu, \Sigma) \tag{4.17}$$

According Bayesian theory, the posterior probability density function of μ is proportional to the product of its likelihood and prior probability density function (4.18).

$$p(\mu|R) \propto p(R|\mu)\pi(\mu) \tag{4.18}$$

Similar to the calculation of marginal likelihood (4.13), we choose the conjugate prior $\pi(\mu) \sim N(\bar{\mu}, Q^{-1})$. The posterior distribution $p(\mu|R)$ is as (4.19) and the proof can be found in Appendix A.1.2.

$$p(\mu|R) \sim N(\ (Q + D^T\Sigma^{-1}D)^{-1}(Q\bar{\mu} + D^T\Sigma^{-1}R)\ ,\ (Q + D^T\Sigma^{-1}D)^{-1}\ ) \tag{4.19}$$

Furthermore, if let $\bar{\mu} = 0$, (4.18) can be simplified to (4.20).

$$p(\mu|R) \sim N((Q + D^T\Sigma^{-1}D)^{-1}D^T\Sigma^{-1}R, (Q + D^T\Sigma^{-1}D)^{-1}) \tag{4.20}$$

4.2.4 Computational Complexity

The new BART works with the covariance matrix Σ whose dimension is $n \times n$. n is the number of observations. When the data is big, the huge covariance matrix could cause computational problems, for example, the computation of likelihood needs to calculate the $|\Sigma|$ and Σ^{-1}. Their exact calculation requires $O(n^3)$ operations which becomes an impossible mission for a personal computer when n is greater than, for example, one million. In this section, we will investigate the computational complexity of the new BART model. Since a preprocessing step called reordering can greatly simplify the discussion, before digging into the computational stuff, it's worth to spend some time to understand the reordering.

Supposing a tree has b bottom nodes. The dummy matrix D maps n observations to them. Based on this mapping, the observations can be partitioned at most b sets. Reordering means that we reorder all the observations to make them ordered successively in each partition. Since any reordering is a map and can be achieved by multiplying a permutation matrix. Let's assume permutation matrix P^T (transpose of P) can realize the reordering. Then, the reordered dummy matrix, D_P, can be denoted as following.

$$P^T D = D_P, \qquad D = P D_P \tag{4.21}$$

where

$$D_P = \begin{bmatrix} d'_{11} & d'_{12} & ... & d'_{1b} \\ d'_{21} & d'_{22} & ... & d'_{2b} \\ \vdots & \vdots & \ddots & \\ d'_{n1} & d'_{n2} & ... & d'_{nb} \end{bmatrix}$$

and

$$d'_{ij} = \begin{cases} 0, & i \notin n_j, \\ 1, & i \in n_j. \end{cases} \qquad i = 1, ..., n; \qquad j = 1, ..., b.$$

where $n_j, j = 1, .., b$ is the index set of observations that be mapped to j^{th} bottom node.

Intuitively, D_P is formatting as follows.

$$D_P = \begin{bmatrix} 1 & 0 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & ... & 0 \\ 0 & 1 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & ... & 0 \\ 0 & 0 & ... & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & ... & 0 \\ 0 & 0 & ... & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & ... & 1 \end{bmatrix}$$

Recall the example in Section 4.2.1, it's easy to find the reordered matrix D_P and permutation matrix P.

$$D = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = PD_P = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Similar to (4.21), R and Σ also have their reordered counterparts.

$$R = PR_P, \quad \Sigma = P\Sigma P^T \tag{4.22}$$

In Appendix A.2, we proved that reordering didn't change the values of marginal likelihood ratio (4.16) and posterior distribution (4.20). So, the discussion of computational complexity can be carried on their reordering expressions.

$$\begin{aligned} \frac{p(R_P|D_P^{i+1})}{p(R_P|D_P^i)} = & \frac{|Q^{i+1}|^{1/2}}{|Q^i|^{1/2}} \frac{|\underline{Q^i + (D_P^i)^T \Sigma_P^{-1} D_P^i}|^{1/2}}{|\underline{Q^{i+1} + (D_P^{i+1})^T \Sigma_P^{-1} D_P^{i+1}}|^{1/2}} \\ & exp\{\frac{1}{2} R_P^T \Sigma_P^{-1} [D_P^{i+1} (\underline{Q^{i+1} + (D_P^{i+1})^T \Sigma_P^{-1} D_P^{i+1}})^{-1} (D_P^{i+1})^T \\ & - D_P^i (\underline{Q^i + (D_P^i)^T \Sigma_P^{-1} D_P^i})^{-1} (D_P^i)^T] \Sigma_P^{-1} R_P\} \end{aligned} \tag{4.23}$$

and,

$$p(\mu_P|R_P) \sim N((\underline{Q + D_P^T \Sigma_P^{-1} D_P})^{-1} D_P^T \Sigma_P^{-1} R_P, (\underline{Q + D_P^T \Sigma_P^{-1} D_P})^{-1}) \tag{4.24}$$

For the new BART, we assume the precision matrix Σ^{-1} and Σ_P^{-1} are known. The possible computation burden comes from the underline item in (4.23) and (4.24).

$$Q + D_P^T \Sigma_P^{-1} D_P \tag{4.25}$$

In Appendix A.1.1, we show (4.25) is a $b \times b$ symmetric matrix and its calculation is the sum of all non-zero entries in Σ_P^{-1}. In (4.23) and (4.24), we need to calculate $|A|$ and $(A)^{-1}$. They need $O(b^3)$ operations, b is the number of bottom nodes. Fortunately, in the new BART, the size of tree which is the number of bottom nodes are small (usually less than 20). So, if the number of nonzero entries in Σ_P^{-1} is $O(n)$, the MCMC updating of single tree needs $O(n)$ operations. The details of calculating (4.23) and (4.24) can be found in Appendix A.3. However, we have to compute Σ^{-1} for back comparing and buildup tuning range in section 4.4.4. A sparse approximation approach is adopted and introduced in Appendix B.

4.2.5 Example

In order to compare the new and old BART, we make an example to demonstrate their similarities and differences. The simulation data was created as follows.

$$y_i = f(x_i) + \eta_i \qquad i \in \{1, ..., n\}$$

where, $f(x_i) = x_i^3$, $x_i \in (-1, 1)$; $n = 200$. We assumed the error term η_i followed a normal distribution $\eta_i \sim N(0, \Sigma)$. There are two scenarios about the structure of the error term.

(1) η_i are independent and identically distributed (i.i.d.)

In this scenario, $\Sigma = \sigma^2 I$, and the new BART should be identical to the old BART. Figure 4.2 (left) proved this claim.

(2) η_i are correlated

In this scenario, η_i was created as follows.

$$\epsilon_i \sim N(0, \sigma^2), \quad i = 1, ..., n$$

$$\eta_1 = \epsilon_1, \quad \eta_j = \rho\epsilon_{j-1} + \epsilon_j, \quad 0 < \rho < 1, \quad j = 2, ..., n$$

We can denote η_i in a matrix form.

$$\boldsymbol{\eta} = A\boldsymbol{\epsilon}$$

where

$$\boldsymbol{\eta} = \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_n \end{bmatrix}, \quad A = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ \rho & 1 & 0 & \dots & 0 & 0 \\ 0 & \rho & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & \rho & 1 \end{bmatrix}_{n\times n}, \quad \boldsymbol{\epsilon} = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_n \end{bmatrix} \tag{4.26}$$

The inverse of Σ can be calculate by

$$\Sigma^{-1} = \sigma^{-2} A^{-T} A^{-1}$$

Let $\sigma = 0.1$ and $\rho = 0.8$, we examined the new and old BART with above two settings of error term. Figure 4.2 shows the results. When the errors are i.i.d. the two BART models are consistent. But, when the errors are correlated the two models are different from each other. In Table 4.1, we measured the differences between the two models. Compare to the old BART, the new BART fitted bad to the training data but outperformed in restoring the function $f(s)$. It means if the correlated structure of errors is known the new BART can correct its fit to the real signal $f(x)$ rather than the noise according the information getting from covariance matrix Σ.

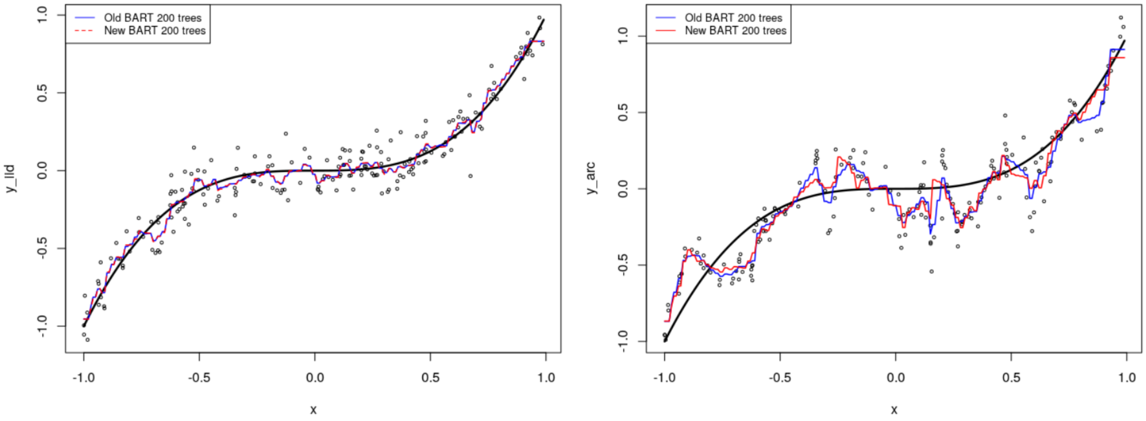


Figure 4.2: Left figure shows if the errors were i.i.d. the new BART degenerated to the old BART. Right figure shows the new BART was different from the old BART when the errors were correlated.

Table 4.1: Comparing $BART_{new}$ and $BART_{old}$

Model	Training Data MSE	Training Data R^2	Restoring $f(x)$ MSE
$BART_{old}$	0.008474055	94.0%	0.01849994
$BART_{new}$	0.02189554	84.5%	0.01547938
$\frac{BART_{new}-BART_{old}}{BART_{old}}$	158.4%	-10.1%	-16.3%

4.3 Gaussian Process Bart Model

In the new BART model (4.3), the correlation structure of error term is arbitrary because Σ is a general covariance matrix. However, in real world applications, people always assumes the error term is satisfied some special correlation structure. There are different ways to model it. In spatial statistics, as discussed in section 1.2, one of the most popular ways is using the Gaussian process. So, by combining Gaussian process and the new BART we proposed a new nonlinear spatial regression model (4.27) which is named Gaussian process BART.

$$y(s_i) = f_{BART}(X(s_i)) + w(s_i) + \epsilon(s_i) \tag{4.27}$$

where

- $y(s_i)$ denotes the response variable observed at location s_i, $i = 1, 2, ..., n$.
- $X(s_i)$ are covariates observed at s_i.
- $f_{BART}(X(s_i))$ is the mean spatial trend function of $X(s_i)$.
- $w(s_i)$ is a Gaussian process modeling the effect of unobserved covariates.
- $\epsilon(s_i) \overset{\text{i.i.d.}}{\sim} N(0, \tau^2)$ denotes the independent and identically distributed noise.

In Chapter 1, we introduced the two categories of spatial regression models. The nonlinear spatial regression models don't include the term $w(s)$, which causes two problems. One is that the models can't take into account the latent covariates. Another is that the models overfit the effects of observed covariates. On the other hand, the spatial linear mixed models can capture the latent covariates' effects with $w(s)$ but they can't model the nonlinear effects of the observed covariates. Gaussian process BART model (4.27) is the first spatial regression model that is able to handle the nonlinear effects of both observed and latent covariates.

4.3.1 Analysis of Variation

In this section, we proposed a method, called analysis of variation, to gain a deep level understanding about the Gaussian process BART model (4.27). Figure 4.3 illustrates the idea of analysis of variation. The discussion can be divided into three parts.

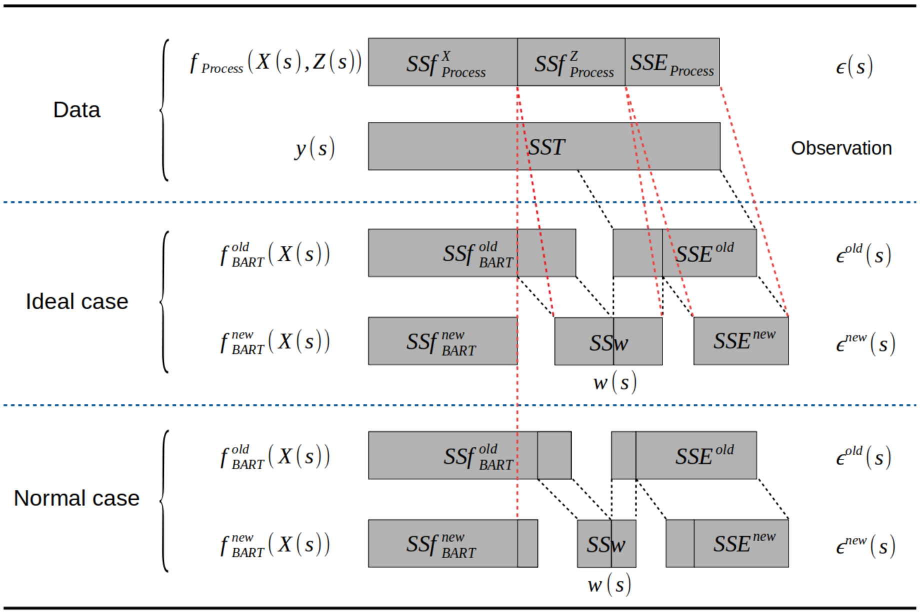


Figure 4.3: Analysis of Variation

First, the data generating process. From a physics point of view, observed data or observations are generated by the underlying physical process and plus the pure error. We call the underlying physical process data generating process. Based on this idea, the total variations in the observations can be divided into two parts, the variations explained by data generating process and the variations of pure error. In Figure 4.3, we denote the variations with sum of square errors. The data generating process $f_{Process}(X(s), Z(s))$ can include observed covariates $X(s)$ and latent covariates $Z(s)$.

So, the total variations in data can be divided into three parts, $SSf^X_{process}$, $SSf^Z_{process}$ and $SSE_{process}$.

Second, the ideal case. Figure 4.3 shows the ideal case that the Gaussian process BART model (4.27) can perfectly explain the three parts variations in the data, $f^{new}_{BART}(X(s))$ catching $SSf^X_{process}$, $w(s)$ catching $SSf^Z_{process}$ and $SSE_{process}$ going into $\epsilon^{new}(s)$. It also indicates that the nonlinear models without $w(s)$, like the old BART model, will overfit the observed covariates process $f_{process}(X(s))$. Because $f^{old}_{BART}(X(s))$ will fit some variations belonging to the latent covariates process $f_{process}(Z(s))$ $(SSf^Z_{process})$.

Third, the normal case. In practice the ideal case rarely happens. The reasons may include, the new BART is still overfitting, the effects of latent covariates doesn't behave as spatial dependence, the assumption (stationary, isotropy) or parameter setting of the Gaussian process $w(s)$ is not suitable to the real data ,etc. However, comparing to the old BART, if $w(s)$ with its explained variations SSw can shrink the variations SSf^{new}_{BART} and SSE^{new} , the existing of $w(s)$ is preferable. The shrinkage of SSf^{new}_{BART} can reduce overfitting of the new BART and restore more close to the real underlying process of observed covariates $X(s)$. The example in section 4.2.5 supports this claim.

4.3.2 The Failure of Likelihood Based MCMC

At first glance, Gibbs sampling is a good choice to estimate f_{BART} and the parameters $\boldsymbol{\theta}$ in model (4.27). $\boldsymbol{\theta}$ includes parameters in the covariance function of Gaussian process $w(s)$ and τ^2. (4.28) and (4.29) are the two steps in MCMC updating.

$$\boldsymbol{\theta} \mid f_{BART} \tag{4.28}$$

$$f_{BART} \mid \Sigma^{-1}(\boldsymbol{\theta}) \tag{4.29}$$

First, given f_{BART}, model (4.27) can be convert to a Bayesian hierarchical model (4.30).

$$p(\boldsymbol{\theta}|y) \propto p(\boldsymbol{\theta}) \times N(w(s)|0, \boldsymbol{C}) \times N(y|f_{BART} + w(s), \tau^2 \boldsymbol{I}) \tag{4.30}$$

Furthermore, the Gaussian process $w(s)$ can be integrated out.

$$p(\boldsymbol{\theta}|y) \propto p(\boldsymbol{\theta}) \times N(y|f_{BART}, \boldsymbol{C} + \tau^2 \boldsymbol{I})$$

Let $\Sigma = \boldsymbol{C} + \tau^2 \boldsymbol{I}$, we can get the posterior distribution $p(\boldsymbol{\theta}|y)$ (4.31) which can be used to update (4.28).

$$p(\boldsymbol{\theta}|y) \propto p(\boldsymbol{\theta}) \times \frac{1}{\sqrt{|\Sigma|}} exp\{-\frac{1}{2}(y - f_{BART})^T \Sigma^{-1} (y - f_{BART})\} \tag{4.31}$$

Second, if parameters $\boldsymbol{\theta}$ are known, the precision matrix Σ^{-1} could be calculated as well. The problem of updating (4.29) given Σ^{-1} was already solved in section 4.2.

Everything looks good so far. However, the devil is in the detail. Let's look at an experiment first. The simulation data is created as follows.

$$f(x(s_i)) = x(s_i)^3, \quad x(s_i) \sim unif(1,3), \quad i = 1, ..., n$$

$$w(s) \sim GP(0, C(\cdot)), \quad C(|s_i - s_j|) = \sigma^2 exp\{-\phi|s_i - s_j|\} \tag{4.32}$$

$$\epsilon(s_i) \overset{\text{i.i.d.}}{\sim} N(0, \tau^2)$$

where $\sigma = 1$, $\tau = 1$ and $\phi = 6$. $x(s_i) \sim unif(1,3)$ denotes that $x(s_i)$ follows a uniform distribution in the range $(1, 3)$.

The MCMC samples of τ^2, σ^2 and ϕ is showed in Figure 4.4 (a). The Markov chain couldn't burn into stationary. If we fixed one of the parameters, $\tau = 1$, then the chain achieved stationary in Figure 4.4 (b). But, in this case, the estimated parameter ϕ is big (≈ 150) which makes the covariance matrix $\Sigma = \sigma^2 I$. So, the new BART model degenerated to the old BART model.

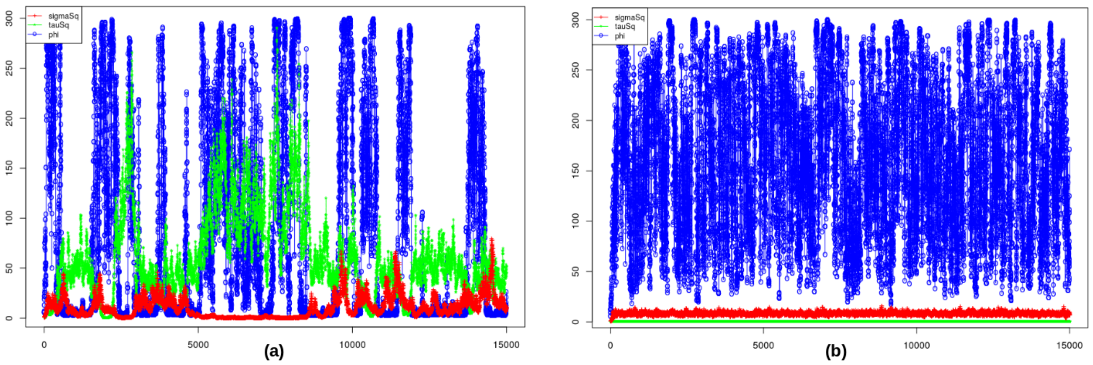


Figure 4.4: The failure of Likelihood based MCMC

For the problem shown in Figure 4.4 (a), the reason is because both BART and Gaussian process are nonparameter and nonlinear model. They are sensitive to the changes of data. An disturbance in the data may cause dramatic turbulence in the Markov chain. When we fixed one or several parameters, this problem may be solved, like the case in Figure 4.4 (b). However, the degeneration issue comes out. To explain this issue, let's recall the example in section 4.2.5. Table 4.1 tell us that working with the true parameter ρ and Σ the new BART fitted bad to the training data. Actually, the fitting will get worse when $|\rho|$ approaches to 1. Suppose we know nothing about the parameters of Σ and all the information comes from the data which determines the likelihood. In likelihood based MCMC, the likelihood will guide its searching behavior in parameter space. As a result, the data/likelihood will lead the parameter ρ going to zero. In other word, the correlation structure of Σ will be eliminated and Σ will degenerate to $\sigma^2 I$. The same thing happens in spatial context. If there is no any prior information about the parameters of the covariance function $C(\cdot)$, the data/likelihood will lead MCMC searching to eliminate the spatial dependent structure in $C(\cdot)$ and makes $\Sigma = \sigma^2 I$. Figure 4.4 (b) just showed this situation. We want to estimate the parameters which must be known first. It looks like we are locked in a dead loop. In next section, we will introduce a key to open this lock, which is called back comparing.

4.3.3 Back Comparing and Tuning Range

In section 4.3.2, we discussed the failure of likelihood based MCMC. The reason of failure is because the data leads the search in parameter space and tend to eliminate the correlation structure. So, the solution should pull the parameter search in the opposite direction. Instead let the data totally control the parameter searching process, we need to intervene it by proposing candidates that scatter over a larger range in parameter space. We proposed a strategy, back comparing, to select the good candidates. Figure 4.5 demonstrates the idea of back comparing. First, we propose an candidate $\boldsymbol{\theta}$. Second, use this candidate to fit the new BART model. With fitted BART model f_{BART}^{new}, the variation of residuals SSE_{real}^{new} can be calculated. Meanwhile, with the value of candidate $\boldsymbol{\theta}$, it's easy to calculate the proposed variation of the mixed errors $SSME_{proposed}^{new}$ which includes the errors comes from $w(s)$ and $\epsilon(s)$. Then, we compare SSE_{real}^{new} and $SSME_{proposed}^{new}$. There are three possible cases.

(1) $SSE_{real}^{new} < SSME_{proposed}^{new}$

Over-estimation. The proposed $\boldsymbol{\theta}$ estimates more variations ($SSME_{proposed}^{new}$) when it works with the new BART f_{BART}^{new}.

(2) $SSE_{real}^{new} > SSME_{proposed}^{new}$

Under-estimation. The proposed $\boldsymbol{\theta}$ estimates less variations ($SSME_{proposed}^{new}$) when it works with the new BART f_{BART}^{new}.

(3) $SSE_{real}^{new} \approx SSME_{proposed}^{new}$

Good-estimation. The proposed $\boldsymbol{\theta}$ provides good estimation about the variations of $SSME_{proposed}^{new}$ when it works with the new BART f_{BART}^{new}.

Back comparing provides us a criteria to identify the good estimation of parameters. Figure 4.6 illustrates the parameter searching process. After proposed a set

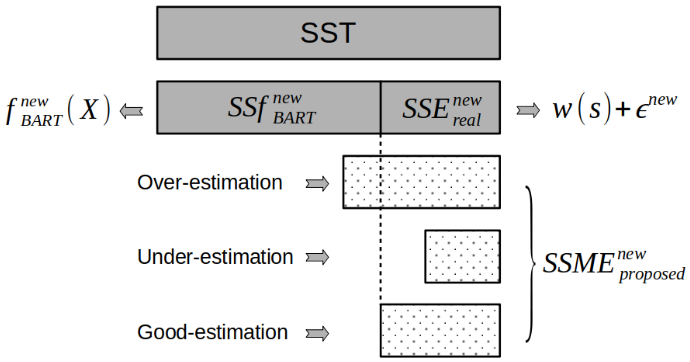


Figure 4.5: Back Comparing

of parameters $\{\boldsymbol{\theta}^{(0)}, ..., \boldsymbol{\theta}^{(n)}\}$ we apply the backing comparing to each of them. The good estimations be picked out and put into a new set, called tuning range. All the proposed parameters in the tuning range are good for both the model (4.27) and the data. People can select the one that fits to their goals best. A more intuitive analogy is the speaker volume control knob. You can tune the knob to get the volume comfortable to you. However, the best volume differs from person to person. Even you will adjust it when the situation changes, for example, the environment changes from quiet to noisy.

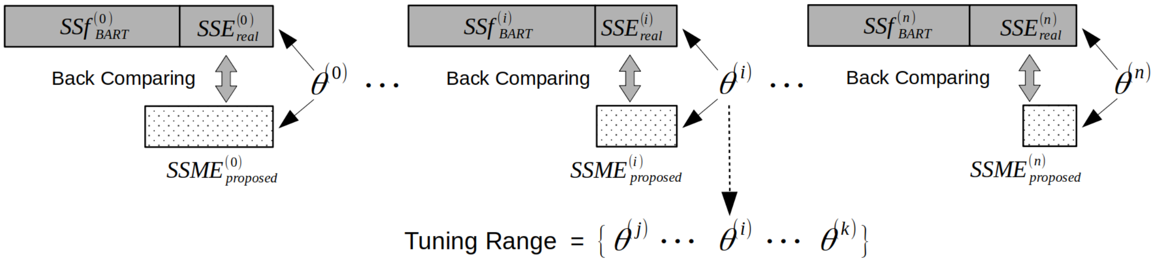


Figure 4.6: Parameter space searching for the buildup of tuning range

There is still a question. How can we properly propose a parameter set $\{\boldsymbol{\theta}^{(0)}, ..., \boldsymbol{\theta}^{(n)}\}$ for searching the tuning range? The answer is that we can use the information getting from the old BART or the liner mixed models. All approaches in this section will be demonstrated with concrete examples in section 4.4.

4.4 Experiments and Results

In this section, we will show the applications of model (4.27) in two type of problems, one dimension problems and two dimension problems. The methods, back comparing and tuning range, will be discussed carefully. The idea of analysis of variation introduced in section 4.3.1 will provide guidance to the parameter selection in tuning range.

4.4.1 One Dimension Experiment

In one dimension, the class of autoregressive (AR) processes, and its extensions, autoregressive moving-average (ARMA) processes are popular choices for modeling time-varying processes. By Wold decomposition theorem, any AR(p) process is a special Guassian process called Gaussian linear process if it satisfies the recursions

$$y_t = \phi_1 y_{t-1} + ... + \phi_p y_{t-p} + \epsilon_t$$

where $\{\epsilon_t\}$ is an i.i.d. sequence of $N(0, \sigma^2)$ random variables, and the polynomial $\phi(z) = 1 - \phi_1 z - ... - \phi_p z^p$ has no zeros inside or on the unit circle (Brockwell and Davis, 2002). It means that the Gaussian process BART model (4.27) can be used in one dimension problems.

Recall the example in section 4.2.5. It's not exactly an AR(1) process but an AR(1) error process. However, it's also a Gaussian linear process. Because the sequence $\boldsymbol{\eta} = \{\eta_t\}$ always follows a multivariate Gaussian distribution.

$$\boldsymbol{\eta} \sim MVN(0, \Sigma), \qquad \Sigma = \sigma^2 A A^T$$

where matrix A was defined in (4.26). In this case, the Gaussian process BART model (4.27) becomes to (4.33). Next, we will treat this model with previously proposed methods.

$$y(t_i) = f_{BART}^{new}(X(t_i)) + \eta(t_i) \quad i = 1, .., n \tag{4.33}$$

Note: In this experiment, the parameters in model (4.33) are $\boldsymbol{\theta} = \{\sigma, \rho\}$. Their real value are $\sigma = 0.1$ and $\rho = 0.8$ (the green point in Figure 4.7).

(1) Back comparing and Tuning range

In order to apply back comparing to find the tuning range, first, we need to propose a searching set $\{\boldsymbol{\theta}^{(\mathbf{0})}, ..., \boldsymbol{\theta}^{(\boldsymbol{n})}\}$ in parameter space. The estimation from old BART model can provide clues, $\hat{\sigma} = 0.1042189$ where is the yellow dash line located in Figure 4.7. We can search σ in the neighbor interval of $\hat{\sigma}$ which is $(0.05, 0.15)$ shown in Figure 4.7. For the parameter ρ, its value must be constrained in $(0, 1)$ to keep the Gaussian process $\eta(t)$ stationary. We divided each interval into 10 segments and selected the centers as the searching set. So, the searching set included 100 candidates $\{\boldsymbol{\theta}^{(\mathbf{0})}, ..., \boldsymbol{\theta}^{(\mathbf{99})}\}$ which is showed in Figure 4.7. As the discussion in section 4.3.3, to apply back comparing we need to compare SSE_{real}^{new} and $SSME_{proposed}^{new}$. Figure 4.7 shows the back comparing results $SSE_{real}^{new} - SSME_{proposed}^{new}$. The cells with negative (positive) value represent over-estimation (nuder-estimation). The tuning range (green cells) was selected under the criteria $|SSE_{real}^{new} - SSME_{proposed}^{new}| < 0.5$. We can change this criteria to control the size of tuning range. In Figure 4.7, the left panel shows the results that was obtained by working on fitted training data. First, the model was fitted with all the observed data. Then, the observed data and their model fitted values were used to calculate the variations, SSE_{real}^{new} and $SSME_{proposed}^{new}$. In this case, the BART model tends to overfit the data and causes the parameters under estimated. To tackle this issue, instead of fitting training data we can use predicted testing data and the observed testing data to compute the variations. The predicted testing data was generated be a 4-folds cross-validation. Figure 4.7 right panel presents the back comparing results and tuning range in this case. Now, you may have a question. Why the real value (green point) of parameters is not included in the tuning range? Let's look at the results of back comparing. Compared to the real value, all the values in tuning range are under estimated which indicates their corresponding variations SSw (see Figure 4.3) are less than the real value case. Recall the analysis of variation and Figure 4.3. The real value corresponds to the ideal case (Figure 4.3) which rarely happen. While, the values in tuning range correspond to the normal case (Figure 4.3). Last but not least, the number of folds in cross-validation should not be too small (less than 4) to damage the correlation structure of covariance matrix Σ.

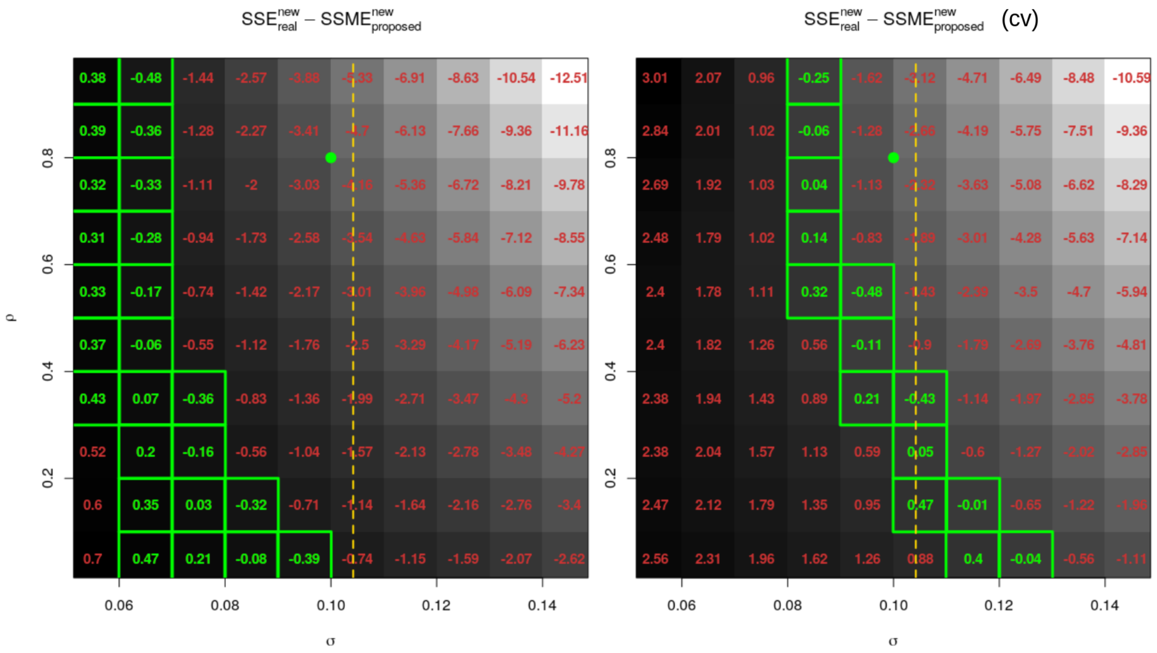


Figure 4.7: Back comparing and Tuning range. The green dot is the real value of parameters. The vertical yellow dash line shows the estimation of σ from the old BART. The green cells indicate the tuning range. Left panel shows the tuning range that selected using the fitted training data. While, the tuning range in right panel was using the predicted testing data in a 4-folds cross-validation.

(2) Guidance of parameter selection in tuning range

As discussed in Section 4.3.3, any candidate in tuning range is good for the model and data but maybe not for your purpose. Besides personal purpose, there is a guidance for selecting the parameter values in tuning range according the idea of analysis of variation in Section 4.3.1. In Figure 4.3, compare to the old BART model the more SSf_{BART}^{new} and SSE^{new} be shrink, the better the Gaussian process BART model (4.27) performs in both interpretation and prediction. Figure 4.8 shows the values $SSf_{BART}^{old} - SSf_{BART}^{new}$ and $SSE^{old} - SSE^{new}$. So, under the guidance of analysis of variation we should select the big values of these two subtractions which indicates the preference of correlation structure (big ρ).

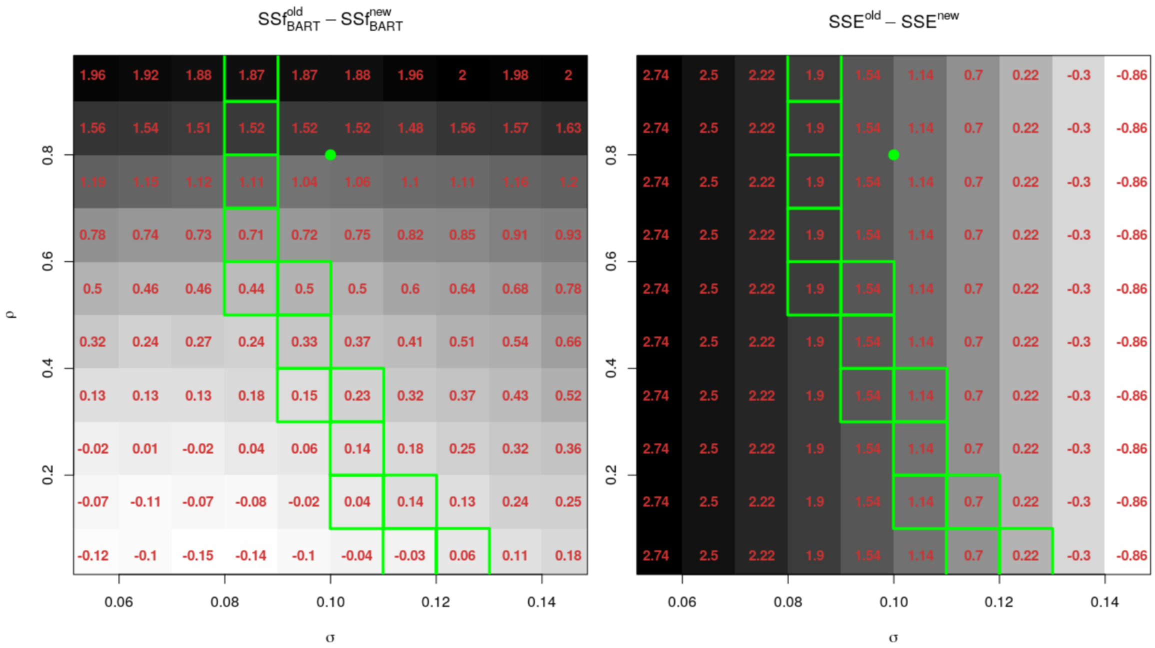


Figure 4.8: Guidance of parameter selection in tuning range

Moreover, with the proposed parameter values we can decompose the variations in the Gaussian process $\eta(t)$ into pure error variation $SSE_{proposed}^{new}$ and correlated variation $SSw_{proposed}^{new}$.

$$SSE_{proposed}^{new} = n\sigma^2, \quad SSw_{proposed}^{new} = SSME_{proposed}^{new} - SSE_{proposed}^{new}$$

where $SSME_{proposed}^{new}$ is in Figure 4.5. Their values was showed in Figure 4.9. The analysis of variation (Figure 4.3) suggests to select big value of $SSw_{proposed}^{new}$ and small value of $SSE_{proposed}^{new}$. It is consistent to the previous guidance.

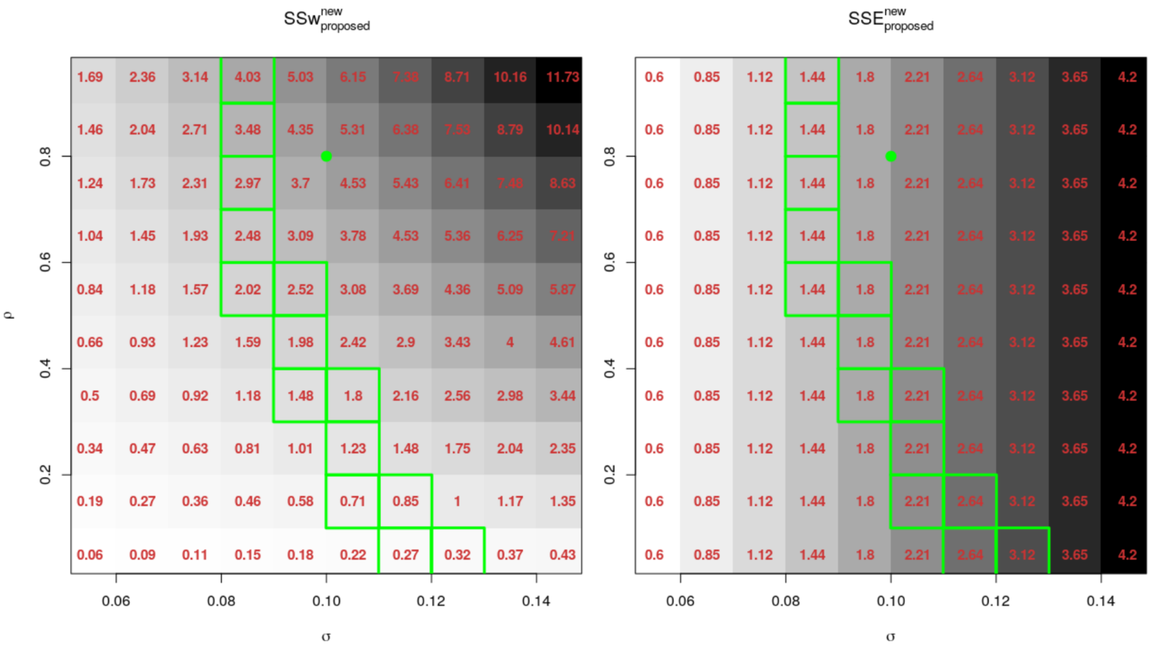


Figure 4.9: The variance decomposition of Gaussian process $\eta(t)$.

(3) Results

The analysis of variation recommended $\boldsymbol{\theta}_{top} = \{\sigma, \rho\} = \{0.085, 0.95\}$ at the top of tuning range. To study the effect of different values in tuning range we selected another one $\boldsymbol{\theta}_{bottom} = \{\sigma, \rho\} = \{0.125, 0.05\}$ at the bottom of tuning range. Their comparison in Figure 4.10 shows the significant differences. In Figure 4.11, we compared the top one with the real value (left) and the bottom one with the old BART (right). Although the top one $\{\sigma, \rho\} = \{0.085, 0.95\}$ is different from the real value $\{\sigma, \rho\} = \{0.1, 0.8\}$, their fits are quite similar. It indicates that the guidance from analysis of variation is effective and can lead us closing to the real value. On the other hand, the fit of bottom one $\{\sigma, \rho\} = \{0.125, 0.05\}$ is very close to the fit of old BART $\{\sigma, \rho\} = \{0.1042189, 0\}$. Based on the comparisons, it's easy to imagine that if we scan the tuning range from top to bottom the model fitting will degenerate from the new BART with (near) real correlation structure to the (near) old BART. In other word, the covariance matrix Σ will approximately degenerate to $\sigma^2 I$.

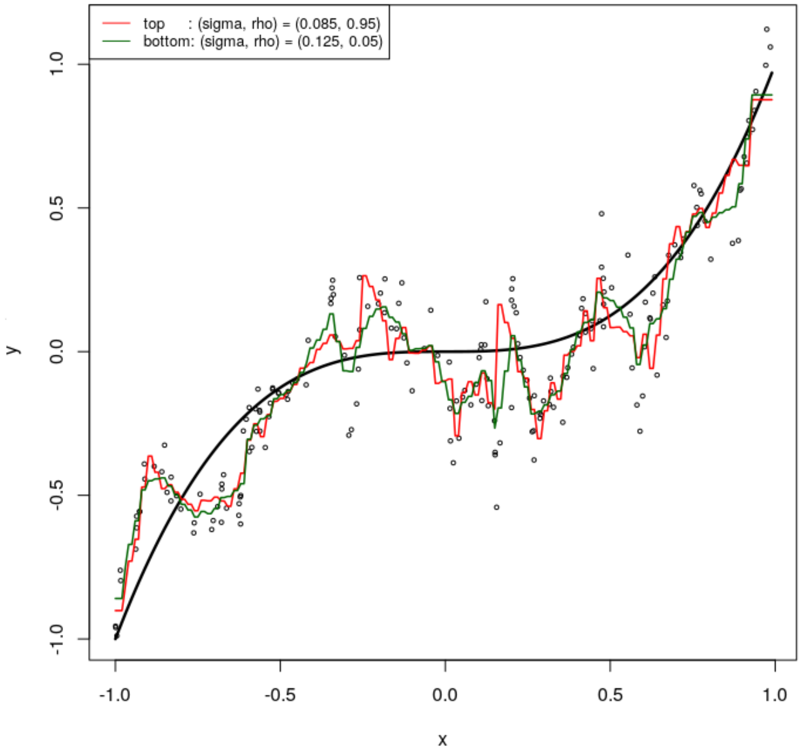


Figure 4.10: Two extreme candidates from the tuning range. The comparison shows they impose different influences on the model fitting.

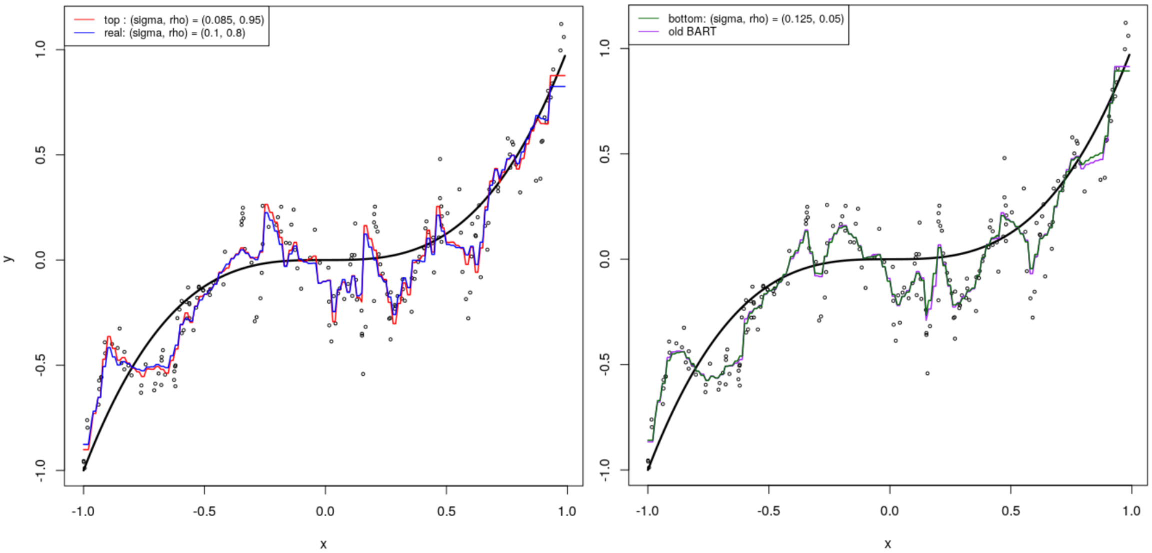


Figure 4.11: Comparing the two extreme candidates in tuning range to the real value and old BART. Left panel shows the similarity of model fitting between the candidate at the top of tuning range and the real value. Right panel shows the similarity of model fitting between the candidate at the bottom of tuning range and the old BART

4.4.2 Two Dimension Experiment

The two dimension experiment is created following the Gaussian BART model (4.27).

$$y(s_i) = f(x(s_i)) + w(s_i) + \epsilon(s_i), \quad i \in \{1, ..., 400\} \tag{4.34}$$

where

- $f(x(s_i)) = x(s_i)^3, \quad x(s_i) \sim unif(1, 3)$.
- $w(s_i) \sim GP(0, C(\cdot, \cdot|\sigma, \phi)), \quad C(s_j, s_k|\sigma, \phi) = \sigma^2 exp\{-\phi * d(s_j, s_k)\}$,

  where $d(s_j, s_k)$ is the Euclidean distance between point s_j and s_k.
- $\epsilon(s_i) \overset{\text{i.i.d}}{\sim} N(0, \tau^2)$.
- The real value of parameters are $\{\sigma, \phi, \tau\} = \{1, 6, 1\}$.

We can explore the created data in Figure 4.12. Similar to the one dimension experiment, we will check this experiment in three parts.

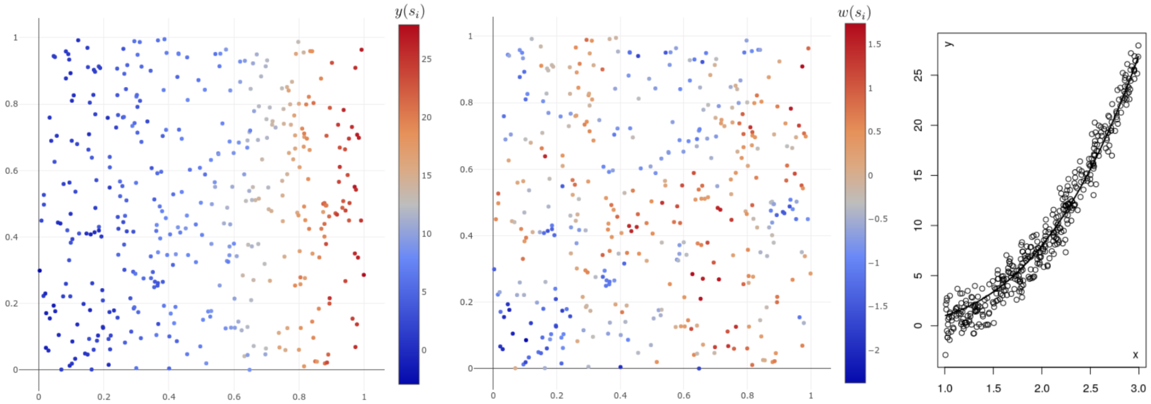


Figure 4.12: Experiment data exploration. Left and middle panels show the spatial maps of $y(s_i)$ and $w(s_i)$ respectively. Right panel shows relation between x and y.

(1) Back comparing and Tuning range

First, we need to propose the searching set in parameter space. The liner mixed model regression Kriging (4.35) can provide clues.

$$y(s_i) = \beta_0 + x(s_i)\beta_1 + w(s_i) + \epsilon(s_i) \tag{4.35}$$

The parameters estimated by (4.35) are $\{\hat{\sigma}, \hat{\phi}, \hat{\tau}\} = \{1.18, 5.46, 1.05\}$. According these estimations, we created the searching set as follows. 10 equally divide the interval (0.5,1.4) for σ; 10 equally divide the interval (1,10) for ϕ; 7 equally divide the interval (0.4,1.6) for τ. Then, back comparing was applied to build the tuning range. Figure 4.13 (left) shows the back comparing results and selected tuning range. In this experiment, instead of sum square error (SSE) the mean square error (MSE) was used to avoid large values. Figure 4.13 (right) indicates that we took on strict criteria, $|MSE^{new}_{real} - MSME^{new}_{proposed}| < 1$, to select the tuning range. The exact values of back comparing are listed in Table 4.2.

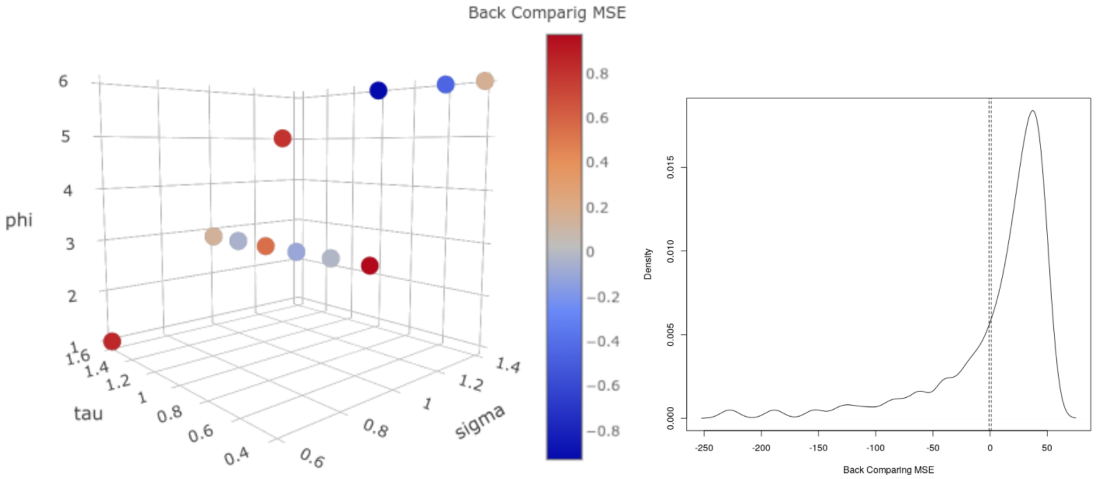


Figure 4.13: Back Comparing and Tuning Range (left). Back comparing MSE density and selection criteria (right).

(2) Guidance of parameter selection in tuning range

As discussed in the one dimension experiment, $MSf^{old}_{BART} - MSf^{new}_{BART}$ and $MSE^{old} - MSE^{new}$ can be used as guidance to select the good candidates in tuning range. Fig-

Table 4.2: Back Comparing and Candidates Selection Guidance

τ	σ	ϕ	Back Comparing MSE	$MSf^{old}_{BART} - MSf^{new}_{BART}$	$MSE^{old} - MSE^{new}$
0.4	0.9	3	0.97500489	50.90653	1.11422824
0.4	1.4	6	0.16215768	50.29464	1.11422824
0.6	0.9	3	-0.02155375	50.02426	0.99994253
0.6	1.4	6	-0.45026158	49.79650	0.99994253
0.8	0.9	3	-0.10393848	50.10187	0.83994253
1.0	0.9	3	0.54719834	50.95872	0.63422824
1.0	1.4	6	-0.92835327	49.68413	0.63422824
1.2	0.9	3	-0.04935067	50.61360	0.38279967
1.4	0.9	3	0.14343539	51.10353	0.08565681
1.4	1.2	5	0.80226033	49.45199	0.08565681
1.6	0.6	1	0.85251483	51.78393	-0.25720033

ure 4.14 illustrates their relative values (color) and positions in tuning range. Their exact value are listed in Table 4.2. According the first guidance, $MSf^{old}_{BART} - MSf^{new}_{BART}$, the candidates $\{\tau, \sigma, \phi\} = \{1.6, 0.6, 1\}$, $\{1.4, 0.9, 3\}$, $\{1, 0.9, 3\}$ are the top three candidates. But, when we check with the second guidance, $MSE^{old} - MSE^{new}$, it gives the opposite order, and the value is negative for the candidate $\{1.6, 0.6, 1\}$. In this case, $\{1.4, 0.9, 3\}$ and $\{1, 0.9, 3\}$ are both good. I choose the second one $\{\tau, \sigma, \phi\} = \{1, 0.9, 3\}$ because the sum of subtractions, $MSf^{old}_{BART} - MSf^{new}_{BART} + MSE^{old} - MSE^{new}$,is bigger than the first one's.

(3) Results

The motivation of developing the Gaussian process BART model (4.27) is trying to gain the advantages of both the spatial linear mixed regression models and the spatial nonlinear regression models. On one hand, compare to the linear mixed regression models, Gaussian process BART model should be capable to handle the nonlinear relationships between the observed variables $y(s)$ and $x(s)$. On the other

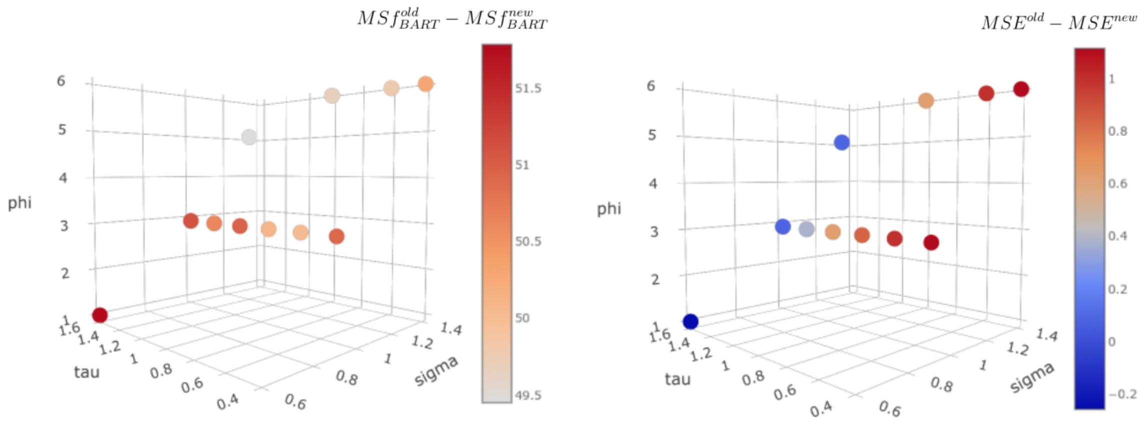


Figure 4.14: Left shows the values of $MSf_{BART}^{old} - MSf_{BART}^{new}$ in tuning range. Right shows the values of $MSE^{old} - MSE^{new}$ in tuning range.

hand, compare to the nonlinear regression models, Gaussian process BART model should be able to understand the spatial dependence which may be caused by the latent variables. Obviously, we already achieved the second goal. For the first goal, let's compare the results between Gaussian process BART model (4.34) and linear mixed model (4.35).

- First, they have similar ability to understand the spatial dependent structure in the data. It is because their estimated parameters are both close to the real value.

- Second, they have different ability to understand the relationship between $y(s)$ and $x(s)$. Figure 4.15 demonstrates the differences. obviously, the Gaussian process BART model greatly captured the nonlinear relation $f(x) = x^3$ between $y(s)$ and $x(s)$.

- Last but not least, the failure of fitting nonlinear trend may cause the linear mixed model violates its assumption that the Gaussian process $w(s)$ is stationary. In this experiment, the linear mixed model failed to extract the nonlinear trend $f(x) = x^3$. So, the stationary assumption must be violated. Actually, there are many literature working on this problem. They were trying to estimate a non-stationary Gaussian process using methods like, spatial partitioning, process convolution, low rank splines or basis functions, etc. While, the Gaussian process BART model (4.27) which is able to capture both linear and nonlinear trend naturally makes the stationary assumption much more robust than it in the linear mixed model.

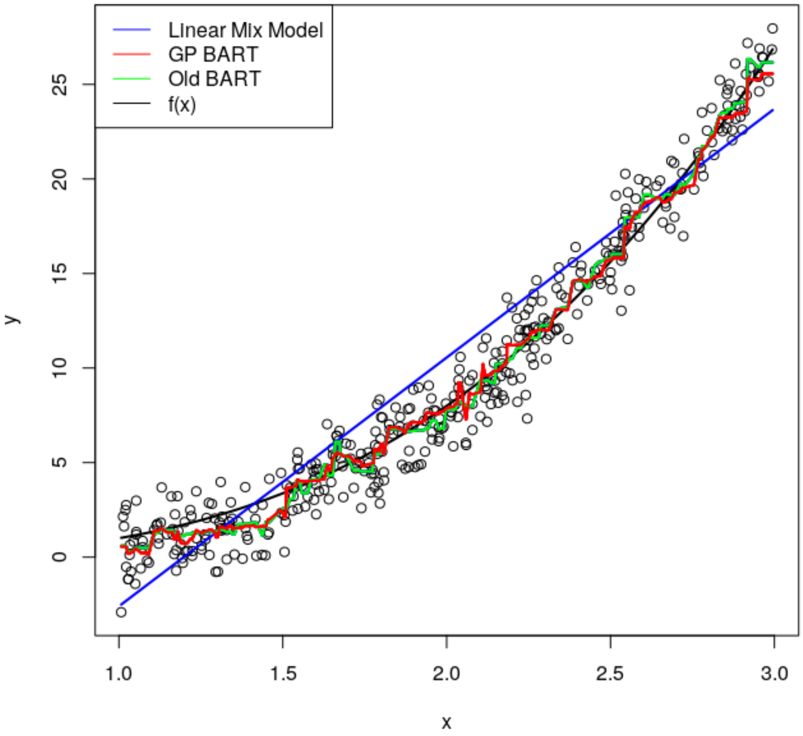


Figure 4.15: The fitting results of Gaussian process BART, old BART and Linear mixed model

Moreover, we can compare the Gaussian process BART to old BART. In Figure 4.15, they look similar but still have some differences. It is because the Gaussian process BART foresees the spatial dependence when it fits the data. Like the example in Section 4.2.5, the Gaussian process BART model should perform better in fitting the underlying process $f(x)$ than the old BART. To illustrate that, we can calculate the mean square errors (MSE) as follows.

$$MSE_f^{old} = \frac{\sum_{i=1}^{n}(f(x_i) - \hat{f}_{old}(x_i))^2}{n}, \quad MSE_f^{GP} = \frac{\sum_{i=1}^{n}(f(x_i) - \hat{f}_{GP}(x_i))^2}{n}$$

From the experiment, we got $MSE_f^{old} = 0.3594093$ and $MSE_f^{GP} = 0.3418461$. This result proves the claim that Gaussian process BART performs better in restoring the underlying process $f(x)$ than the old BART.

4.4.3 Testing on Real Data

In this section, we test Gaussian process BART on real data which is the soil carbon stock data in Chapter 3. In order to visually compare the results among different models, we chose two environmental covariates to do the test. From the results in Chapter 3, we know the environmental covariates NDVI14 and REDL14 have nonlinear relationships with the response variable y (see Figure 3.8). So, we choose them and construct the models as follows.

The linear mixed model:

$$y(s_i) = f_{LMX}(X(s_i)) + w(s_i) + \epsilon(s_i) \tag{4.36}$$

The Gaussian process BART model:

$$y(s_i) = f_{GPBART}(X(s_i)) + w(s_i) + \epsilon(s_i) \tag{4.37}$$

The old BART model:

$$y(s_i) = f_{BART}(X(s_i)) + \epsilon(s_i) \tag{4.38}$$

where

- $s_i, \;\; i = 1, 2, ..., 6213$, there are 6213 observations in different locations.

- We test two scenarios, one is using variable NDVI14 only, another is using NDVI14 and REDL14 two variables. So, in the first scenario,

  $$X(s_i) = \{x_{NDVI14}(s_i)\}$$

  In the second scenario,

  $$X(s_i) = \{x_{NDVI14}(s_i), x_{REDL14}(s_i)\}$$

- In the first scenario, $f_{LMX}(X(s_i))$ in the linear mixed model (4.36) is:

  $$f_{LMX}(X(s_i)) = \beta_0 + x_{NDVI14}(s_i) * \beta_1$$

  In the second scenario, $f_{LMX}(X(s_i))$ in the linear mixed model (4.36) is:

  $$f_{LMX}(X(s_i)) = \beta_0 + x_{NDVI14}(s_i) * \beta_1 + x_{REDL14}(s_i) * \beta_2$$

- $w(s_i) \sim GP(0, C(\cdot, \cdot|\sigma, \phi)), \quad C(s_j, s_k|\sigma, \phi) = \sigma^2 exp\{-\phi * d(s_j, s_k)\},$

  where $d(s_j, s_k)$ is the Euclidean distance between point s_j and s_k. We use the R package ”fields” (Nychka *et al.*, 2017) to check this model. In that package the parameter ϕ is set to 1 defaultly. So, only the unknown parameter σ will be estimated.

- $\epsilon(s_i) \overset{\text{i.i.d}}{\sim} N(0, \tau^2)$. The unknown parameter τ will be estimated.

The results will be presented in two scenarios as well.

(1) The first scenario (NDVI14)

The estimations of the linear mixed model (4.36) shows in Table 4.3. We use the estimations σ and τ to fit the Gaussian process BART model (4.37). Figure 4.16 shows the fitting results of these 3 models. We can see the differences among them. Since the Gaussian process BART model (4.37) used the same value of covariance

Table 4.3: Linear Mixed Model Estimations (the First Scenario)

β_0	β_1	σ	τ
4.0475534	0.4237016	0.5282045	0.6557

parameters with the linear mixed model (4.36), the fitting of Gaussian process BART shrinks more to linear mixed model comparing to the old BART. Meanwhile, the Gaussian process BART keeps its non-linearity comparing to the linear mixed model.

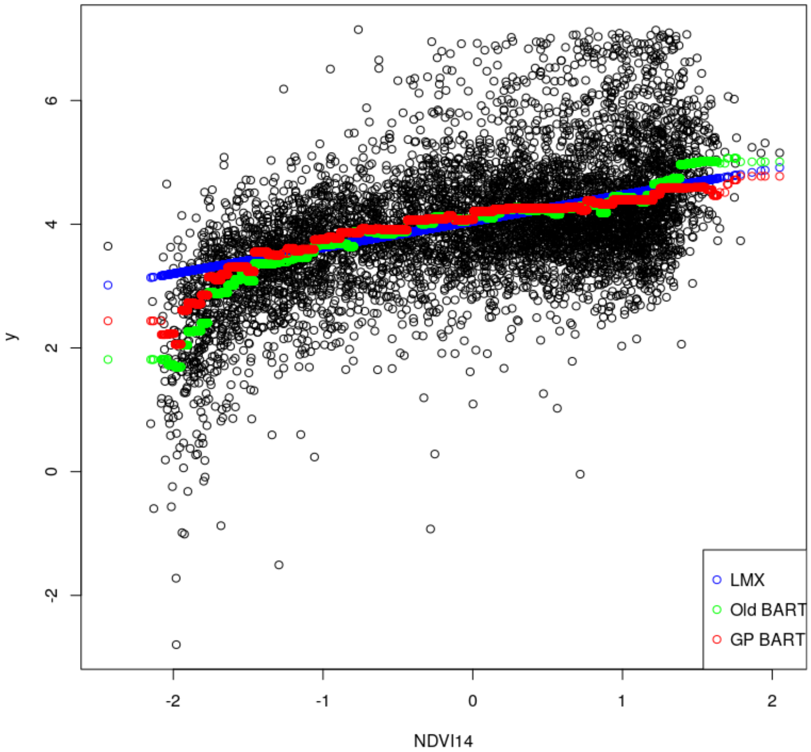


Figure 4.16: The fitting results of Gaussian process BART, old BART and Linear mixed model on real data with one covariate NDVI14.

(2) The second scenario (NDVI14 and REDL14)

The real soil carbon stock data and two environmental covariates NDVI14 and REDL14 are showed in Figure 4.17.

The estimations of the linear mixed model (4.36) shows in Table 4.4. Similar to the first scenario, we use the estimations σ and τ to fit the Gaussian process BART

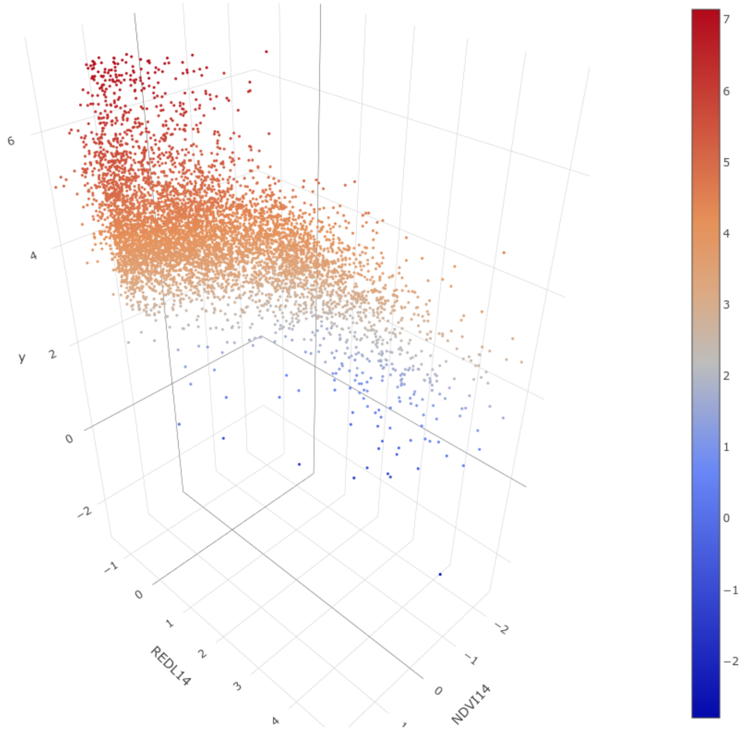


Figure 4.17: The read data with two environmental covariates NDVI14 and REDL14.

model (4.37).

Table 4.4: Linear Mixed Model Estimations (the Second Scenario)

β_0	β_1	β_2	σ	τ
4.0413586	0.163074	-0.2934858	0.5040833	0.6539

Figure 4.18 illustrates the different model fittings. Comparing to the linear mixed model, both Gaussian process BART and old BART successfully captured the non-linear relationships between the covariates and response variables.

Figure 4.19 shows the differences among the three models. Similar to the first scenario, the Gaussian process BART shrinks more to linear mixed model comparing to the old BART. It's because that the Gaussian process BART model (4.37) used the same value of covariance parameters with the linear mixed model (4.36).

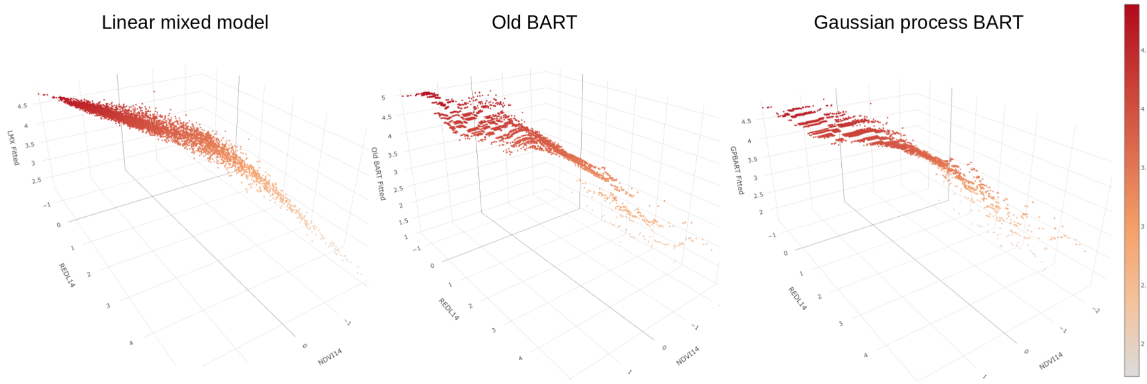


Figure 4.18: The model fittings on read data.

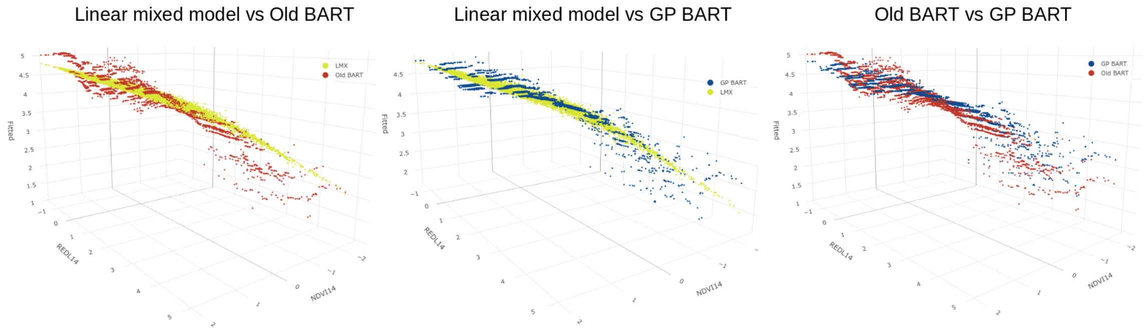


Figure 4.19: The results comparison among different models.

4.4.4 Discussion on Computation Issues

All the computation issues with Gaussian process BART model (4.27) are related to parameter space searching for the buildup of tuning range. The issues and possible solutions are discussed in this section.

(1) The curse of dimensionality

As showing in the experiments, to construct the tuning range we have to propose a searching set in parameter space. This searching set suffers from the curse of dimensionality as the parameters increasing. The first possible solution is using low dimensional parametric Gaussian process models, for example, the Matérn family. The second possible solution is using random search to instead grid search. The ad hoc information of random search can be used to locate the promising areas in parameter space. Another possible solution is parallel computing. In theory, all the points in searching set can be test in parallel. Since the computation resources is limited, we can partition the searching set into subsets and deploy different computational resources to each of them.

(2) The inverse of covariance matrix

For every time searching, we have to inverse the covariance matrix Σ with the proposed parameters in searching set. It's because the algorithm of the new BART needs Σ^{-1} rather than Σ (see Appendix A). Since the dimension of Σ is $n \times n$ where n is the number of observations, as the data increasing, the calculation of Σ^{-1} will become the computational bottleneck of model (4.27). A possible solution is creating a sparse matrix which has $O(n)$ non-zero entries to approximate Σ^{-1}. A popular approach called nearest neighbor Gaussian process (Finley *et al.*, 2019) is presented in Appendix B.

Chapter 5

CONCLUSION

This chapter summarizes the key ideas and contributions of the dissertation. Ideas for further research are also discussed.

5.1 Summary of Contributions

- Chapter 1 provided a unifying view of existing models for spatial regression. A classification based on their capability of modeling latent variables was introduced.

- In Chapter 2, a multistage workflow equipped with nonlinear models was proposed for the spatial prediction problem in reef species abundance study. The methods, empirical maximum likelihood analysis and random smoothing, were developed to solve the zero-inflated issue in sampling data. Three strategies, prior knowledge, aggregation and iteration were introduced to help the nonlinear models overcome the out of sample prediction issue.

- Chapter 3 developed a novel two-stage model for the spatial regression problems in soil carbon stock (SOC) analysis. In the first stage, a universal regression Kriging model captures the linear and stationary effects of covariates. The remaining nonlinear and non-stationary effects are modeled by a generalized additive model in the second stage.

- In Chapter 4, the traditional BART model was extended to a new BART model which can accommodate the general correlated errors. A novel nonlinear spatial regression model called Gaussian process BART can then be built by combining the new BART and Gaussian process. Because of the failure of likelihood based MCMC in parameter estimation, the methods, back comparing and tuning range, were proposed based on the idea of analysis of variation.

5.2 Future Work

Promising paths for future work involve:

- Applying the Gaussian process BART model to real world problems.
- Solving the computation issue of parameter space searching for the buildup of tuning range.
- Updating R package "BART" with the new algorithm for accommodating correlated errors.

REFERENCES

Adhikari, K., U. Mishra, P. Owens, Z. Libohova, S. Wills, W. Riley, F. Hoffman and D. Smith, “Importance and strength of environmental controllers of soil organic carbon changes with scale”, Geoderma **375**, 114472 (2020).

Ainsworth, L. M., C. B. Dean and R. Joy, “Zero-inflated spatial models: Application and interpretation”, (2016).

Andrew Zammit-Mangion, “Frk: Fixed rank kriging”, R package version 0.2.2.1 (2020).

Appelhans, T., E. Mwangomo, D. R. Hardy, A. Hemp and T. Nauss, “Evaluating machine learning approaches for the interpolation of monthly air temperature at mt. kilimanjaro, tanzania”, Spatial Statistics **14**, 91–113 (2015).

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

Breiman, L., “Random forests”, **45**, 1, 532 (2001).

Brockwell, P. J. and R. A. Davis, *Introduction to Time Series and Forecasting* (Springer-Verlag New York, 2002).

Brunsdon, C., S. Fotheringham and M. Charlton, “Geographically weighted regression-modelling spatial non-stationarity”, Journal of the Royal Statistical Society. Series D (The Statistician) **47**, 3, 431–443 (1998).

Chipman, H. A., E. I. George and R. E. McCulloch, “Geographically weighted regression-modelling spatial non-stationarity”, The Annals of Applied Statistics **4(1)**, 266–298 (2010).

Coleman, F. C., K. M. Scanlon and C. C. Koenig, “Groupers on the edge: Shelf edge spawning habitat in and around marine reserves of the northeastern gulf of mexico”, The Professional Geographer **63**, 4, 456–474 (2011).

Cressie, N. and G. Johannesson, “Fixed rank kriging for very large spatial data sets”, Journal of the Royal Statistical Society: Series B (Statistical Methodology) **70**, 1, 209–226 (2008).

Cressie, N. A. C., *Statistics for Spatial Data* (John Wiley & Sons, Inc., 1993).

Cristianini, N. and J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods* (Cambridge University Press, 2000).

Datta, A., S. Banerjee, A. O. Finley and A. E. Gelfand, “Hierarchical nearest-neighbor gaussian process models for large geostatistical datasets”, Journal of the American Statistical Association **111**, 514, 800–812 (2016).

Diesing, M. and D. Stephens, “A multi-model ensemble approach to seabed mapping”, Journal of Sea Research **100**, 62 – 69, meshAtlantic: Mapping Atlantic Area Seabed Habitats for Better Marine Management (2015).

Dobesch, H., P. Dumolard and I. Dyras, *Spatial Interpolation for Climate Data: The Use of GIS in Climatology and Meteorology* (ISTE Ltd, 2007).

Drexler, M. and C. H. Ainsworth, “Generalized additive models used to predict species abundance in the gulf of mexico: an ecosystem modeling tool”, PloS one **8**, 5 (2013).

Farmer, N. A. and J. S. Ault, “Grouper and snapper movements and habitat use in dry tortugas, florida”, Mar Ecol Prog Ser **433**, 169–184 (2011).

Farmer, N. A. and J. S. Ault, “Modeling coral reef fish home range movements in dry tortugas, florida”, The Scientific World Journal **2014**, 14 (2014).

Finley, A. O., A. Datta, B. D. Cook, D. C. Morton, H. E. Andersen and S. Banerjee, “Efficient algorithms for bayesian nearest neighbor gaussian processes”, Journal of Computational and Graphical Statistics **28**, 2, 401–414 (2019).

Furrer, R., M. G. Genton and D. Nychka, “Covariance tapering for interpolation of large spatial datasets”, Journal of Computational and Graphical Statistics **15**, 3, 502–523 (2006).

Gelfand, A. E., P. J. Diggle, M. Fuentes and P. Guttorp, *Handbook of Spatial Statistics* (Chapman & Hall/CRC, 2010).

Geoga, C. J., M. Anitescu and M. L. Stein, “Scalable gaussian process computations using hierarchical matrices”, (2019).

Goff, J. A., C. J. Jenkins and S. Williams, “Seabed mapping and characterization of sediment variability using the usseabed data base”, Continental Shelf Research **28**, 4, 614 – 633 (2008).

Guisan, A., T. C. Edwards and T. Hastie, “Generalized linear and generalized additive models in studies of species distributions: setting the scene”, Ecological Modelling **157**, 2, 89 – 100 (2002).

Guisan, A., R. Tingley, J. B. Baumgartner, I. Naujokaitis-Lewis, P. R. Sutcliffe, A. I. T. Tulloch, T. J. Regan, L. Brotons, E. McDonald-Madden, C. Mantyka-Pringle, T. G. Martin, J. R. Rhodes, R. Maggini, S. A. Setterfield, J. Elith, M. W. Schwartz, B. A. Wintle, O. Broennimann, M. Austin, S. Ferrier, M. R. Kearney, H. P. Possingham and Y. M. Buckley, “Predicting species distributions for conservation decisions”, Ecology Letters **16**, 12, 1424–1435 (2013).

Hackbusch, W., *Hierarchical Matrices: Algorithms and Analysis* (2015).

Haining, R. P., R. Kerry and M. A. Oliver, “Geography, spatial data analysis, and geostatistics: An overview”, Geographical Analysis **42**, 731 (2010).

Harter, S., H. Moe, J. Reed and A. David, “Fish assemblages associated with red grouper pits at pulley ridge, a mesophotic reef in the gulf of mexico”, Fishery Bulletin **115**, 419–432 (2017).

Hastie, T. and R. Tibshirani, “Generalized additive models”, Statistical Science **Vol. 1**, 297–318 (1986).

Hastie, T. and R. Tibshirani, *Generalized Additive Models* (Chapman and Hall, New York, 1990).

Kelly, S., “Basic introduction to pygame”, (2016).

Lembke, C., S. Grasty, A. Silverman, H. A. Broadbent, S. E. Butcher and S. Murawski, “The camera-based assessment survey system (c-bass): A towed camera platform for reef fish abundance surveys and benthic habitat characterization in the gulf of mexico”, Continental Shelf Research **151**, 62–71 (2017).

Li, J., A. D. Heap, A. Potter and J. J. Daniell, “Application of machine learning methods to spatial interpolation of environmental variables”, Environmental Modelling & Software **26**, 12, 1647 – 1659 (2011).

Lin, Y.-P., W.-C. Lin, Y.-C. Wang, W.-Y. Lien, T. Huang, C.-C. Hsu, D. S. Schmeller and N. D. Crossman, “Systematically designating conservation areas for protecting habitat quality and multiple ecosystem services”, Environmental Modelling & Software **90**, 126 – 146 (2017).

Lindgren, F., H. Rue and J. Lindstrm, “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 (2011).

Mateo-Sánchez, M. C., A. Gastón, C. Ciudad, J. I. García-Viñas, J. Cuevas, C. López-Leiva, A. Fernández-Landa, N. Algeet-Abarquero, M. Marchamalo, M. Fortin and S. Saura, “Seasonal and temporal changes in species use of the landscape: how do they impact the inferences from multi-scale habitat modeling?”, Landscape Ecology **31**, 1261–1276 (2015).

McDonald, A., J. S. Parslow and A. J. Davidson, “Interpretation of a modified linear model of catch-per-unit-effort data in a spatially-dynamic fishery”, Environmental Modelling & Software **16**, 2, 167 – 181, environmental Modelling and Socioeconomics (2001).

Nychka, D., R. Furrer, J. Paige and S. Sain, “fields: Tools for spatial data”, R package version 10.3 (2017).

Pfeffermann, D., “New important developments in small area estimation”, Statistical Science **28**, 1, 4068 (2013).

Prosser, D., C. Ding, R. Erwin, T. Mundkur, J. Sullivan and E. C. Ellis, “Species distribution modeling in regions of high need and limited data: waterfowl of china”, Avian Research **9**, 1–14 (2018).

Ramchoun, H., M. J. Idrissi, Y. Ghanou and M. Ettaouil, “Multilayer perceptron: Architecture optimization and training”, Int. J. Interact. Multim. Artif. Intell. **4**, 26–30 (2016).

Robertson, G. P., “Geostatistics in ecology: Interpolating with known variance”, Ecology **68**, 3, 744–748 (1987).

Ros-Pena, L., T. Kneib, C. Cadarso-Surez, N. Klein and M. Marey-Prez, “Studying the occurrence and burnt area of wildfires using zero-one-inflated structured additive beta regression”, Environmental Modelling & Software **110**, 107 – 118, special Issue on Environmental Data Science and Decision Support: Applications in Climate Change and the Ecological Footprint (2018).

Sainte-Marie, B. and B. Hargrave, “Estimation of scavenger abundance and distance of attraction to bait”, Marine Biology **94**, 431–443 (1987).

Saul, S. and S. Purkis, “Semi-automated object-based classification of coral reef habitat using discrete choice models”, Remote Sensing **7**, 12, 15894–15916 (2015).

Saul, S., J. Walter, D. Die, D. Naar and B. Donahue, “Modeling the spatial distribution of commercially important reef fishes on the west florida shelf”, Fisheries Research **143**, 12 – 20 (2013).

Schoener, T., “A brief history of optimal foraging ecology”, (1987).

Simon Wood, “mgcv: Mixed gam computation vehicle with automatic smoothness estimation”, R package version 1.8-31 (2019).

Somerton, D. and C. T. Glendhill, “Report of the national marine fisheries service workshop on underwater video analysis, august 4-6, 2004”, (2005).

Staff, S. S. and T. Loecke, “Rapid carbon assessment: Methodology, sampling and summary”, (2016).

Stein, M. L., *Statistical Interpolation of Spatial Data: Some Theory for Kriging* (Springer, New York, 1999).

Stohlgren, T. J., P. Ma, S. Kumar, M. Rocca, J. T. Morisette, C. S. Jarnevich and N. Benson, “Ensemble habitat mapping of invasive plant species”, Risk Analysis **30**, 2, 224–235 (2010).

Stoner, A. W., “Effects of environmental variables on fish feeding ecology: implications for the performance of baited fishing gear and stock assessment”, Journal of Fish Biology **65**, 6, 1445–1471 (2004).

Stratford, D. S., C. A. Pollino and A. E. Brown, “Modelling population responses to flow: The development of a generic fish population model”, Environmental Modelling & Software **79**, 96 – 119 (2016).

Streich, M. K., M. J. Ajemian, J. J. Wetz and G. W. Stunz, “A comparison of fish community structure at mesophotic artificial reefs and natural banks in the western gulf of mexico”, Marine and Coastal Fisheries **9**, 1, 170–189 (2017).

Vecchia, A. V., "Estimation and model identification for continuous spatial processes", Journal of the Royal Statistical Society. Series B (Methodological) **50**, 2, 297–312 (1988).

Wenger, S. J. and J. D. Olden, "Assessing transferability of ecological models: an underappreciated aspect of statistical validation", Methods in Ecology and Evolution **3**, 2, 260–267 (2012).

Williamson, D. J., G. L. Burn, S. Simoncelli, J. Griffi, R. Peters, D. M. Davis and D. M. Owen, "Machine learning for cluster analysis of localization microscopy data", Nat Commun **11(1)**, 1493 (2020).

Ye, L., L. Gao, R. Marcos-Martinez, D. Mallants and B. A. Bryan, "Projecting australia's forest cover dynamics and exploring influential factors using deep learning", Environmental Modelling & Software **119**, 407 – 417 (2019).

Yuan, M. and Y. Lin, "Model selection and estimation in regression with grouped variables", Journal of the Royal Statistical Society, Series B **68(1)**, 4967 (2006).

APPENDIX A

BART FOR CORRELATED DATA

A.1 Marginal Likelihood and Posterior Distribution

A.1.1 Marginal Likelihood

The marginal likelihood $p(R|D)$ can be derived as follows.
First, by (4.12), we know

$$p(R|D,\mu) = (2\pi)^{-\frac{n}{2}}|\Sigma|^{-\frac{1}{2}}exp\{-\frac{1}{2}(R-D\mu)^T\Sigma^{-1}(R-D\mu)\} \tag{A.1}$$

If given $\pi(\mu) \sim N(\bar{\mu}, Q^{-1})$, where

$$\pi(\mu) = (2\pi)^{-\frac{b}{2}}|Q|^{\frac{1}{2}}exp\{-\frac{1}{2}(\mu-\bar{\mu})^TQ(\mu-\bar{\mu})\} \tag{A.2}$$

The marginal distrionbution of $p(R|D)$ can be calculated by integrated out μ.

$$p(R|D) = \int p(R|D,\mu)\pi(\mu)d\mu$$

Let's check the product of likelihood and prior.

$$\begin{aligned} p(R|D,\mu)\pi(\mu) &= (2\pi)^{-\frac{n}{2}}|\Sigma|^{-\frac{1}{2}}exp\{-\frac{1}{2}(R-D\mu)^T\Sigma^{-1}(R-D\mu)\}* \\ &\quad (2\pi)^{-\frac{b}{2}}|Q|^{\frac{1}{2}}exp\{-\frac{1}{2}(\mu-\bar{\mu})^TQ(\mu-\bar{\mu})\} \\ &= (2\pi)^{-\frac{n+b}{2}}|\Sigma|^{-\frac{1}{2}}|Q|^{\frac{1}{2}}* \\ &\quad exp\{-\frac{1}{2}\underbrace{[(R-D\mu)^T\Sigma^{-1}(R-D\mu)+(\mu-\bar{\mu})^TQ(\mu-\bar{\mu})]}_{(*)}\} \end{aligned} \tag{A.3}$$

Since

$$\begin{aligned} (*) &= R^T\Sigma^{-1}R - 2R^T\Sigma^{-1}D\mu + \mu^TD^T\Sigma^{-1}D\mu + \mu^TQ\mu - 2\bar{\mu}^TQ\mu + \bar{\mu}^TQ\bar{\mu} \\ &= \mu^T(D^T\Sigma^{-1}D+Q)\mu - 2\underline{(R^T\Sigma^{-1}D+\bar{\mu}^TQ)}\mu + R^T\Sigma^{-1}R + \bar{\mu}^TQ\bar{\mu} \end{aligned}$$

Then, we can introduce a variable v and think about the following term.

$$\begin{aligned} &(\mu-v)^T(D^T\Sigma^{-1}D+Q)(\mu-v) \\ &= \mu^T(D^T\Sigma^{-1}D+Q)\mu - 2\underline{v^T(D^T\Sigma^{-1}D+Q)}\mu + v^T(D^T\Sigma^{-1}D+Q)v \end{aligned}$$

To make the underline coefficient equal, we can let

$$v^T(D^T\Sigma^{-1}D+Q) = R^T\Sigma^{-1}D+\bar{\mu}^TQ$$

$$v^T = (R^T\Sigma^{-1}D+\bar{\mu}^TQ)(D^T\Sigma^{-1}D+Q)^{-1}$$

and

$$v = (Q+D^T\Sigma^{-1}D)^{-1}(Q\bar{\mu}+D^T\Sigma^{-1}R) \tag{A.4}$$

Finally

$$(*) = (\mu - v)^T(Q + D^T\Sigma^{-1}D)(\mu - v) + C$$

where

$$C = -v^T(Q + D^T\Sigma^{-1}D)v + R^T\Sigma^{-1}R + \bar{\mu}^T Q\bar{\mu}$$

Plug (A.1) and (A.2) into the integral term.

$$\begin{aligned}
&\int p(R|D,\mu)p(\mu)d\mu \\
&= (2\pi)^{-\frac{n+b}{2}}|\Sigma|^{-\frac{1}{2}}|Q|^{\frac{1}{2}}exp\{-\frac{1}{2}C\}\int exp\{-\frac{1}{2}(\mu - v)^T(Q + D^T\Sigma^{-1}D)(\mu - v)\}d\mu \\
&= (2\pi)^{-\frac{n+b}{2}}|\Sigma|^{-\frac{1}{2}}|Q|^{\frac{1}{2}}exp\{-\frac{1}{2}C\}(2\pi)^{\frac{b}{2}}|Q + D^T\Sigma^{-1}D|^{-\frac{1}{2}} \\
&\qquad \cdot \int (2\pi)^{-\frac{b}{2}}|Q + D^T\Sigma^{-1}D|^{\frac{1}{2}}exp\{-\frac{1}{2}(\mu - v)^T(Q + D^T\Sigma^{-1}D)(\mu - v)\}d\mu \\
&= \frac{(2\pi)^{-\frac{n}{2}}|\Sigma|^{-\frac{1}{2}}|Q|^{\frac{1}{2}}}{|Q + D^T\Sigma^{-1}D|^{\frac{1}{2}}}exp\{-\frac{1}{2}C\}
\end{aligned}$$

After simplifying we can get (A.5) which is same to (4.14).

$$p(R|D) = \frac{(2\pi)^{-\frac{n}{2}}|\Sigma|^{-\frac{1}{2}}|Q|^{\frac{1}{2}}}{|Q + D^T\Sigma^{-1}D|^{\frac{1}{2}}}exp\{-\frac{1}{2}(-v^T(Q + D^T\Sigma^{-1}D)v + \bar{\mu}^T Q\bar{\mu} + R^T\Sigma^{-1}R)\} \tag{A.5}$$

where, $v = (Q + D^T\Sigma^{-1}D)^{-1}(Q\bar{\mu} + D^T\Sigma^{-1}R)$.

A.1.2 Posterior Distribution

Based on the proof of marginal likelihood above, it's easy to get the posterior distribution $p(\mu|R)$. By (4.18), we have

$$p(\mu|R) \propto p(R|\mu)\pi(\mu) = p(R|D,\mu)\pi(\mu)$$

If given, $p(R|D,\mu)$ in (A.1) and $\pi(\mu)$ in (A.2) . Then, by (A.3) and (A.4), we can directly prove that

$$p(\mu|R) \sim N(\ (Q + D^T\Sigma^{-1}D)^{-1}(Q\bar{\mu} + D^T\Sigma^{-1}R)\ ,\ (Q + D^T\Sigma^{-1}D)^{-1}\ ) \tag{A.6}$$

A.2 Invariant under Reordering

If P is a permutation matrix, it has the property that $P^{-1} = P^T$. Then, according (4.21) and (4.22) we can prove (A.7).

$$\begin{aligned} D^T\Sigma^{-1}R &= (PD_P)^T(P\Sigma_P P^T)^{-1}(PR_P) \\ &= D_P^T P^T P \Sigma_P^{-1} P^T P R_P \\ &= D_P^T \Sigma_P^{-1} R_P \end{aligned} \tag{A.7}$$

If given $Q = \tau^{-2}I$, similar to (A.7) we can prove that

$$Q + D^T\Sigma^{-1}D = Q + (PD_P)^T P\Sigma_P^{-1}P^T PD_P = Q + D_P^T\Sigma_P^{-1}D_P \tag{A.8}$$

Let's recall (4.16) and (4.20),

$$\frac{p(R|D^{i+1})}{p(R|D^i)} = \frac{|Q^{i+1}|^{\frac{1}{2}}}{|Q^i|^{\frac{1}{2}}} \frac{|Q^i + (D^i)^T\Sigma^{-1}D^i|^{\frac{1}{2}}}{|Q^{i+1} + (D^{i+1})^T\Sigma^{-1}D^{i+1}|^{\frac{1}{2}}} \cdot exp\{\frac{1}{2}R^T\Sigma^{-1}$$
$$[D^{i+1}(Q^{i+1} + (D^{i+1})^T\Sigma^{-1}D^{i+1})^{-1}(D^{i+1})^T - D^i(Q^i + (D^i)^T\Sigma^{-1}D^i)^{-1}(D^i)^T]\Sigma^{-1}R\}$$

and,

$$p(\mu|R) \sim N((Q + D^T\Sigma^{-1}D)^{-1}D^T\Sigma^{-1}R, (Q + D^T\Sigma^{-1}D)^{-1})$$

By applying (A.7) and (A.8), obviously, (4.16) and (4.20) are invariant under reordering.

A.3 On the Calculation of Marginal Likelihood Ratio

A.3.1 Calculate Matrix A

We use matrix A to denote (4.25)

$$A = Q + D_P^T \Sigma_P^{-1} D_P, \qquad Q = \tau^{-2} I$$

According the discussion of reordering in section 4.2.4, it's easy to know A is a symmetric matrix. We can denote it as follows.

$$A = \begin{bmatrix} a_{11}+\tau^{-2} & a_{12} & \dots & a_{1b} \\ a_{21} & a_{22}+\tau^{-2} & \dots & a_{2b} \\ \vdots & \vdots & \ddots & \\ a_{b1} & a_{b2} & \dots & a_{bb}+\tau^{-2} \end{bmatrix} \tag{A.9}$$

where

$$a_{ji} = a_{ij} = \sum_{h \in n_i} \sum_{l \in n_j} q_{hl}, \quad i \le j, \quad i,j \in \{1,...,b\}$$

$n_k, k \in \{1,...,b\}$ is the index set of observations that associated with bottom node k and q_{hl} is the h^{th} row and l^{th} column entry in Σ_P^{-1}.

$$\Sigma_P^{-1} = \begin{bmatrix} q_{11} & q_{12} & \dots & q_{1n} \\ q_{21} & q_{22} & \dots & q_{2n} \\ \vdots & \vdots & \ddots & \\ q_{n1} & q_{n2} & \dots & q_{nn} \end{bmatrix}$$

So, the operations to calculate A is the summation of non-zero entries in Σ^{-1}.

A.3.2 The Block Form of Matrix E

Plug A into (4.23), we can get

$$\begin{aligned} \frac{p(R|D^{i+1})}{p(R|D^{i})} =& \frac{|Q^{i+1}|^{1/2}}{|Q^{i}|^{1/2}} \frac{|A^{i}|^{1/2}}{|A^{i+1}|^{1/2}} \\ & \cdot exp\{\frac{1}{2} R_P^T \Sigma_P^{-1} \underbrace{[D_P^{i+1}(A^{i+1})^{-1}(D_P^{i+1})^T - D_P^{i}(A^{i})^{-1}(D_P^{i})^T]}_{E} \Sigma_P^{-1} R_P\} \end{aligned}$$

To understand the form of E, we have to consider the birth and death operations respectively. Without losing generality, we can assume that birth or death operation occurs in $(i+1)^{th}$ MCMC iteration. Since the dummy matrix D has very special form (see section 4.2.1), we developed an algorithm as following to achieve computational efficiency.

(1) Birth

In this scenario, the tree has b bottom nodes at i step and $b+1$ nodes at $i+1$ step. So, $(A^i)^{-1}$ and $(A^{i+1})^{-1}$ are $b \times b$ and $(b+1) \times (b+1)$ matrices. We can denote them by block matrices as follows.

$$(A^{i+1})^{-1} = \begin{bmatrix} V_{11}^{i+1} & V_{12}^{i+1} \\ V_{21}^{i+1} & V_{22}^{i+1} \end{bmatrix}, \quad (A^i)^{-1} = \begin{bmatrix} V_{11}^i & v_{12}^i \\ v_{21}^i & v_{22}^i \end{bmatrix}$$

where, V_{11}^{i+1} and V_{11}^i are $(b-1) \times (b-1)$ matrices; $V_{12}^{i+1} = (V_{21}^{i+1})^T$ is $(b-1) \times 2$ matrix; $v_{12}^i = v_{21}^i$ is a $b-1$ column vector; v_{22}^i is a scalar.

We create a matrix

$$(A^i)_{ex}^{-1} = \begin{bmatrix} V_{11}^i & v_{12}^i & v_{12}^i \\ v_{21}^i & v_{22}^i & v_{22}^i \\ v_{21}^i & v_{22}^i & v_{22}^i \end{bmatrix}$$

Let $B = (A^{i+1})^{-1} - (A^i)_{ex}^{-1}$, then, we can get

$$B = \begin{bmatrix} V_{11}^{i+1} - V_{11}^i & V_{12}^{i+1} - \begin{bmatrix} v_{12}^i & v_{12}^i \end{bmatrix} \\ V_{21}^{i+1} - \begin{bmatrix} v_{21}^i \\ v_{21}^i \end{bmatrix} & V_{22}^{i+1} - \begin{bmatrix} v_{22}^i & v_{22}^i \\ v_{22}^i & v_{22}^i \end{bmatrix} \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1(b+1)} \\ b_{21} & b_{22} & \dots & b_{2(b+1)} \\ \vdots & \vdots & \ddots & \\ b_{(b+1)1} & b_{(b+1)2} & \dots & b_{(b+1)(b+1)} \end{bmatrix}$$

(2) Death

Similar to birth scenario, we denote the matrices $(A^i)^{-1}$ and $(A^{i+1})^{-1}$ as following.

$$(A^i)^{-1} = \begin{bmatrix} V_{11}^i & V_{12}^i \\ V_{21}^i & V_{22}^i \end{bmatrix}, \quad (A^{i+1})^{-1} = \begin{bmatrix} V_{11}^{i+1} & v_{12}^{i+1} \\ v_{21}^{i+1} & v_{22}^{i+1} \end{bmatrix}$$

where, V_{11}^{i+1} and V_{11}^i are $(b-2) \times (b-2)$ matrices; $V_{12}^i = (V_{21}^i)^T$ is $(b-2) \times 2$ matrix; $v_{12}^{i+1} = v_{21}^{i+1}$ is a $b-2$ column vector; v_{22}^{i+1} is a scalar.

Create a matrix

$$(A^{i+1})_{ex}^{-1} = \begin{bmatrix} V_{11}^{i+1} & v_{12}^{i+1} & v_{12}^{i+1} \\ v_{21}^{i+1} & v_{22}^{i+1} & v_{22}^{i+1} \\ v_{21}^{i+1} & v_{22}^{i+1} & v_{22}^{i+1} \end{bmatrix}$$

Different from the birth scenario, $B = (A^{i+1})_{ex}^{-1} - (A^i)^{-1}$

$$B = \begin{bmatrix} V_{11}^{i+1} - V_{11}^i & \begin{bmatrix} v_{12}^{i+1} & v_{12}^{i+1} \end{bmatrix} - V_{12}^i \\ \begin{bmatrix} v_{21}^{i+1} \\ v_{21}^{i+1} \end{bmatrix} - V_{21}^i & \begin{bmatrix} v_{22}^{i+1} & v_{22}^{i+1} \\ v_{22}^{i+1} & v_{22}^{i+1} \end{bmatrix} - V_{22}^i \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1b} \\ b_{21} & b_{22} & \dots & b_{2b} \\ \vdots & \vdots & \ddots & \\ b_{b1} & b_{b2} & \dots & b_{bb} \end{bmatrix}$$

Block form of matrix E

We can denote E as a block matrix

$$E = D_P^{i+1}(A^{i+1})^{-1}(D_P^{i+1})^T - D_P^i(A^i)^{-1}(D_P^i)^T = \begin{bmatrix} E_{11} & E_{12} & ... & E_{1b'} \\ E_{21} & E_{22} & ... & E_{2b'} \\ \vdots & \vdots & \ddots & \\ E_{b'1} & E_{b'2} & ... & E_{b'b'} \end{bmatrix}$$

where

$$b' = \begin{cases} b+1, & \text{Birth,} \\ b, & \text{Death.} \end{cases}$$

Each block matrix has a special form

$$E_{ij} = E_{ji}^T = b_{ij}\mathbb{J}_{ij}, \quad i \le j, \quad i,j \in \{1,...,b'\}$$

where b_{ij} is the (i,j) element of matrix B calculated in birth or death step; $\mathbb{J}_{ij}$ is a card(n_i) × card(n_j) matrix whose entries are 1s. (card(n_k) is the cardinality of set n_k).

A.3.3 Calculate Marginal Likelihood Ratio

Let's set

$$R_P^T\Sigma_P^{-1} = [\omega_1 \quad \omega_2 \quad \dots \quad \omega_{b'}], \quad \omega_i = [\omega_{ij}], \quad j \in n_i$$

and

$$u = R_P^T\Sigma_P^{-1}E\Sigma_P^{-1}R_P$$

Then, u can be calculated

$$\begin{aligned} u &= [\omega_1 \quad \omega_2 \quad \dots \quad \omega_{b'}]\, E \begin{bmatrix} \omega_1^T \\ \omega_2^T \\ \vdots \\ \omega_{b'}^T \end{bmatrix} \\ &= \sum_{i=1}^{b'}\sum_{j=1}^{b'} \omega_i E_{ij} \omega_j^T \\ &= \sum_{i=1}^{b'}\sum_{j=1}^{b'} (\omega_i \mathbb{J}_{ij} \omega_j^T) b_{ij} \\ &= \sum_{i=1}^{b'}\sum_{j=1}^{b'} [(\sum_{h\in n_i} \omega_{ih})(\sum_{l\in n_j} \omega_{jl}) b_{ij}] \end{aligned}$$

Finally, we can get the marginal likelihood ratio as follows.

$$\frac{p(R|D^{i+1})}{p(R|D^i)} = \begin{cases} \tau^{-1}\frac{|A^i|}{|A^{i+1}|}exp\{\frac{1}{2}u\} & Birth \\ \tau\frac{|A^i|}{|A^{i+1}|}exp\{\frac{1}{2}u\} & Death \end{cases}$$

APPENDIX B

NEAREST NEIGHBOR GAUSSIAN PROCESS

The easy way to understand nearest neighbor Gaussian process is from the linear form of Gaussian process (B.1).

$$\boldsymbol{w} = \boldsymbol{H}\boldsymbol{w} + \boldsymbol{\eta} \tag{B.1}$$

where $\boldsymbol{w}$ is an instance of Gaussian process $W \sim GP(0, K(\cdot, \cdot|\boldsymbol{\theta}))$ and $\boldsymbol{w} \sim \boldsymbol{N(0, C)}$, C is the covariance matrix calculated by $K(\cdot, \cdot|\boldsymbol{\theta})$. The structure of $\boldsymbol{H}$ is as follows.

$$\boldsymbol{H} = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 \\ h_{21} & 0 & 0 & \dots & 0 \\ h_{31} & h_{32} & 0 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ h_{n1} & h_{n2} & \dots & h_{n(n-1)} & 0 \end{bmatrix}$$

$$\begin{gathered} w_1 = 0 + \eta_1 \\ w_2 = h_{21}w_1 + \eta_2 \\ w_3 = h_{31}w_1 + h_{32}w_2 + \eta_3 \\ \vdots \\ w_n = h_{n1}w_1 + h_{n2}w_2 + \dots + h_{n(n-1)}w_{(n-1)} + \eta_n \end{gathered}$$

and

$$\boldsymbol{\eta} \sim \boldsymbol{N(0, \Lambda)}$$

where $\boldsymbol{\Lambda}$ is diagonal with entries $\Lambda_{11} = var(w_1)$ and $\Lambda_{ii} = var(w_i|\{w_j : j < i\})$ for $i = 2, \dots, n$.

Since $\boldsymbol{I} - \boldsymbol{H}$ is nonsingular

$$\boldsymbol{I} - \boldsymbol{H} = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \\ -h_{21} & 1 & 0 & \dots & 0 \\ -h_{31} & -h_{32} & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ -h_{n1} & -h_{n2} & \dots & -h_{n(n-1)} & 1 \end{bmatrix}$$

Then, (B.1) can be transformd to $\boldsymbol{w} = \boldsymbol{(I - H)^{-1}\eta}$. So,

$$\boldsymbol{C} = \boldsymbol{(I - H)^{-1}\Lambda(I - H)^{-T}} \tag{B.2}$$

Recall,

$$w_{i+1} = h_{(i+1)1}w_1 + h_{(i+1)2}w_2 + \dots + h_{(i+1)i}w_i + \eta_{i+1} \tag{B.3}$$

Let, $\boldsymbol{h_{i+1}} = (h_{(i+1)1}, ..., h_{(i+1)i})$ and $\boldsymbol{w_{i+1}} = (w_1, ..., w_i, w_{i+1}) = (\boldsymbol{w_i}, w_{i+1})$, where $\boldsymbol{w_i} = (w_1, ..., w_i)$.

Note: For any matrix M and set of indices $I_1, I_2 \in \{1, 2, ..., n\}$, let $M[I_1, I_2]$ denote the submatrix of M formed by the rows indexed by I_1 and columns indexed by I_2.

Let

$$var(w_1, ..., w_{i+1}) = \boldsymbol{C}$$

Then,

$$var(w_1, ..., w_i) = \boldsymbol{C[1:i, 1:i]}$$

and

$$\boldsymbol{C} = \begin{bmatrix} \boldsymbol{C[1:i, 1:i]} & \boldsymbol{C[1:i, i+1]} \\ \boldsymbol{C[i+1, 1:i]} & \boldsymbol{C[i+1, i+1]} \end{bmatrix}$$

By equation (B.3), we can get

$$\boldsymbol{C[i+1, 1:i]} = \boldsymbol{h_{i+1}} \cdot \boldsymbol{C[1:i, 1:i]}$$

$$\boldsymbol{C[i+1, i+1]} = \Lambda_{i+1,i+1} + \boldsymbol{h_{i+1}} \cdot \boldsymbol{C[1:i, i+1]}$$

Then, $\boldsymbol{h_{i+1}}$ and $\Lambda_{i+1,i+1}$ can be calculated as follows.

$$\boldsymbol{h_{i+1}} = \boldsymbol{C[i+1, 1:i]C[1:i, 1:i]^{-1}} \tag{B.4}$$

$$\Lambda_{i+1,i+1} = \boldsymbol{C[i+1, i+1]} - \boldsymbol{h_{i+1}} \cdot \boldsymbol{C[1:i, i+1]} \tag{B.5}$$

Using (B.4) and (B.5), the covariance matrix $\boldsymbol{C}$ can be decomposited by (B.2). However, the computational complexity of (B.4) still increases as the dimension of $\boldsymbol{C[1:i, 1:i]}$ increasing ($O(n^3)$). In order to achieve the sparsity, we permit no more than m elements in $\boldsymbol{h_i}$ (the i-th row of matrix $\boldsymbol{H}$) to be nonzero.

Let $ne(i)$ to represent the number of nearest neighbors of point $i = 1, ..., n$ and $ne(i) \leq m$. Then equation (B.4) and (B.5) become

$$\boldsymbol{h_{ne[i+1]}} = \boldsymbol{C[i+1, ne(i+1)]C[ne(i+1), ne(i+1)]^{-1}} \tag{B.6}$$

$$\Lambda_{i+1,i+1} = \boldsymbol{C[i+1, i+1]} - \boldsymbol{h_{ne[i+1]}} \cdot \boldsymbol{C[ne(i+1), i+1]} \tag{B.7}$$

The size of linear system { (B.6), (B.7) } is at most $m \times m$. So, the computational complexity decreases from $O(n^3)$ to $O(nm^3)$.

From (B.2), (B.6) and (B.7), we can get that

$$\boldsymbol{\tilde{C} = (I - H)^{-1}\Lambda(I - H)^{-T}} \tag{B.8}$$

$$\boldsymbol{\tilde{C}^{-1} = (I - H)^{T}\Lambda^{-1}(I - H)} \tag{B.9}$$

where $\boldsymbol{H}$ and $\boldsymbol{\Lambda}$ are computed from (B.6) and (B.7) respectively.

Since $\Sigma = \boldsymbol{C} + \tau^2\boldsymbol{I}$, by Sherman Woodbury Morrison (SWM) identity, we can get:

$$\Sigma^{-1} = (\boldsymbol{C} + \tau^2\boldsymbol{I})^{-1} = \tau^{-2}\boldsymbol{I} - \tau^{-4}(\boldsymbol{C}^{-1} + \tau^{-2}\boldsymbol{I})^{-1}$$

Then, the approximation of Σ^{-1} is as follows:

$$\tilde{\Sigma}^{-1} = \tau^{-2}\boldsymbol{I} - \tau^{-4}(\boldsymbol{\tilde{C}}^{-1} + \tau^{-2}\boldsymbol{I})^{-1} \tag{B.10}$$

The calculation of $(\boldsymbol{\tilde{C}}^{-1} + \tau^{-2}\boldsymbol{I})^{-1}$ can enjoy the sparsity of $\boldsymbol{\tilde{C}}^{-1}$.