이학박사 학위논문

Bayesian Nonparametric Regression via Overcomplete Systems with B-spline Bases

B-스플라인 과완비 체계를 이용한 비모수 베이즈 회귀 모형 연구

2021년 8월

서울대학교 대학원

통계학과

박세원

Bayesian Nonparametric Regression via Overcomplete Systems with B-spline Bases

지도교수 이재용

이 논문을 이학박사 학위논문으로 제출함

2021년 4월

서울대학교 대학원
통계학과
박세원

박세원의 이학박사 학위논문을 인준함

2021년 6월

위원장 오희석

부위원장 이재용

위원 장원철

위원 임채영

위원 조성일

Bayesian Nonparametric Regression via Overcomplete Systems with B-spline Bases

by

Sewon Park

A Thesis
submitted in fulfillment of the requirement
for the degree of
Doctor of Philosophy
in
Statistics

Department of Statistics
College of Natural Sciences
Seoul National University
August, 2021

Abstract

In this dissertation, we propose the Lévy Adaptive B-Spline regression (LABS) model, an extension of the LARK models, to estimate functions with varying degrees of smoothness. LABS model is a LARK with B-spline bases as generating kernels. By changing the degrees of the B-spline basis, LABS can systematically adapt the smoothness of functions, i.e., jump discontinuities, sharp peaks, etc. Results of simulation studies and real data examples support that this model catches not only smooth areas but also jumps and sharp peaks of functions. The LABS model has the best performance in almost all examples. We also provide theoretical results that the mean function for the LABS model belongs to the specific Besov spaces based on the degrees of the B-spline basis and that the prior of the model has the full support on the Besov spaces.

Furthermore, we develop a multivariate version of the LABS model by introducing tensor product of B-spline bases named Multivariate Lévy Adaptive B-Spline regression (MLABS). MLABS model has comparable performance on both regression and classification problems. Especially, empirical results demonstrate that MLABS has more stable and accurate predictive abilities than state-of-the-art nonparametric regression models in relatively low-dimensional data.

Keywords: Lévy Random Measure; Besov Space; Tensor Product B-spline Basis; Reversible Jump Markov Chain Monte Carlo.

Student Number: 2015-20297

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Nonparametric regression model

Suppose we have a random sample of size n, $\mathbf{x}_1, \ldots, \mathbf{x}_n$, $\mathbf{x}_i \in \mathcal{X}$ and response variables $\mathbf{Y} = (Y_1, \ldots, Y_n)^T \in \mathbb{R}^n$ satisfying the following relationship,

$$Y_i = f(\mathbf{x}_i) + \varepsilon_i, \quad \varepsilon_i \sim \mathcal{N}(0, \sigma^2), \quad i = 1, \ldots, n, \tag{1.1}$$

where f is an unknown nonparametric regression function which maps $\mathcal{X}$ to $\mathbb{R}$. Here, we consider $\mathcal{X} = \mathbb{R}^p$. The nonparametric regression function is determined by only data without taking a prespecified structure. The nonparametric regression aims to identify the relationships between the predictors and responses and then to make further predictions on a new data set $\mathbf{x}^\star$ based on relationships as mentioned above. If p is one, the purpose is to locally approximate a target function referred to as the nonparametric function estimation. Moreover, when the responses takes discrete values (e.g, $Y \in \{0, 1\}$), the function f is estimated using classification algorithms. A common way of estimating an unknown mean function f is to express it as a sum of the functions

$$f(\mathbf{x}) \approx \sum_{j \in J} \phi_j(\mathbf{x}),$$

where the functions ϕ_j is specified nonparametrically. The most widely used form of ϕ_j is $\phi_j(\mathbf{x}) := g(\mathbf{x}; \boldsymbol{\theta}_j) \cdot \beta_j$, where $\{\beta_j\}_{j \in J}$, $\beta_j \in \mathbb{R}$ denote unknown coefficients, $\{\phi\}_{j \in J}$ is a basis set on $\mathcal{X}$ whose parameters is $\{\boldsymbol{\theta}_j\}_{j \in J}$. For recovering a regression function f, it is crucial which a basis set $\{\phi\}_{j \in J}$ is selected and then how to estimate β_js. There exist other basis sets like decision trees, and splines. Some nonparametric regression estimators for either univariate or multivariate data will be reviewed in the next section.

1.2 Literature Review

1.2.1 Literature review of nonparametric function estimation

In nonparametric function estimation, we often face smooth curves except for a finite number of jump discontinuities and sharp peaks, common in many climate and economic datasets. Heavy rainfalls cause a sudden rise in the water level of a river. The COVID-19 epidemic brought about a sharp drop in unemployment rates. Policymakers' decisions can give rise to abrupt changes. For instance, the United States Congress passed the National Minimum Drinking Age Act in 1984, which has been debated over several decades in the United States, establishing 21 as the minimum legal alcohol purchase age. This act caused a sudden rise in mortality for young Americans around 21. The abrupt changes can provide us with meaningful information on these issues, and it is vital to grasp the changes. There has been much research into the estimation of the local smoothness of the functions. The first approach minimizes the penalized sum of squares based on a locally varying smoothing parameter or penalty function across the whole domain. Pintore et al. (2006), Liu and Guo (2010), and Wang et al. (2013) modeled the smoothing parameter of smoothing spline to vary over the domain. Ruppert and Carroll (2000), Crainiceanu et al. (2007), Krivobokova et al. (2008), and Yang and Hong (2017) suggested the penalized splines based on the local penalty that adapts to spatial heterogeneity in the regression function. The second approach is the adaptive free-knot splines that choose the number and location of the knots from the data. Friedman (1991) and Luo and Wahba (1997) determined a set of knots using stepwise forward/backward knot selection procedures. Zhou and Shen (2001) avoided the problems of stepwise schemes and proposed optimal knot selection schemes introducing the knot relocation step. Smith and Kohn (1996), Denison et al. (1998a), Denison et al. (1998b), and DiMatteo et al. (2001) studied Bayesian estimation of free knot splines using MCMC techniques. The third approach is to use wavelet shrinkage estimators, including VisuShrink based on the universal threshold (Donoho and Johnstone, 1994), SureShrink based on Stein's unbiased risk estimator (SURE) function (Donoho and Johnstone, 1995), Bayesian thresholding rules by utilizing a mixture prior (Abramovich et al., 1998), and empirical Bayes methods for level-dependent threshold selection (Johnstone and Silverman, 2005). The fourth approach is to detect jump discontinuities in the regression curve. Koo (1997), Lee (2002), and Yang and

Song (2014) dealt with the estimation of discontinuous function using B-spline basis functions. Qiu and Yandell (1998), Qiu (2003), Gijbels et al. (2007), and Xia and Qiu (2015) identified jumps based on local polynomial kernel estimation.

In this thesis, we consider a function estimation method using overcomplete systems. A subset of the vectors $\{\phi\}_{j\in J}$ of Banach space $\mathcal{F}$ is called a *complete system* if

$$\|\eta - \sum_{j\in J} \beta_j \phi_j\| < \epsilon, \quad \forall \eta \in \mathcal{F}, \forall \epsilon > 0,$$

where $\beta_j \in \mathbb{R}$ and $J \in \mathbb{N} \cup \infty$. Such a complete system is *overcomplete* if removal of a vector ϕ_j from the system does not alter the completeness. In other words, an overcomplete system is constructed by adding basis functions to a complete basis (Lewicki and Sejnowski, 2000). Coefficients β_j in the expansion of η with an overcomplete system are not unique owing to the redundancy intrinsic in the overcomplete system. The non-uniqueness property can provide more parsimonious representations than those with a complete system (Simoncelli et al., 1992).

The Lévy Adaptive Regression Kernels (LARK) model, first proposed by Tu (2006), is a Bayesian regression model utilizing overcomplete systems with Lévy process priors. Tu (2006) showed the LARK model had sparse representations for η from an overcomplete system and improvements in nonparametric function estimation. Pillai et al. (2007) found out the relationship between the LARK model and a reproducing kernel Hilbert space (RKHS), and Pillai (2008) proved the posterior consistency of the LARK model. Chu et al. (2009) used continuous wavelets as the elements of an overcomplete system. Wolpert et al. (2011) obtained sufficient conditions for LARK models to lie in the some

Besov space or Sobolev space. Lee et al. (2020) devised an extended LARK model with multiple kernels instead of only one type of kernel.

1.2.2 Literature review of multivariate nonparametric regression

There has been much research on constructing the functions ϕ_j or selecting both basis elements and estimation techniques for multivariate data. The first approach is kernel-based methods are connected to the reproducing kernel Hilbert space (RKHS). By the representer theorem (Kimeldorf and Wahba, 1971), a regression function f over the RKHS can be expressed as

$$\hat{f}(\mathbf{x}) = \sum_{i=1}^{n} k(\mathbf{x}_i, \mathbf{x})\beta_i,$$

where k is a positive-definite real-valued kernel on $\mathcal{X} \times \mathcal{X}$ (See Wahba (1990) for details). A solution to regularization problems in a reproducing kernel Hilbert space (RKHS) is the well-known Support Vector Machine (SVM) (Boser et al., 1992; Cortes and Vapnik, 1995) with the kernel trick, which leads to computationally efficiency. Tipping (2000) developed a probabilistic SVM by putting a Gaussian prior for β_j, which obtained a sparser representation than the SVM.

Moreover, another approach for kernel-based methods is to take advantage of overcomplete bases, as mentioned above. In the Bayesian framework, an example of methods using the overcomplete system is the Bayesian additive regression kernels (BARK), proposed by Ouyang (2008). He proposed sparse additive models using a multivariate Gaussian kernel with the diagonal covariance function as an extension of the LARK method for multi-dimensional cases.

The second approach is to use the spline functions. The most representative spline-based model is the multivariate adaptive regression splines (MARS) introduced by Friedman (1991). The MARS has a form of a weighted sum of spline functions as

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{J} B_j(\mathbf{x}; \boldsymbol{\theta}_j)\beta_j,$$

where $\boldsymbol{\theta}_j$ is the parameter vector of the jth tensor product of univariate linear spline functions $B_j(\mathbf{x}; \boldsymbol{\theta})$. It has the advantages of capturing the nonlinear relationships and interactions between variables and simplifying high-dimensional problems into low-dimensional settings. Denison et al. (1998b) and Francom et al. (2018) proposed Bayesian approaches to the MARS and improved predictive performance compared to the original model. The neural network (NN) with two layers of hidden units can also be represented as a sum of spline functions as

$$\begin{aligned}
\mathbf{h}^{(1)} &= g^{(1)}\left((\boldsymbol{\beta}^{(1)})^T\mathbf{x} + \mathbf{a}^{(1)}\right), \\
\mathbf{h}^{(2)} &= g^{(2)}\left((\boldsymbol{\beta}^{(2)})^T\mathbf{h}^{(2)} + \mathbf{a}^{(2)}\right), \\
\hat{f}(\mathbf{x}) &= \sum_j h_j^{(2)}\beta_j^{(3)} + a^{(3)},
\end{aligned}$$

where $g^{(i)}$ is the ReLU (Rectified Linear Unit) activation function, $\boldsymbol{\beta}^{(i)}$ is the weights, and $\mathbf{a}^{(i)}$ is the bias for the ith hidden layer. The ReLU activation is $\max(x, 0)$ which equals the linear spline function in the tensor product bases of the MARS.

The third approach is ensemble methods, which combine several decision trees. That is, ϕ_j is a single tree model. These are divided into two main categories: bagging (Breiman, 1996) and boosting (Freund et al., 1999; Friedman, 2001). The bagging builds many trees based on different bootstrapped samples and averages the results of them. As an improved bagging model, random forest (Breiman, 2001) constructs many independent trees based on a random subset of the features and combines them. The boosted trees sequentially estimate regression trees and aggregate them to form a strong tree model. Chen and Guestrin (2016) developed the scalable and enhanced version of the gradient boosting algorithm named extreme gradient boosting. In the Bayesain framework, Chipman et al. (2010) proposed the Bayesian additive regression trees (BART) that constructs the function as

$$\hat{f}(\mathbf{x}) = \sum_{j=1}^{J} \mathcal{T}_j(\mathbf{x}; \mathcal{M}_j),$$

where $\mathcal{T}_j$ is a jth tree structure, and $\mathcal{M}_j$ is a set of parameters at the jth terminal nodes (also called leaves). The BART has become quite popular owing to theoretical results and outstanding empirical performance. Linero (2018) and Linero and Yang (2018) enhanced the BART model placing a sparsity inducing Dirichlet prior in high-dimensional problems.

1.3 Outline

In Chapter 2, we propose the Bayesian nonparametric method for function estimation via a Lévy process prior, which remedies the disadvantage for the LARK using a variety of B-spline bases as elements of an overcomplete system.

We show that the mean functions of the proposed model lie almost surely in the specific Besov space, and the prior has sufficiently large support on the same Besov space.

In Chapter 3, for the multi-dimensional data, we develop the multivariate nonparametric regression based on the LABS's framework by bringing in tensor products of the MARS. Numerical results illustrate the prorperties of the proposed model. A summary and future work are given in Chapter 4. Proofs for the main theorems in Chapter 2 are provided in Appendix A.1.

Chapter 2

Bayesian nonparametric function estimation using overcomplete systems with B-spline bases

2.1 Introduction

In this chapter, we develop a fully Bayesian approach with B-spline basis functions as the elements of an overcomplete system and call it the Lévy Adaptive B-Spline regression (LABS). Our main contributions to this work can be summarized as follows. First, The LABS model can systematically represent the smoothness of functions varying locally by changing the degrees of the B-spline basis. The form of a B-spline basis depends on the locations of knots and can be symmetrical or asymmetrical. The varying degree of B-spline basis enables the LABS model to adapt to the smoothness of functions. Second, We investigate two theoretical properties of the LABS model. First, the mean function of the LABS model exists in specific Besov spaces based on the types of degrees of B-spline basis. Second, the prior of the LABS model has full support on some Besov spaces. Thus, the proposed LABS model extends the range of smoothness classes of the mean function. Third, e e provide empirical results demonstrating that our model performs well in the spatially inhomogeneous functions, such as the functions with both jump discontinuities, sharp peaks, and smooth parts. The LABS model achieved the best results in almost every experiment compared to the popular nonparametric function estimation methods. In particular, the LABS model showed remarkable performance in estimating functions with jump discontinuities and outperformed other competing models.

The rest of the chapter is organized as follows. In section 2.2, we introduce the Lévy Adaptive Regression Kernels and discuss its properties. In section 2.3, we propose the LABS model and present an equivalent model with latent variables that make the posterior computation tractable. We present three theorems that the function spaces for the proposed model depend upon the degree of B-spline basis and that the prior has large support in some function spaces. We describe the detailed algorithm of posterior sampling using reversible jump Markov chain Monte Carlo in section 2.4. In section 2.5, the LABS model is compared with other methods in two simulation studies, and in section 2.6, three real-world data sets are analyzed using the LABS model. In the last section, we discuss some improvements and possible extensions of the proposed model.

2.2 Lévy adaptive regression kernels

In this section, we give a brief introduction to the LARK model. Let Ω be a complete separable metric space, and ν be a positive measure on $\mathbb{R} \times \Omega$ with $\nu(\{0\}, \Omega) = 0$ satisfying L_1 integrability condition,

$$\int\int_{\mathbb{R}\times A} (1 \wedge |\beta|)\nu(d\beta, d\omega) < \infty, \tag{2.1}$$

for each compact set $A \subset \Omega$. The Lévy random measure L with Lévy measure ν is defined as

$$L(d\omega) = \int_{\mathbb{R}} \beta N(d\beta, d\omega),$$

where N is a Poisson random measure with intensity measure ν. We denote $L \sim$ Lévy(ν). For any $t \in \mathbb{R}$, the characteristic function of $L(A)$ is

$$\mathbb{E}\left[e^{itL(A)}\right] = \exp\left\{\int\int_{\mathbb{R}\times A} (e^{it\beta} - 1)\nu(d\beta, d\omega)\right\}, \quad \text{for all } A \subset \Omega. \tag{2.2}$$

Let $g(x, \omega)$ be a real-valued function defined on $\mathcal{X} \times \Omega$ where $\mathcal{X}$ is another set. By integrating g with respect to a Lévy random measure L, we define a real-valued function on $\mathcal{X}$:

$$\eta(x) \equiv L[g(x)] = \int_{\Omega} g(x, \omega)L(d\omega) = \int_{\Omega}\int_{\mathbb{R}} g(x, \omega)\beta N(d\beta, d\omega), \forall x \in \mathcal{X}. \tag{2.3}$$

We call g a *generating function* of η.

When $\nu(\mathbb{R} \times \Omega) = M$ is finite, a Lévy random measure can be represented as $L(d\omega) = \sum_{j\leq J} \beta_j \delta_{\omega_j}$, where J has a Poisson distribution with mean $M > 0$ and $\{(\beta_j, \omega_j)\} \overset{iid}{\sim} \pi(d\beta_j, d\omega_j) := \nu/M, j = 1, 2, \ldots, J$. In this case, equation

(2.3) can be expressed as

$$\eta(x) = \sum_{j=1}^{J} g(x, \omega_j)\beta_j, \tag{2.4}$$

where $\{(\beta_j, \omega_j)\}$ is the random set of finite support points of a Poisson random measure. If g is bounded, L_1 integrability condition (2.1) implies the existence of (2.3) for all x. See Lee et al. (2020).

If a Lévy measure satisfying (2.1) is infinite, the number of the support points of $N(\mathbb{R}, \Omega)$ is infinite almost surely. Tu (2006) proved that the truncated Lévy random field $L_\epsilon[g]$ converges in distribution to $L[g]$ as $\epsilon \to 0$, where

$$L_\epsilon[g] = \int\int_{[-\epsilon,\epsilon]^c \times \Omega} g(x,\omega)\beta N(d\beta, d\omega) = \int\int_{\mathbb{R}\times\Omega} g(x,\omega)\beta N_\epsilon(d\beta, d\omega),$$

and N_ϵ is a Poisson measure on $\mathbb{R}$ with mean measure

$$\nu_\epsilon(d\beta, d\omega) := \nu(d\beta, d\omega) I_{|\beta|>\epsilon}.$$

This truncation was often used as an approximation of the posterior. For posterior computation methods for the Poisson random measure without truncation, see Lee (2007) and Lee and Kim (2004).

Together with data generating mechanism (1.1), the LARK model is defined as follows:

$$\mathbb{E}[Y|L, \theta] = \eta(x) \equiv \int_\Omega g(x,\omega) L(d\omega)$$
$$L|\theta \sim \text{Lèvy}(\nu)$$
$$\theta \sim \pi_\theta(d\theta),$$

where Lèvy(ν) denotes the Lèvy process which has the characteristic function and ν is a Lèvy measure satisfying (2.1). Tu (2006) used gamma, symmetric gamma, and symmetric α-stable (SαS) $(0 < \alpha < 2)$ Lèvy random fields. The conditional distribution for Y has a hyperparameter θ and π_θ denotes the prior distribution of θ. The generating function $g(x, \omega)$ is used as elements of an overcomplete system. Tu (2006) and Lee et al. (2020) used the Gaussian kernel, the Laplace kernel, and Haar wavelet as generating functions:

- Haar kernel: $g(x, \omega) := I\left(|\frac{x-\chi}{\lambda}| \le 1\right)$
- Gaussian kernel: $g(x, \omega) = \exp\left\{-\frac{(x-\chi)^2}{2\lambda^2}\right\}$
- Laplacian Kernel: $g(x, \omega) = \exp\left\{-\frac{|x-\chi|}{\lambda}\right\}$

with $\omega := (\chi, \lambda) \in \mathbb{R} \times \mathbb{R}^+ := \Omega$. All of the above generating functions are bounded.

This LARK model can be represented in a hierarchical structure as follows:

$$\begin{aligned}
Y_i |\, \eta(\mathbf{x}_i) &\overset{ind}{\sim} \mathcal{N}(\eta(\mathbf{x}_i), \sigma^2) \\
\eta(\mathbf{x}_i) &= \sum_{j=1}^{J} g(\mathbf{x}_i, \boldsymbol{\omega}_j)\beta_j \\
J \,|\, \epsilon &\sim \text{Pois}(\nu_\epsilon(\mathbb{R}, \Omega)) \\
(\beta_j, \boldsymbol{\omega}_j)|J, \epsilon &\overset{i.i.d}{\sim} \pi(d\beta_j, d\boldsymbol{\omega}_j) := \frac{\nu_\epsilon(d\beta_j, d\boldsymbol{\omega}_j)}{\nu_\epsilon(\mathbb{R}, \Omega)}
\end{aligned}$$

for $j = 1, \ldots, J$. J is the random number that is stochastically determined by Lèvy random measure, $(\beta_1, \ldots, \beta_J)$ is the unknown coefficients of a mean function and $(\boldsymbol{\omega}_1, \ldots, \boldsymbol{\omega}_J)$ is the parameters of the generating functions. To obtain samples from the posterior distribution under the LARK model, the reversible jump Markov chain Monte Carlo (RJMCMC) proposed by Green (1995) is used because some parameters have varying dimensions.

The LARK model stochastically extracts features and finds a compact representation for $\eta(\cdot)$ based on an overcomplete system. That is, it enables functions to be represented by the small number of elements from an overcomplete system. However, both the LARK model and most methods for function estimation use only one type of kernel or basis and can find out the restricted smoothness of the target function. These models cannot afford to capture all parts of the function with various degrees of smoothness. For example, we consider a noisy modified Heavisine function sampled at $n = 512$ equally spaced points on $[0, 1]$ in Figure 2.1. The data contains both smooth and non-smooth regions such as peaks and jumps. As shown in panel (a) of Figure 2.1, it is difficult for the LARK model with a finite Lèvy measure using Gaussian kernels to estimate jump parts of the data. We, therefore, propose a new model which can adapt the smoothness of function systematically by using a variety of B-spline bases as the generating elements of an overcomplete system.

2.3 Lévy adaptive B-spline regression

We consider a general type of basis function as the generating elements of an overcomplete system instead of specific kernel functions such as Haar, Laplacian, and Gaussian. The LABS model uses B-spline basis functions, which can systematically express jumps, sharp peaks, and smooth parts of the function.

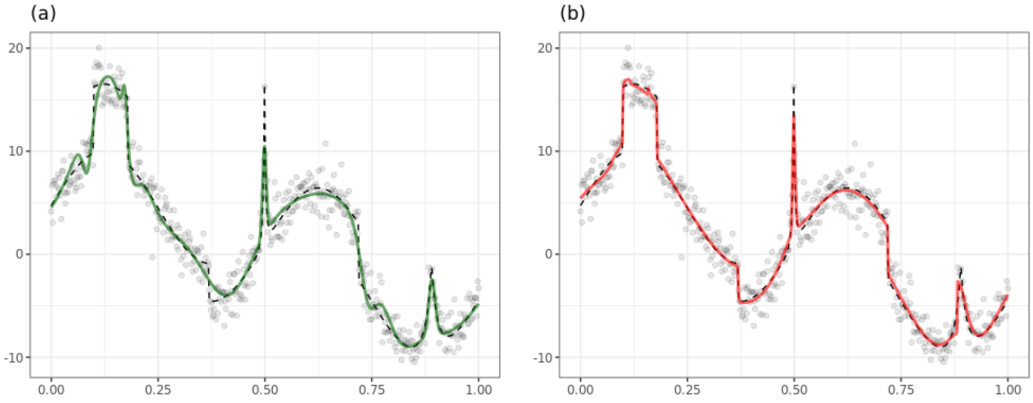


Figure 2.1: Comparison of curve fitting functions with (a) LARK and (b) LABS model for the modified Heavisine dataset. The solid lines are estimated functions and the dashed line represents the true function.

2.3.1 B-spline basis

The B-spline basis function consists of piecewise k degree polynomials with $k-1$ continuous derivatives. In general, the B-spline basis of degree k can be derived utilizing the Cox-de Boor recursion formula:

$$\begin{aligned} B^*_{0,i}(x) &:= \begin{cases} 1 & \text{if} \quad t_i \le x < t_{i+1} \\ 0 & \quad\text{otherwise} \end{cases} \\ B^*_{k,i}(x) &:= \frac{x - t_i}{t_{i+k} - t_i} B^*_{k-1,i}(x) + \frac{t_{i+k+1} - x}{t_{i+k+1} - t_{i+1}} B^*_{k-1,i+1}(x), \end{aligned} \tag{2.5}$$

where t_i are called knots which must be in non-descending order $t_i \le t_{i+1}$ (De Boor, 1972), (Cox, 1972). The B-spline basis of degree k, $B^*_{k,i}(x)$ then needs $(k+2)$ knots, $(t_i, \ldots, t_{i+k+1})$. For convenience of notation, we redefine the B-spline basis of degree k with a knot sequence $\boldsymbol{\xi}_k := (\xi_{k,1}, \ldots, \xi_{k,k+2})$ as follows.

$$
\begin{aligned}
B_0(x;\boldsymbol{\xi}_0) &:= \begin{cases} 1 \quad \text{if} \quad \xi_{0,1} \leq x < \xi_{0,2} \\ 0 \qquad\quad \text{otherwise} \end{cases} \\
B_k(x;\boldsymbol{\xi}_k) &:= \frac{x-\xi_{k,1}}{\xi_{k,(k+1)}-\xi_{k,1}} B_{k-1}(x;\boldsymbol{\xi}^{\star}_k) + \frac{\xi_{k,(k+2)}-x}{\xi_{k,(k+2)}-\xi_{k,2}} B_{k-1}(x;\boldsymbol{\xi}^{\star\star}),
\end{aligned}
\tag{2.6}
$$

where $\boldsymbol{\xi}^{\star}_k := (\xi_{k,1}, \xi_{k,2}, \dots, \xi_{k,(k+1)})$ and $\boldsymbol{\xi}^{\star\star}_k := (\xi_{k,2}, \xi_{k,3}, \dots, \xi_{k,(k+2)})$.

The B-spline basis functions can have a variety of shapes and smoothness determined by knot locations and its degree. For example, a B-spline basis function can be a piecewise constant (k = 0), linear (k = 1), quadratic (k = 2) , and cubic (k = 3) functions. Furthermore, the B-spline basis with equally spaced knots has a symmetric form. These bases are called Uniform B-splines. Examples of the B-spline basis functions of different degrees with equally spaced knots are shown in Figure 2.2.

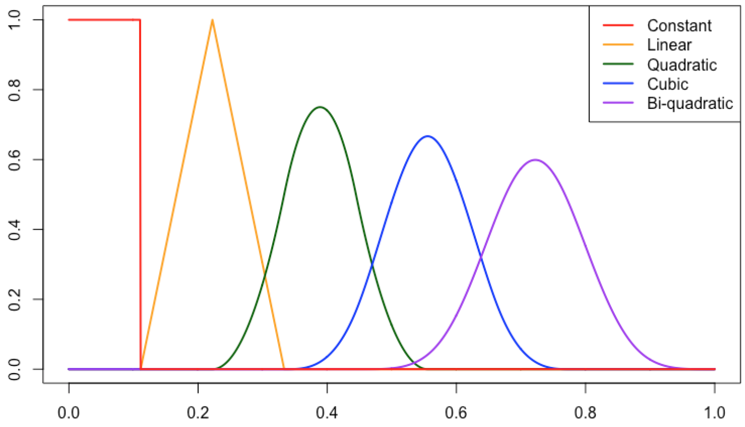


Figure 2.2: Different shapes of the B-spline basis function by increasing the degree k

2.3.2 Model specification

The LARK model with one kernel type can't estimate well functions with both continuous and discontinuous parts. To improve this, we consider various B-spline basis functions simultaneously for estimating all parts of the unknown function. The proposed model uses a B-spline basis to generate an overcomplete system with varying degrees of smoothness systematically. For example, the B-spline basis functions of degrees 0, 1, and 2 or more are for jumps, sharp peaks, and smooth parts of the function, respectively.

We consider the mean function can be expressed as a random finite sum:

$$\eta(x) = \sum_{k \in S} \sum_{1 \leq l \leq J_k} B_k(x; \boldsymbol{\xi}_{k,l}) \beta_{k,l}, \tag{2.7}$$

where S denotes the subset of degree numbers of B-spline basis and $B_k(x; \boldsymbol{\xi}_k)$ is a B-spline basis of degree k with knots, $\boldsymbol{\xi}_k \in \mathcal{X}^{(k+2)} := \Omega$. Generating functions of the LARK model are replaced by the B-spline basis functions. J_k has a Poisson distribution with $M_k > 0$ and $\{(\beta_{k,l}, \boldsymbol{\xi}_{k,l})\} \overset{iid}{\sim} \pi_k(d\beta_k, d\boldsymbol{\xi}_k) := \nu_k(d\beta_k, d\boldsymbol{\xi}_k)/\nu_k(\mathbb{R} \times \Omega)$. In this chapter, we assume

$$\pi_k(d\beta_k, d\boldsymbol{\xi}_k) = \mathcal{N}(\beta_k; 0, \phi_k^2)\, d\beta_k \cdot \mathcal{U}(\boldsymbol{\xi}_k; \mathcal{X}^{(k+2)}) d\boldsymbol{\xi}_k.$$

The mean function can be also defined as

$$\eta(x) \equiv \sum_{k \in S} \int_{\Omega} B_k(x; \boldsymbol{\xi}_k) L_k(d\boldsymbol{\xi}_k). \tag{2.8}$$

The stochastic integral representation of the mean function is determined by

$$L_k \sim \text{Lévy}(\nu_k(d\beta_k, d\boldsymbol{\xi}_k)), \quad \forall k \in S,$$

where $\nu_k(d\beta_k, d\boldsymbol{\xi}_k)$ is a finite Lévy measure satisfying $M_k \equiv \nu_k(\mathbb{R} \times \Omega) < \infty$. Although the Lévy measure ν_k satisfying (2.1) may be infinite, the Poisson integrals and sums above are well defined for all bounded measurable compactly-supported $B_k(\cdot, \cdot)$ for which for all $k \in S$,

$$\int \int_{\mathbb{R}\times\Omega} (1 \wedge |\beta_k B_k(\cdot; \boldsymbol{\xi}_k)|) \nu_k(d\beta_k, d\boldsymbol{\xi}_k) < \infty. \tag{2.9}$$

In this chapter, we consider only finite Lévy measures in the proposed model. In other words, we restrict our attention to the Lévy measure of a compound Poisson process. The proposed model is more complex than the LARK model with one kernel and is expected to give a more accurate estimate of the regression function. It can estimate a mean function having both smooth and peak shapes. The proposed model can write in hierarchical form as

$$\begin{aligned}
Y_i|x_i &\overset{ind}{\sim} \mathcal{N}(\eta(x_i), \sigma^2), \quad i = 1, 2, \cdots, n,\\
\eta(x) &= \beta_0 + \sum_{k\in S} \sum_{1\le l\le J_k} B_k(x; \boldsymbol{\xi}_{k,l})\beta_{k,l},\\
\sigma^2 &\sim \text{IG}\left(\frac{r}{2}, \frac{rR}{2}\right),\\
J_k &\sim \text{Poi}(M_k),\\
M_k &\sim \text{Ga}(a_{\gamma_k}, b_{\gamma_k}),\\
\beta_{k,l} &\overset{iid}{\sim} \mathcal{N}(0, \phi_k^2), \quad l = 1, 2, \cdots, J_k,\\
\boldsymbol{\xi}_{k,l} &\overset{iid}{\sim} \mathcal{U}(\mathcal{X}^{(k+2)}), \quad l = 1, 2, \cdots, J_k,
\end{aligned} \tag{2.10}$$

for $k \in S$. We set $\beta_0 = \overline{Y}$ and $\phi_k = 0.5 \times (\max_i\{Y_i\} - \min_i\{Y_i\})$.

2.3.3 Support of LABS model

In this section, we present three theorems on the support of the LABS model. We first define the modulus of smoothness and Besov spaces.

Definition 2.3.1. *Let $0 < p \leq \infty$ and $h > 0$. For $f \in L^p(\mathcal{X})$, the rth order modulus of smoothness of f is defined by*

$$\omega_r(f,t)_p := \sup_{h \leq t} \|\Delta_h^r f\|_p,$$

where $\Delta_h^r f(x) = \sum_{k=0}^{r} \binom{r}{k}(-1)^{r-k} f(x+kh)$ for $x \in \mathcal{X}$ and $x + kh \in \mathcal{X}$.

If $r = 1$, $\omega_1(f,t)_p$ is the modulus of continuity. There exist equivalent definitions in defining Besov spaces. We follow DeVore and Lorentz (1993)[2.10, page 54].

Definition 2.3.2. *Let $\alpha > 0$ be given and let r be the smallest integer such that $r > \alpha$. For $0 < p, q < \infty$, the Besov space $\mathbb{B}^{\alpha}_{p,q}$ is the collection of all functions $f \in L_p(\mathcal{X})$ such that*

$$|f|_{\mathbb{B}^{\alpha}_{p,q}} = \left(\int_0^{\infty} [t^{-\alpha}\omega_r(f,t)_p]^q \frac{dt}{t}\right)^{1/q}$$

is finite. The norm on $\mathbb{B}^{\alpha}_{p,q}$ is defined as

$$\|f\|_{\mathbb{B}^{\alpha}_{p,q}} = \|f\|_p + |f|_{\mathbb{B}^{\alpha}_{p,q}}.$$

The Besov space is a general function space depending on the smoothness of functions in $L_p(\mathcal{X})$ and especially can allow smoothness of spatially inhomogeneous functions, including spikes and jumps. The Besov space has three parameters, α, p, and q, where α is the degree of smoothness, p represents that $L_p(\Omega)$ is the function space where smoothness is measured, and q is a parameter for a finer tuning on the degree of smoothness.

Theorem 2.3.1. *For fixed $k \in S$ and $\boldsymbol{\xi}_k \in \mathcal{X}^{(k+2)}$, the B-spline basis $B_k(x; \boldsymbol{\xi}_k)$ falls in $\mathbb{B}^{\alpha}_{p,q}(\mathcal{X})$ for all $1 \le p, q < \infty$ and $\alpha < k + 1/p$.*

The proof is given in Appendix A.1. For instance, the B-spline basis with degree 0 satisfies $B_k(\cdot, \boldsymbol{\xi}_k) \in \mathbb{B}^{\alpha}_{p,q}$ for $\alpha < 1/p$, the B-spline basis with degree 1 is in $\mathbb{B}^{\alpha}_{p,q}$ for $1 + 1/p$ and the B-spline basis with degree 2 falls in $\mathbb{B}^{s}_{p,q}$ for $2 + 1/p$.

The following theorem describes the mean function of the LABS model, η, is in a Besov space with smoothness corresponding to degrees of B-spline bases used in the LABS model. The proof of the theorem closely follows that of Wolpert et al. (2011). The proof of Theorem 2.3.2 is given in Appendix A.1.

Theorem 2.3.2. *Suppose $\mathcal{X}$ is a compact subset of $\mathbb{R}$. Let ν_k be a Lévy measure on $\mathbb{R} \times \mathcal{X}^{(k+2)}$ that satisfies the following integrability condition,*

$$\int\int_{\mathbb{R}\times\mathcal{X}^{(k+2)}} (1 \wedge |\beta_k|)\nu_k(d\beta_k, d\boldsymbol{\xi}_k) < \infty. \tag{2.11}$$

and $L_k \sim$ Lévy(ν_k) for all $k \in S$. Define the mean function of the LABS model, $\eta(\cdot) = \sum_{k\in S} \int_{\mathcal{X}^{(k+2)}} B_k(x; \boldsymbol{\xi}_k) L_k(d\boldsymbol{\xi}_k)$ on $\mathcal{X}$ where $B_k(x; \boldsymbol{\xi}_k)$ satisfies (2.11) for each fixed $x \in \mathcal{X}$. Then, η has the convergent series

$$\eta(x) = \sum_{k\in S} \sum_{l} B_k(x; \boldsymbol{\xi}_{k,l})\beta_{k,l} \tag{2.12}$$

where S is a finite set including degree numbers of B-spline basis. Furthermore, η lies in the Besov space $\mathbb{B}^{\alpha}_{p,q}(\mathcal{X})$ with $\alpha < \min(S) + \frac{1}{p}$ almost surely.

For example, if a zero element is included in S then the mean function of the LABS, η falls in $\mathbb{B}^{\alpha}_{p,q}$ with $\alpha < \frac{1}{p}$ almost surely, which consists of functions no longer continuous. If $S = \{3, 5, 8\}$, then, η belongs to $\mathbb{B}^{\alpha}_{p,q}$ with $\alpha < 3 + \frac{1}{p}$ almost surely. Moreover, it is highly possible that the function spaces for the LABS model may be larger than those of the LARK model using one type of kernel function. Specifically, the mean function for the LABS model with $S = \{0, 1\}$ falls in $\mathbb{B}^{\alpha}_{p,p}$ with $\alpha < \frac{1}{p}$ almost surely. If that of the LARK model using only one Laplacian kernel falls in $\mathbb{B}^{\alpha}_{p,p}$ with $\alpha < 1 + \frac{1}{p}$, then the function spaces of the LABS model with given $\alpha < \frac{1}{p}$ are larger than those of the LARK model for the range of smoothness parameter, $\frac{1}{p} < \alpha < 1 + \frac{1}{p}$ by the properties of the Besov space.

The next theorem shows that the prior distribution of our LABS model has sufficiently large support on the Besov space $\mathbb{B}^{\alpha}_{p,q}$ with $1 \leq p, q < \infty$ and $\alpha > 0$. For $\eta_0 \in \mathbb{B}^{\alpha}_{p,q}(\mathcal{X})$, denote the ball around η_0 of radius δ,

$$\bar{b}_{\delta}(\eta_0) = \{\eta : \|\eta - \eta_0\|_p < \delta\}$$

where $\|\cdot\|_p$ is a L_p norm. The proof of Theorem 2.3.3 is given in Appendix A.1.

Theorem 2.3.3. *Let $\mathcal{X}$ be a bounded domain in $\mathbb{R}$. Let ν_k be a finite measure on $\mathbb{R} \times \mathcal{X}^{(k+2)}$ and $L_k \sim Levy(\nu_k)$ for all $k \in S$. Suppose η has a prior Π for the LABS model* (2.10)*. Then, $\Pi(\bar{b}_{\delta}(\eta_0)) > 0$ for every $\eta_0 \in \mathbb{B}^{\alpha}_{p,q}(\mathcal{X})$ and all $\delta > 0$.*

2.4 Algorithm

Based on the prior specifications and the likelihood function, the joint posterior distribution of the LABS model (2.10) is

$$
\begin{aligned}
[\boldsymbol{\beta}, \boldsymbol{\xi}, \boldsymbol{J}, \boldsymbol{M}, \sigma^2 \,|\, \boldsymbol{Y}] &\propto [\boldsymbol{Y} \,|\, \eta, \sigma^2] \times [\boldsymbol{\beta}, \boldsymbol{\xi} \,|\, \boldsymbol{J}] \times [\boldsymbol{J} \,|\, \boldsymbol{M}] \times [\boldsymbol{M}] \times [\sigma^2] \\
&\propto \left[(\sigma^2)^{-n/2} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (Y_i - \beta_0 - \sum_{k \in S} \sum_{l=1}^{J_k} B_k(x_i; \boldsymbol{\xi}_{k,l}) \beta_{k,l})^2 \right\} \right] \\
&\times \prod_{k \in S} \left[\exp\left\{ -\frac{1}{2\sigma_k^2} \sum_{l=1}^{J_k} \beta_{k,l}^2 \right\} \right] \times \prod_{k \in S} \left[\frac{1}{|\mathcal{X}^{(k+2)}|^{J_k}} \prod_{l=1}^{J_k} I(\boldsymbol{\xi}_{k,l} \in \mathcal{X}^{(k+2)}) \right] \\
&\times \prod_{k \in S} \left[\frac{M_k^{J_k}}{J_k!} \exp\{-M_k\} \right] \times \prod_{k \in S} \left[M_k^{a_{\gamma_k} - 1} \exp\{-b_{\gamma_k} M_k\} \right] \\
&\times \left[(\sigma^2)^{-\frac{r}{2}+1} \exp\left\{ -\frac{rR}{2\sigma^2} \right\} \right]. \qquad (2.13)
\end{aligned}
$$

The parameters $\boldsymbol{\beta}$ and $\boldsymbol{\xi}$ of the LABS model have varying dimensions as J_k is a random variable. We use the Reversible Jump Markov Chain Monte Carlo (RJMCMC) algorithm (Green, 1995) for the posterior computation.

We consider three transitions in the generation of posterior samples: (a) the addition of basis functions and coefficients; (b) the deletion of basis functions and coefficients; (c) the relocation of knots which affects the shape of basis functions and coefficients. Note that in step (c), the numbers of basis functions and coefficients do not change. We call such move types birth step, death step, and relocation step, respectively. A type of move is determined with probabilities p_b, p_d and p_w with $p_b + p_d + p_w = 1$, where p_b, p_d and p_w are probabilities of choosing the birth, death, and relocation steps, respectively.

Let us denote $\theta_{k,l} = (\beta_{k,l}, \boldsymbol{\xi}_{k,l})$ by an element of $\boldsymbol{\theta}_k = \{\theta_{k,1}, \theta_{k,2}, \ldots, \theta_{k,j}, \ldots, \theta_{k,J_k}\}$, where each $\boldsymbol{\xi}_{k,l}$ has the $(k+2)$ dimensions. In general, the acceptance ratio of the RJMCMC can be expressed as

$$A = \min\left[1, (\text{likelihood ratio}) \times (\text{prior ratio}) \times (\text{proposal ratio}) \times (\text{Jacobian})\right].$$

In our problem the acceptance ratio for each move types is given by

$$A = \min\left[1, \frac{L(\mathbf{Y}|\boldsymbol{\theta}'_k, J'_k)\,\Pi(\boldsymbol{\theta}'_k|J'_k)\Pi(J'_k)q(\boldsymbol{\theta}_k|\boldsymbol{\theta}'_k)}{L(\mathbf{Y}|\boldsymbol{\theta}_k, J_k)\,\Pi(\boldsymbol{\theta}_k|J_k)\Pi(J_k)q(\boldsymbol{\theta}'_k|\boldsymbol{\theta}_k)}\right], \tag{2.14}$$

where $\boldsymbol{\theta}_k$ and J_k refer to the current model parameters and the number of basis functions in the current state, $\boldsymbol{\theta}'_k$ and J'_k denote the proposed model parameters and the number of basis functions of the new state. Here, the Jacobian is 1 in all move types. $q(\boldsymbol{\theta}'_k|\boldsymbol{\theta}_k)$ is the jump proposal distribution that proposes a new state $\boldsymbol{\theta}'_k$ given a current state $\boldsymbol{\theta}_k$. Specifically, we choose the following jump proposal density proposed by Lee et al. (2020) for each move step:

$$\begin{aligned}
q_b(\boldsymbol{\theta}'_k|\boldsymbol{\theta}_k) &= p_b \times b(\theta_{k,J_k+1}) \times \frac{1}{J_k+1}, \\
q_b(\boldsymbol{\theta}'_k|\boldsymbol{\theta}_k) &= p_d \times \frac{1}{J_k}, \\
q_w(\boldsymbol{\theta}'_k|\boldsymbol{\theta}_k) &= p_w \times q(\theta'_{k,r}|\theta_{k,r}),
\end{aligned}$$

where $b(\cdot)$ is a candidate distribution which proposes a new element. For death and change steps, a randomly chosen rth element of $\boldsymbol{\theta}_k$ is deleted and rearranged, respectively. The details regarding updating schemes of each move step are as follows.

(a) **[Birth step]** Assume that the current model is composed of J_k basis functions. If the birth step is selected, a new basis function $B_{k,J_{k+1}}$ and

$\theta_{k,J_{k+1}}$ is accepted with the acceptance ratio

$$\min\left[1, \frac{L(\mathbf{Y}|\boldsymbol{\theta}'_k, J'_k)}{L(\mathbf{Y}|\boldsymbol{\theta}_k, J_k)} \times \frac{\pi(\theta_{k,J_k+1})M_k}{(J_k+1)} \times \frac{p_d/(J_k+1)}{(p_b \times b(\theta_{k,J_k+1}))/(J_k+1)}\right].$$

Especially, a coefficient $\beta_{k,J_{k+1}}$ and an ordered knot set $\boldsymbol{\xi}_{k,J_{k+1}}$ are drawn from their generating distributions and added at the end of $(\beta_{k,1}, \ldots, \beta_{k,J_k})$ and $(\boldsymbol{\xi}_{k,1}, \ldots, \boldsymbol{\xi}_{k,J_k})$. When $J_k = 0$, the birth step must be exceptionally selected until J_k becomes one.

(b) [**Death step**] If the death step is selected, an rth element, $\theta_{k,r}$ uniformly chosen is removed from the existing set of basis functions, coefficients, and ordered knot sets. We can find out the acceptance ratio for a death step similarly. The acceptance ratio is given by

$$\min\left[1, \frac{L(\mathbf{Y}|\boldsymbol{\theta}'_k, J'_k)}{L(\mathbf{Y}|\boldsymbol{\theta}_k, J_k)} \times \frac{J_k}{\pi(\theta_{k,r})M_k} \times \frac{(b(\theta_{k,r}) \times p_b)/J_k}{p_d/J_k}\right].$$

(c) [**Relocation step**] Unlike the other steps, the relocation step keeps the numbers of basis functions or coefficients or ordered knot sets fixed. Therefore, the updating scheme of this step is a Metropolis-Hastings within Gibbs sampler. If the relocation step is selected, a current location $\theta_{k,r}$ is moved to a new location $\theta'_{k,r}$ generated by proposal distributions with the acceptance ratio (2.15). Particularly, since knots of basis function must be in non-descending order, $\xi_{k,r,i}$ which is the ith element of an ordered knot set is sequentially replaced with a new knot location $\xi'_{k,r,i}$ generated by $\mathcal{U}(\xi_{k,r,i-1}, \xi_{k,r,i+1}), i = 1, \ldots, (k+2)$, where $\xi_{k,r,0}$ and $\xi_{k,r,k+1}$ are boundary points of $\mathcal{X}$. That is, each element of a specific knot set $\boldsymbol{\xi}_{k,r} = (\xi_{k,r,1}, \dots, \xi_{k,r,k+2})$ is moved to new knot locations $\boldsymbol{\xi}'_{k,r} = (\xi'_{k,r,1}, \dots, \xi'_{k,r,k+2})$ in turn. The corresponding acceptance ratio is given by

$$\min\left[1, \frac{L(\mathbf{Y}|\boldsymbol{\theta}'_k, J'_k)}{L(\mathbf{Y}|\boldsymbol{\theta}_k, J_k)} \times \frac{\pi(\theta'_{k,r})}{\pi(\theta_{k,r})} \times \frac{q_w(\theta_{k,r}|\theta'_{k,r})}{q_w(\theta'_{k,r}|\theta_{k,r})}\right]. \tag{2.15}$$

When using an independent proposal distribution (i.e. $q_w(\theta'_{k,r}|\theta_{k,r}) = \pi(\theta'_{k,r})$), the acceptance ratio can reduce to

$$\min\left[1, \frac{L(\mathbf{Y}|\boldsymbol{\theta}'_k, J'_k)}{L(\mathbf{Y}|\boldsymbol{\theta}_k, J_k)}\right].$$

Finally, $\beta'_{k,r}$ is sampled from its conditional posterior distribution by using the Gibbs sampling.

The posterior samples of σ and M_k can be generated from their conditional posterior distributions. See Appendix A.1. The pseudo-code for the proposed strategy is given in Algorithm 1.

2.5 Simulation studies

In this section, we evaluate the performance of the LABS model (2.10) and competing methods on simulated data sets. First, we apply the proposed method to four standard examples: Bumps, Blocks, Doppler, and Heavisine test functions introduced by Donoho and Johnstone (1994). Second, we consider three functions that we created ourselves with jumps and peaks to assess the practical performance of the proposed model.

Algorithm 1 A reversible jump MCMC algorithm for LABS

```
procedure LABS($S$)                                          ▷ $S$: set of degree numbers
    Initialize parameters $\boldsymbol{J}, \boldsymbol{\beta}, \boldsymbol{\xi}, \boldsymbol{M}, \sigma^2$ from prior distributions.
    for iteration $i = 1$ to $N$ do
        for $k = 1$ to $|S|$ do                              ▷ $\boldsymbol{J} := \{\boldsymbol{J}_1, \ldots, \boldsymbol{J}_k, \ldots, \boldsymbol{J}_{|S|}\}$
            Update $(\boldsymbol{J}_k, \boldsymbol{\beta}_k, \boldsymbol{\xi}_k)$ through a reversible jump MCMC.
            Sample $M_k$ from the full conditional $\pi(M_k|\text{others})$.        ▷ Gibbs step
        end for
        Sample $\sigma^2$ from the full conditional $\pi(\sigma^2|\boldsymbol{\beta}, \boldsymbol{\xi}, \boldsymbol{J}, \boldsymbol{M}, \boldsymbol{y})$.    ▷ Gibbs step
        Store $i$th MCMC samples.
    end for
end procedure
```

The simulated data sets are generated from equally spaced x's on $\mathcal{X} = [0, 1]$ with sample sizes $n = 128$ and 512. Independent normally distributed noises $\mathcal{N}(0, \sigma^2)$ are added to the true function $\eta(\cdot)$. The root signal-to-noise ratio (RSNR) is defined as

$$\text{RSNR} := \sqrt{\frac{\int_{\mathcal{X}} (f(x) - \bar{f})^2 \, dx}{\sigma^2}},$$

where $\bar{f} := \frac{1}{|\mathcal{X}|} \int_{\mathcal{X}} f(x) \, dx$ and set at 3, 5 and 10. We also run the LABS model for 200,000 iterations, with the first 100,000 iterations discarded as burn-in and retain every 10th sample. For comparison between the methods, we compute the mean squared error of all methods using 100 replicate data sets for each test function. The average of the posterior curves is used for the estimate of the test function.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (\eta(x_i) - \hat{\eta}(x_i))^2.$$

2.5.1 Simulation 1 : DJ test functions

We carry out a simulation study using the benchmark test functions suggested by Donoho and Johnstone (1994), often used in the field of wavelet and nonparametric function estimation. The Donoho and Johnstone test functions consist of four functions called Bumps, Blocks, Doppler, and Heavisine. These test functions are composed of various shapes such as sharp peaks (Bumps), jump discontinuities (Blocks), oscillating behavior (Doppler), and jumps/peaks in smooth functions (Heavisine) (See Figure 2.3).

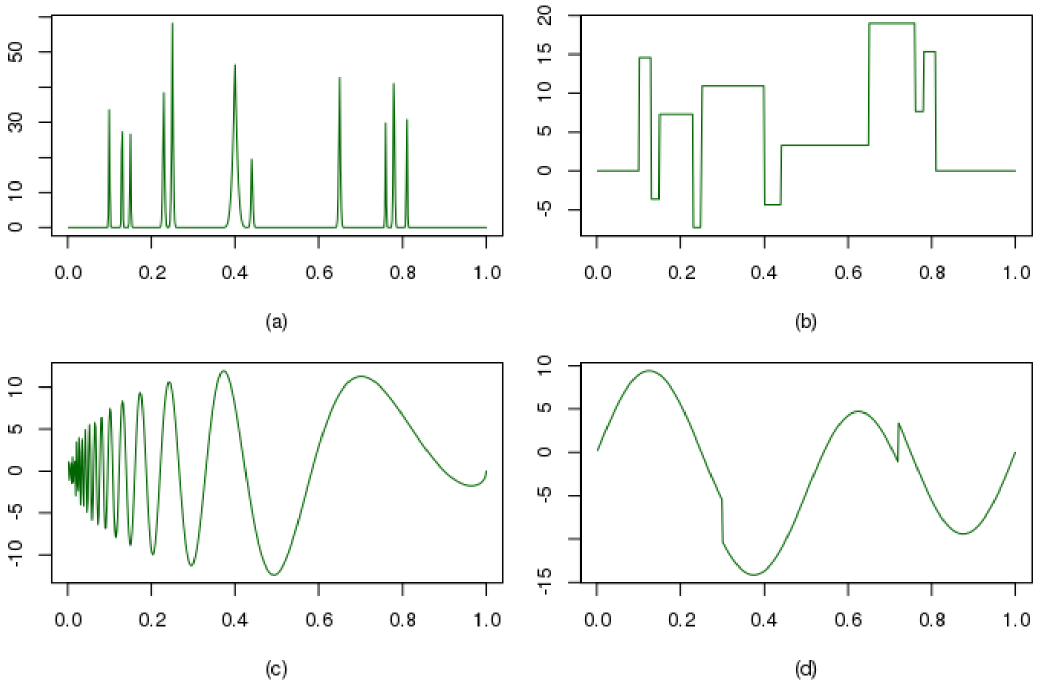


Figure 2.3: The Donoho and Johnstone test functions: (a) Bumps, (b) Blocks, (c) Doppler and (d) Heavisine

The hyperparameters and types of basis functions used in (2.10) are displayed in Table 2.1. For Bumps and Doppler, the parameter r of the prior distribution for σ^2 is set to 100 to speed up convergence. We also take account of the combinations of a B-spline basis based on the shapes of test functions.

	S	r	R	a_{γ_k}	b_{γ_k}
Bumps	{1}	100	0.01	1	1
Blocks	{0}	0.01	0.01	1	1
Doppler	{1,2}	100	0.01	1	1
Heavisine	{0,2}	0.01	0.01	1	1

Table 2.1: The values of hyperparameters of proposed model for each test function

We compare our model with a variety of methods such as B-spline curve of degree 2 with 50 knots (denoted as BSP-2), Local polynomial regression with automatic smoothing parameter selection (denoted by LOESS), Smoothing spline with smoothing parameter selected by cross-validation (denoted by SS), Nadaraya–Watson kernel regression using the Gaussian kernel with bandwidth h which minimizes CV error (denoted by NWK), Empirical Bayes approach for wavelet shrinkage using a Laplace prior with Daubechies "least asymmetric" (la8) wavelets except for the Blocks example, where it uses the Haar wavelet; Johnstone and Silverman (2005) (denoted by EBW), Trend filtering with order # based on a optimal regularization parameter; Tibshirani et al. (2014) (denoted by TF-#), Gaussian process regression with the Radial basis or Laplacian kernel (denoted by GP-R or GP-L), Bayesian curve fitting using piecewise polynomials with $l = \#1, l_0 = \#2$; Denison et al. (1998a) (denoted by BPP-#1-#2), Bayesian adaptive spline surfaces with degree #; Francom et al. (2018) (denoted by BASS-#), and Lévy adaptive regression with multiple kernels; Lee et al. (2020) (denoted by LARMuK). These competitive models are implemented in R (R Core Team, 2020) with various packages: LOESS (Wang, 2010), Empirical Bayes thresholding (Silverman et al., 2005), Gaussian process (Karatzoglou et al., 2004), Bayesian curve fitting using piecewise polynomials (Feng, 2013), and Bayesian adaptive spline surfaces (Francom and Sanso, 2016).

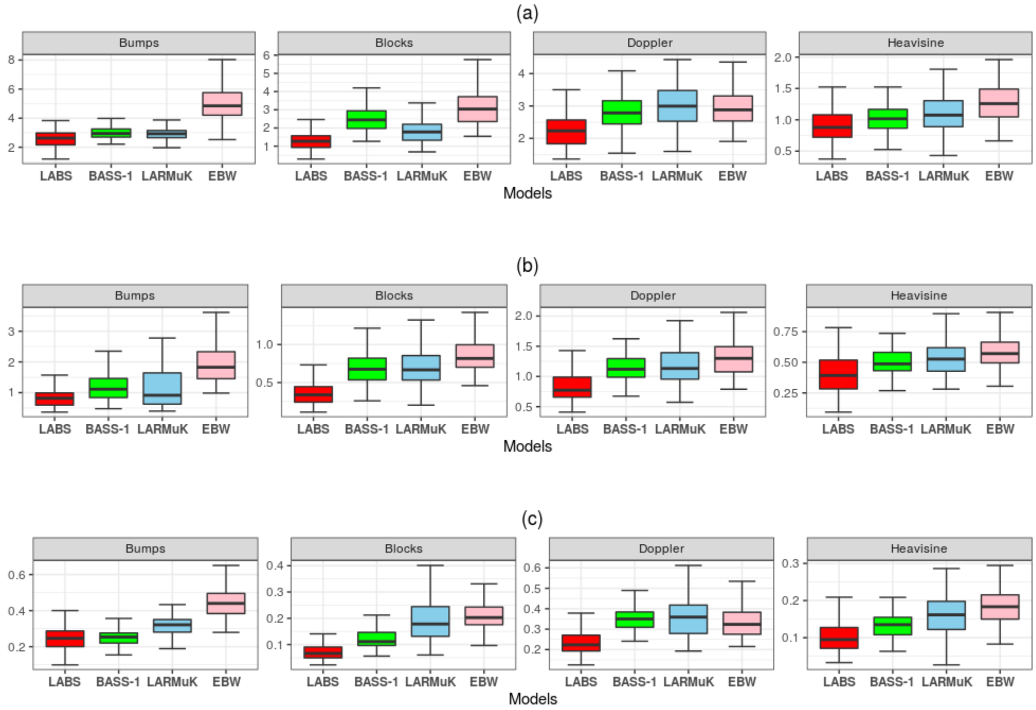


Figure 2.4: Boxplots of MSEs from the simulation study with $n = 128$ and RSNR = (a) 3, (b) 5 and (c) 10

Both Figure 2.4 and Figure 2.5 show that the performance of our model is generally more accurate than other methods. The models in the two figures are selected by better outcomes from simulations. More detailed simulation results can be seen in Appendix A.1. Figure 2.4 shows that the LABS model is superior to others regardless of noise levels with $n = 128$. It also has the smallest average mean square error for all test functions except the Heavisine example with RSNR = 3. Similarly, for sample size $n = 512$, the LABS model comes up with the best performance in Figure 2.5 except for the Doppler function, where it is competitive. Our model removes high frequencies in the interval $[0, 0.1]$ and produces a smooth curve within the corresponding domain. On the contrary, due to a small number of data points in the Doppler example with $n = 128$, most models yield similar smooth curves in $[0, 0.1]$. As a result, the LABS model has an excellent numerical performance. For Blocks example, LABS, in particular, yields the lowest average and standard deviation of the mean squared error in all scenarios. This suggests that our model has an excellent ability to find jump points. Furthermore, LABS has consistently better performance than B-spline regression using only one basis function for four simulated data sets since its overcomplete systems can be constructed by various combinations of B-spline basis functions. See Appendix A.1.

2.5.2 Simulation 2 : Smooth functions with jumps and peaks

Our main interest lies in estimating smooth functions with either discontinuity such as jumps, or sharp peaks or both. We design three test functions to assess the practical performance of the proposed method for our concerns. The first and second example is modified by adding some smooth parts, unlike the original version of the Bumps and Blocks of DJ test functions. Each test

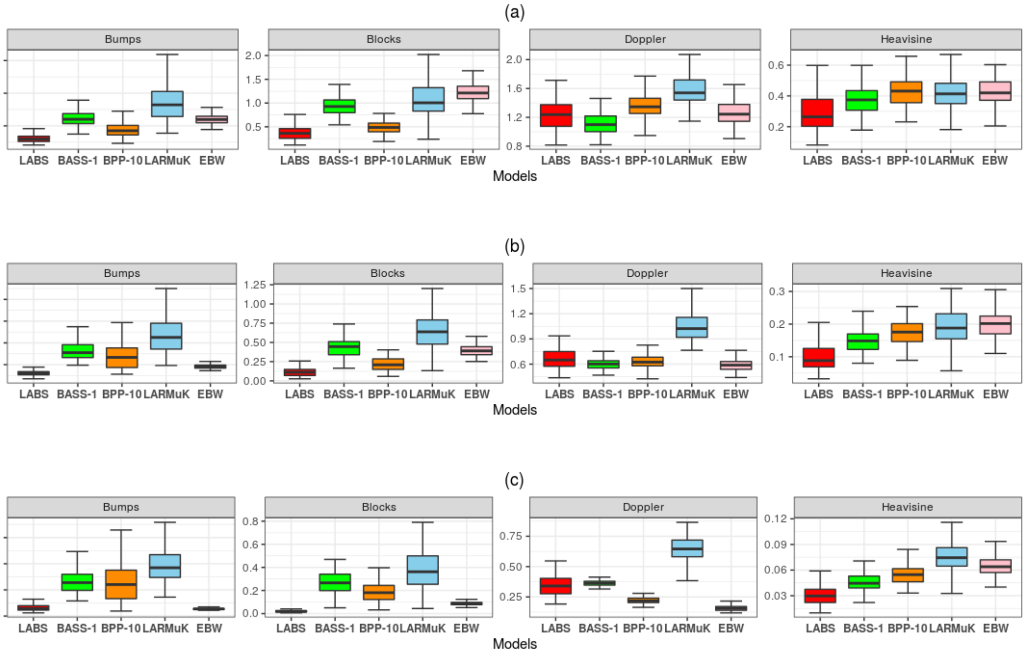


Figure 2.5: Boxplots of MSEs from the simulation study with $n = 512$ and RSNR = (a) 3, (b) 5 and (c) 10

function provided is given by

$$\begin{aligned}\eta_1(x) =& \frac{0.6}{0.92}[4\text{ssgn}(x-0.1) - 5\text{ssgn}(x-0.13) + 5\text{ssgn}(x-0.25) - 4.2\text{ssgn}(x-0.4)\\ &+ 2.1\text{ssgn}(x-0.44) + 4.3\text{ssgn}(x-0.65) - 4.2\text{ssgn}(x-0.81) + 2] + 0.2 + \sin(8\pi x),\\ \eta_2(x) =& [7K_{0.005}(x-0.1) + 5K_{0.07}(x-0.25) + 4.2K_{0.03}(x-0.4) + 4.3K_{0.01}(x-0.65)\\ &+ 5.1K_{0.008}(x-0.78) + 3.1K_{0.1}(x-0.9)] + \cos(4\pi x),\end{aligned}$$

where $\text{sgn}(x) = \mathrm{I}_{(0,\infty)}(x) - \mathrm{I}_{(-\infty,0)}(x)$, $\text{ssgn}(x) = 1 + \text{sgn}(x)/2$ and $K_w(x) := (1+|x/w|)^{-4}$. Finally, we create a sum of jumps, peaks, and some smoothness.

A formula for a final test function is

$$\begin{aligned}\eta_3(x) &= 6\sin(4\pi x) + 7(1 + \text{sgn}(x - 0.1)/2) - 7(1 + \text{sgn}(x - 0.18)/2) \\ &\quad - 2\text{sgn}(x - 0.37) + 17K_{0.01}(x - 0.5) - 3\text{sgn}(x - 0.72) + 10K_{0.05}(x - 0.89).\end{aligned}$$

They are displayed in Figure 2.6. We call in turn them modified Blocks, modified Bumps, and modified Heavisine, respectively.

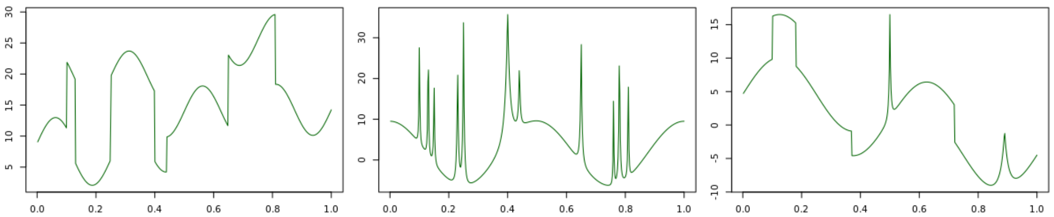


Figure 2.6: Three test functions used in the second simulation: modified Blocks (left), modified Bumps (center) and modified Heavisine (right)

In these experiments, we use two or more types of B-spline basis as elements of overcomplete systems since three functions have different shapes, unlike previous simulation studies. Hyperparameters are similar to the previous ones. All hyperparameters for the prior distributions are summarized in Table 2.2. Again, we only compare our model with BPP, BASS, EBW, TF, and LARMuK models, which have relatively good performance in some test functions of Simulation 1.

Table 2.3 furnishes that the LABS model has the best outcomes when the sample size is 128, difficult to estimate. Furthermore, when $n = 512$, we find out from Table 2.4 that the LABS model performs well in most cases with either the lowest or the second-lowest average MSE values across 100 replicates. In

	S	r	R	a_{γ_k}	b_{γ_k}
Modified Blocks	{0,2,3}	0.01	0.01	1	1
Modified Bumps	{1,2}	50	0.01	1	1
Modified Heavisine	{0,1,2,3}	0.01	0.01	5	1

Table 2.2: Details of hyperparameters of the LABS used in second experiment

particular, the LABS outperforms competitors in modified Blocks, irrespective of the sample size and noise levels as expected. Among all models, the worst-performing method is the BASS-2 since it cannot estimate well many jumps or peak points for given test functions. Figure 2.7 supports that the LABS model has the ability to overcome the noise and adapt to smooth functions with

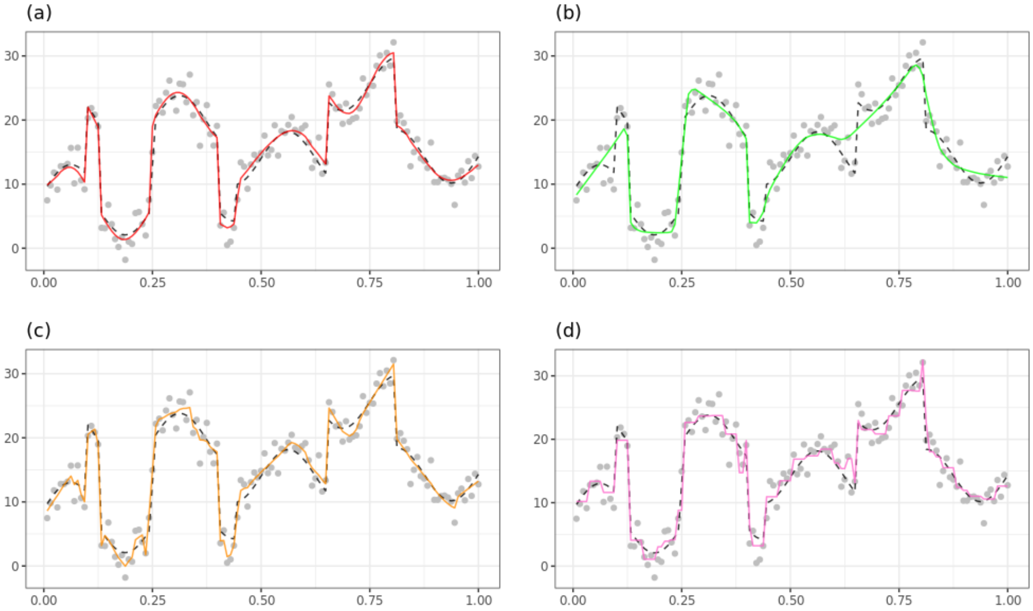


Figure 2.7: Comparisons of the estimates of a data set generated from the modified Blocks with $n = 128$ and RSNR = 3 using (a) LABS, (b) BASS-1, (c) BPP-10, and (d) EBW. Dashed lines represent true curves, solid lines represent estimates of curve.

Model	Modified Blocks			Modified Bumps			Modified Heavisine		
	RSNR=3	RSNR=5	RSNR=10	RSNR=3	RSNR=5	RSNR=10	RSNR=3	RSNR=5	RSNR=10
EBW	3.781(0.7407)	1.538(0.3854)	0.401(0.0684)	3.921(1.0117)	1.557(0.3191)	0.445(0.1043)	2.548(0.4738)	1.326(0.2793)	0.402(0.095)
TF-0	2.174(0.6728)	0.969(0.2857)	0.345(0.0846)	3.606(0.9942)	1.595(0.3653)	0.475(0.0688)	2.272(0.5905)	1.047(0.3202)	0.36(0.107)
TF-1	2.805(0.6289)	1.156(0.2961)	0.323(0.093)	4.378(1.3533)	1.682(0.4021)	0.482(0.0574)	2.791(0.6406)	1.545(0.5612)	0.447(0.3042)
TF-2	3.081(0.835)	1.494(0.2869)	0.534(0.1639)	4.939(1.7152)	1.765(0.4284)	0.474(0.0643)	3.003(0.4896)	1.833(0.5414)	0.752(0.6268)
BPP-10	2.238(0.6436)	0.951(0.2439)	0.367(0.098)	2.949(0.749)	1.351(0.3244)	0.631(0.2488)	2.06(0.645)	0.824(0.2125)	0.287(0.0907)
BPP-21	2.589(0.4787)	1.336(0.2305)	0.985(0.1714)	3.777(0.9094)	2.586(0.6268)	2.39(0.6003)	2.168(0.3821)	1.228(0.2969)	0.825(0.3458)
BASS-1	2.283(0.5013)	0.76(0.2194)	0.172(0.0424)	2.199(0.5625)	0.858(0.1622)	0.276(0.0483)	2.013(0.5502)	0.737(0.198)	0.178(0.0418)
BASS-2	4.038(0.6519)	2.232(0.371)	1.368(0.168)	9.881(1.1796)	7.944(0.7727)	6.999(0.5772)	3.275(0.3734)	2.276(0.3877)	1.378(0.2871)
LARMuK	2.158(0.5735)	0.97(0.2352)	0.298(0.0929)	2.029(0.7688)	0.822(0.2446)	0.271(0.0944)	1.721(0.4757)	0.713(0.1912)	0.219(0.075)
LABS	**1.868(0.5982)**	**0.691(0.2022)**	**0.162(0.044)**	**2.01(0.6006)**	**0.803(0.1491)**	**0.248(0.0457)**	**1.589(0.5081)**	**0.635(0.1727)**	**0.172(0.0472)**

Table 2.3: Average of MSEs over 100 replications for three functions of Simulation 2 with $n = 128$. Estimated standard errors of MSEs are shown in parentheses

Model	Modified Blocks			Modified Bumps			Modified Heavisine		
	RSNR=3	RSNR=5	RSNR=10	RSNR=3	RSNR=5	RSNR=10	RSNR=3	RSNR=5	RSNR=10
EBW	1.525(0.228)	0.644(0.1025)	0.171(0.0245)	1.487(0.2056)	0.595(0.0847)	**0.171(0.0217)**	1.106(0.1523)	0.538(0.0775)	0.153(0.0218)
TF-0	0.734(0.1207)	0.342(0.0573)	0.123(0.0154)	1.605(0.2709)	0.729(0.1123)	0.26(0.0502)	0.778(0.1365)	0.359(0.0579)	0.12(0.0159)
TF-1	1.351(0.2028)	0.642(0.0806)	0.165(0.0221)	1.618(0.4642)	0.769(0.2743)	0.377(0.1101)	0.837(0.1235)	0.432(0.0715)	0.153(0.021)
TF-2	1.544(0.1781)	0.766(0.0775)	0.298(0.0306)	1.831(0.2399)	0.818(0.1013)	0.714(0.1984)	0.973(0.1403)	0.516(0.0717)	0.185(0.0215)
BPP-10	0.608(0.1362)	0.256(0.055)	0.133(0.0412)	1.2(0.1951)	0.623(0.2658)	0.37(0.2954)	0.602(0.123)	0.237(0.0495)	0.079(0.0183)
BPP-21	0.869(0.1552)	0.424(0.0683)	0.283(0.0479)	1.209(0.3056)	0.937(0.3201)	0.789(0.3442)	0.679(0.1367)	0.275(0.0562)	0.134(0.0338)
BASS-1	0.701(0.1489)	0.296(0.0667)	0.124(0.0455)	0.927(0.147)	**0.442(0.0726)**	0.197(0.063)	**0.561(0.126)**	0.246(0.0423)	0.083(0.0166)
BASS-2	4.038(0.6519)	2.232(0.371)	1.368(0.168)	9.881(1.1796)	7.944(0.7727)	6.999(0.5772)	3.275(0.3734)	2.276(0.3877)	1.378(0.2871)
LARMuK	0.775(0.257)	0.416(0.1317)	0.213(0.0841)	1.222(0.4575)	0.805(0.3056)	0.514(0.177)	0.669(0.1749)	0.323(0.0975)	0.14(0.0456)
LABS	**0.583(0.1718)**	**0.234(0.0696)**	**0.071(0.0325)**	**0.919(0.1397)**	0.46(0.0867)	0.194(0.0443)	0.576(0.146)	**0.236(0.0575)**	**0.078(0.024)**

Table 2.4: Average of MSEs over 100 replications for three functions of Simulation 2 with $n = 512$. Estimated standard errors of MSEs are shown in parentheses

either discontinuity such as jumps or sharp peaks, or both.

2.6 Real data applications

We now apply the LABS model (2.10) real-world datasets, including the minimum legal drinking age (MLDA) on mortality rate, the closing bitcoin price index, and the maximum daily value of concentrations of fine particulate matter (PM2.5) in Seoul. All the real data examples exhibit wildly varying patterns that may have jumps or peaks. These fluctuating patterns are expected to further illustrate the features of the LABS model.

In the following applications we set the hyperparameter values of the proposed model, LABS: $a_J = 5$, $b_J = 1$, $r = 0.01$, and $R = 0.01$. In this analysis, we practically choose $S = \{0, 1, 2\}$ because the true curve of real data is unknown, and it may have varying smoothness. We run it 200,000 times with a burn-in of 100,000 and thin by 10 to achieve convergence of the MCMC algorithm. Performance comparisons of our model and some rather good methods in the simulated studies are also conducted.

2.6.1 Example 1: Minimum legal drinking age

The Minimum legal drinking age (MLDA) heavily affects youth alcohol consumption, a sensitive issue worldwide for policymakers. In the past three decades, there have been many studies on the effect of legal access age to alcohol on death rates. The MLDA dataset collected from Angrist and Pischke (2014) contains death rates, a measure of the total number of deaths per 100,000 young Americans per year.

This data has been widely used to estimate the causal effect of policies on the minimum legal drinking age in the area of Regression Discontinuity Design (RDD). Figure 2.8 (a) highlights that the MLDA data might represent a piecewise smooth function with a single jump discontinuity at minimum drinking age of 21, referred to as cutoff in the RDD. Specifically, each observation (or point) in Figure 2.8 corresponds to the death rate from all causes in the monthly interval, and the number of all observations is 48.

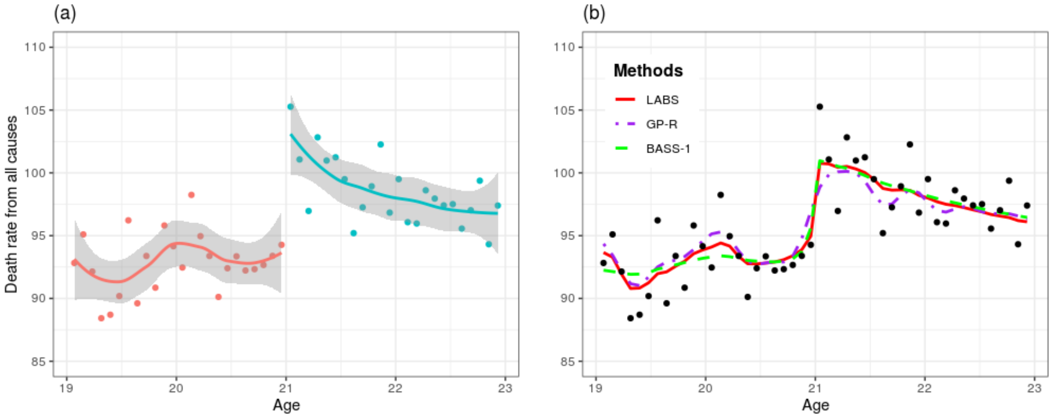


Figure 2.8: (a) A piecewise curve fitting and (b) comparisons of the fitted posterior mean using BASS-1, GP-R and LABS for Minimum legal drinking age (MLDA) dataset

Using the MLDA data, we estimate unknown functions of death rates via LABS and several competing models, including BPP-21, BASS-1, LARMuK, and GP-R. Figure 2.8 (b) shows that three posterior mean estimates for an unknown function. The solid curve denotes LABS, the dotted-dashed curve indicates GP-R, and the dashed curve represents BASS-1.

In Figure 2.8 (b), both LABS and BASS-1 provide similar posterior curves to the piecewise polynomial regression of Figure 2.8 (a). The estimated curves of them also have a jump point at 21. While the estimated function of GP-R is smooth, the mean function for the LABS model has both smooth and jump parts. We calculate the mean squared error with 10-fold cross-validation for comparison between methods. The mean and standard deviation values of cross-validation prediction errors are given in Table 2.5. The lower CV error rate of LABS implies that LABS has a better performance of estimating a smooth function with discontinuous points than the others.

	LABS	BASS-1	BPP-21	LARMuK	GP-R
Mean	**6.5851**	6.7884	8.6643	8.35014	7.25693
Standard Deviation	5.1838	4.98241	6.66641	6.45711	5.23563

Table 2.5: Mean and standard deviation for the error rate of 10-fold cross-validation on MLDA dataset.

2.6.2 Example 2: Bitcoin prices on Bitstamp

Bitcoin is the best-known cryptocurrency based on Blockchain technology. The demands for bitcoin have increased globally because of offering a higher yield and easy access. The primary characteristic of bitcoin is to enable financial transactions from user to user on the peer-to-peer network configuration without a central bank. Unlimited trading methods and smaller market sizes than the stock market lead to high volatility in the bitcoin price. We collected a daily bitcoin exchange rate (BTC vs. USD) on Bitstamp from January 1, 2017, to December 31, 2018. Bitcoin data (sourced from `http://www.bitcoincharts.com`) has 730 observations and eight variables: date, open price (in USD), high price, low price, closing price, volume in bitcoin, volume in currency, and weighted bitcoin price.

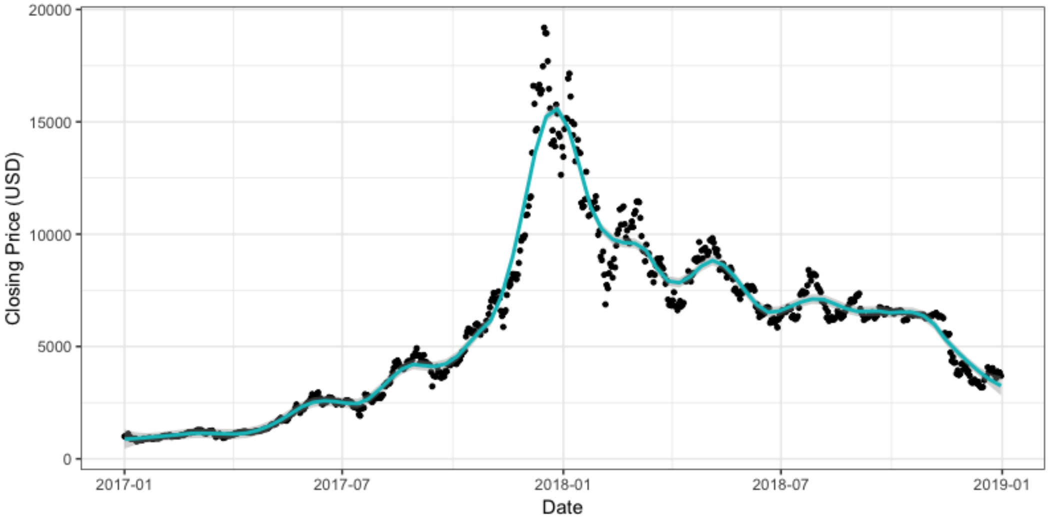


Figure 2.9: Daily bitcoin closing price with a smoothing line

We also add LOESS (locally estimated scatterplot smoothing) regression line to a scatter plot of a daily closing price in Figure 2.9. The dataset shows locally strong upward and downward movements. We apply LABS and other models to estimate the curve of daily bitcoin closing price. Figure 2.10 illustrates the predicted curves of the LABS and competing models for approximating an unknown function of daily bitcoin closing price. There are no significant differences between the estimated posterior curves.

Alternatively, we calculate cross-validation errors to assess model performances. The values of cross-validation errors are given in Table 2.6. Table 2.6 demonstrates that the LABS model provides more accurate function estimation and consistent performance through both minimum mean and relatively low standard deviation values of the cross-validation errors. They also indicate that the Gaussian process is not proper in the cases with locally varying

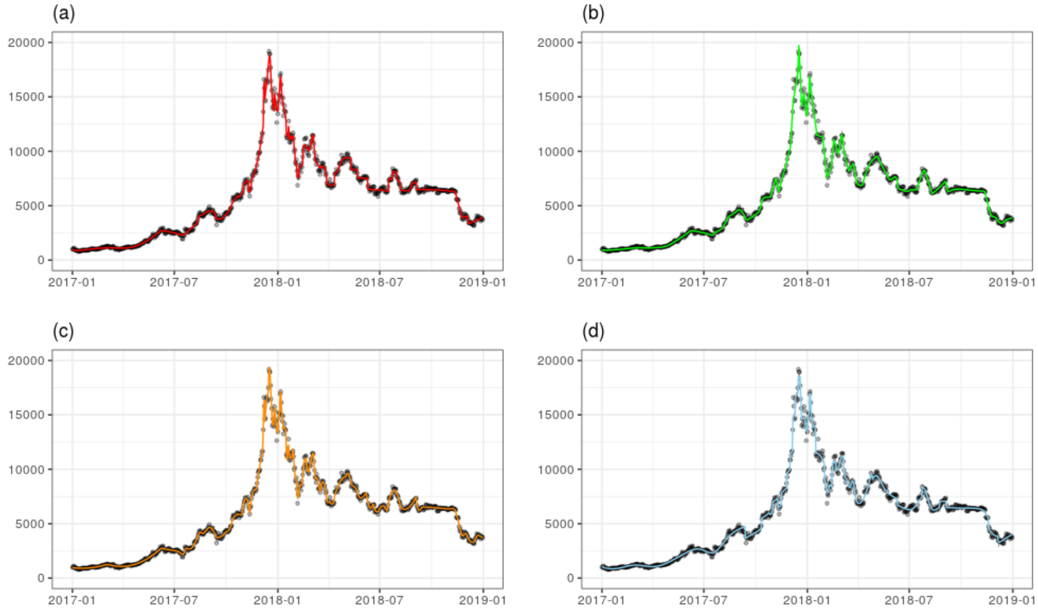


Figure 2.10: Posterior mean of η on Bitcoin dataset using four models: (a) LABS, (b) BASS-1, (c) BPP-21 and (d) LARMuK

smoothness. We can find out that the LABS gives more reliable estimated functions with both discontinuous and smooth parts than other methods.

	LABS	BASS-1	BPP-21	LARMuK	GP-R
Mean	**98014**	109222	99937	149272	583046
Standard Deviation	30057	30052.5	29210	51128.7	137859.7

Table 2.6: Mean and standard deviation for the error rate of 10-fold cross-validation on Bitcoin dataset

2.6.3 Example 3: Fine particulate matter in Seoul

The fine dust has become a national issue, and its forecast received great attention from the media. Much research on fine particulate matter (PM2.5) has been carried out as it gained social attention. According to the studies, Korea's fine dust particles originated from within the country and external sources from China. Many factors cause PM2.5 concentration to rapidly rise or fall and make it difficult to predict its behavior accurately.

We estimate the unknown function of maximum daily concentrations of PM2.5 in Seoul. The PM2.5 dataset collected from the AIRKOREA (`https://www.airkorea.or.kr`) includes 1261 daily maximum values of PM2.5 concentration from January 1, 2015, to June 30, 2018. We removed all observations that have missing values.

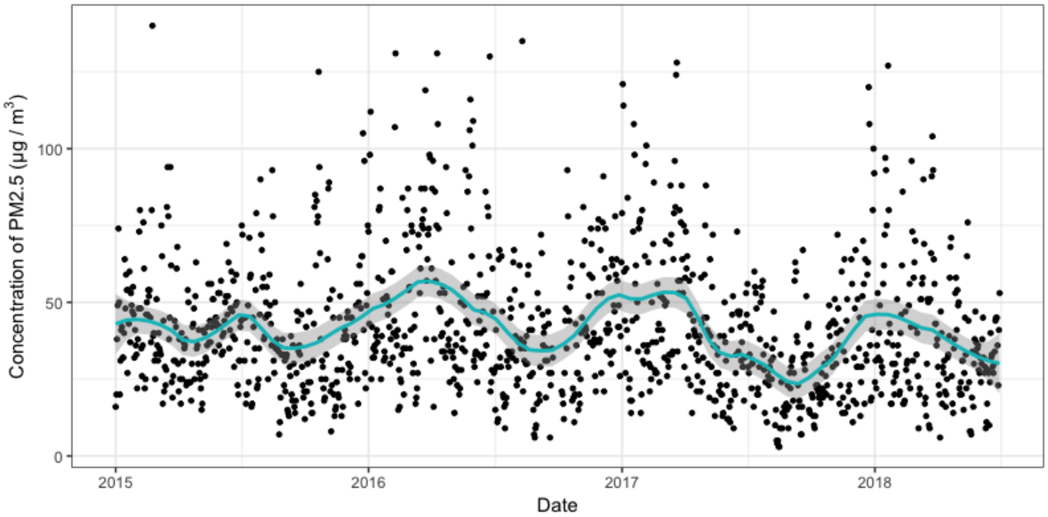


Figure 2.11: Daily maximum concentrations of PM2.5 in Seoul with a smoothing line

Figure 2.11 displays daily fluctuations and seasonality. PM2.5 concentrations are higher in winter and spring than in summer and fall. A LOESS smoothed line added in the figure does not reflect these features well. We take advantage of combinations of basis functions, $S = \{0, 1, 2\}$ to grasp such char-

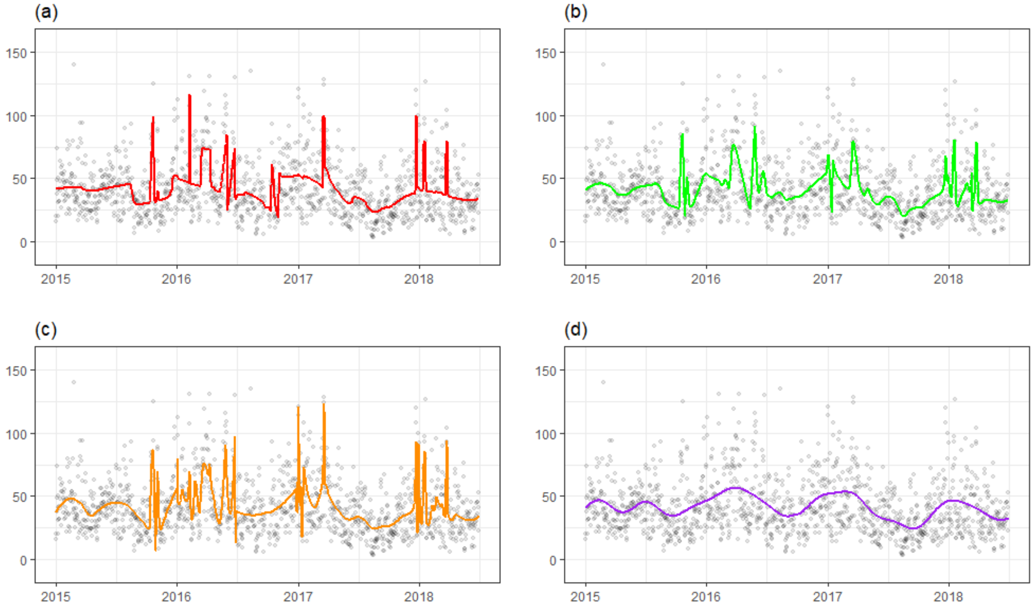


Figure 2.12: Posterior mean of the mean function on PM2.5 dataset using four models: (a) LABS, (b) BASS-1, (c) BPP-10 and (d) GP-R

acteristics of PM2.5 data with multiple jumps and peak points. As shown in Figure 2.12, all four methods represent different estimated lines of the unknown mean function and pick features of the data up in their way. Interestingly, LABS, BASS-1, and BPP-10 react in different ways while they detect peaks, jumps, and smooth parts of PM2.5 data.

We also compute the average and standard deviation of the cross-validated errors of LABS, BPP-10, BASS-1, LARMuK, and GP-R, given in Table 2.7. The LABS model has the lowest cross-validation error among all methods. Moreover, a comparably low standard deviation of LABS supports that it has a more stable performance for estimating any shape of functions due to using all three types of B-spline basis.

	LABS	BASS-1	BPP-10	LARMuK	GP-R
Mean	**384.8863**	393.6049	398.17	399.6718	436.2286
Standard Deviation	56.88069	60.38016	58.63784	53.02499	67.98722

Table 2.7: Mean and standard deviation for the error rate of 10-fold cross-validation on Seoul PM2.5 dataset

2.7 Discussion

We suggested general function estimation methodologies using the B-spline basis function as the elements of an overcomplete system. The B-spline basis can systematically represent functions with varying smoothness since it has nice properties such as local support and differentiability. The overcomplete system and a Lévy random measure enable a function with both continuous and discontinuous parts to capture all features of the unknown regression function. Simulation studies and real data analysis also show that the proposed models perform better than other competing models. We also showed that the prior has full support in certain Besov spaces. The major limitation of the LABS model is the slow mixing of the MCMC algorithm. Future work will develop an efficient algorithm for the LABS model and extend the LABS model for multivariate analysis.

Chapter 3

Bayesian multivariate nonparametric regression using overcomplete systems with tensor products of B-spline bases

3.1 Introduction

In this chapter, we develop a fully Bayesian nonparametric regression with tensor products of B-spline bases based on the Lévy process priors and call it the Multivariate Lévy Adaptive B-Spline regression (MLABS). The MLABS models adaptively as a sum of basis functions. There are three main contributions of this work. First, the proposed method can adapt various smoothness of functions in the multi-dimensional data by changing a set of degrees of the tensor product basis function. Especially, the local support of the B-spline basis can also make more delicate predictions than other existing methods in the non-smooth surface data. Second, Levy process priors encourage sparsity in the expansions and provide automatic selection over the number of basis functions. That is, our model does not suffer from model selection problems like a LABS model. Finally, the MLABS model has comparable performance on regression and classification problems. Empirical results demonstrate that the MLABS has more stable and accurate predictive abilities than state-of-the-art regression models.

The outline of the chapter is as follows. In Section 2, we propose an extended model of LARK and LABS models for multivariate analysis. The posterior computation and details for tensor product bases used in the proposed model are also presented. Simulation experiments comparing the predictive performance of our method with others are provided in Section 3. In Section 4, the proposed model is applied to regression and classification problems using several real-world data sets. We conclude the chapter with a discussion in Section 5.

3.2 Multivariate Lévy adaptive B-spline regression

In this section, we propose an extended model of the LABS model that can only cope with data has one variable for multivariate analyses.

3.2.1 Model specifications

General tensor product B-spline bases require a lot of computations as the number of variables increases. This problem is the so-called "curse of dimensionality", which means computation burden can increase exponentially with dimension. We apply the structure of basis functions of (Bayesian) MARS to that of the LABS model to lessen the computational effort. The idea regarding tensor products of B-spline bases was initially proposed by Bakin et al. (2000). We consider general basis functions without restricted degrees. The MARS model approximates an unknown function as a weighted sum of basis functions that are product of K $(< p)$ univariate spline functions for handling the multi-dimensional or high-dimensional data. It means that it is enough to represent an unknown function by a combination of main effect terms and lower-order interactions.

We first define the jth tensor product of B-spline bases $\mathbf{B} : \mathbb{R}^p \times \Omega' \to \mathbb{R}$ used by a generating function as

$$\mathbf{B}_j(\mathbf{x}_i) := \prod_{l=1}^{K_j} B_{c_l^{(j)}}(x_{i,\nu_l^{(j)}}; \xi_l^{(j)}), \tag{3.1}$$

where $K_j \in \{1, 2, \ldots, K_{\max}\}$ is an interaction order of $\mathbf{B}_j(\mathbf{x}_i)$, $c_l^{(j)} \in S$ is a degree number of univariate B-spline basis, $\nu_l^{(j)} \in \{1, 2, \ldots, p\}$ is an index to determine which a variable is used and $\xi_l^{(j)}$ are a knot sequence on $(\mathcal{X}_{\nu_l^{(j)}})^{(c_l^{(j)}+2)}$, a product space of the $\nu_l^{(j)}$th variable . For the parameters in the jth tensor product of B-spline bases, we write $\mathbf{c}^{(j)} := (c_1^{(j)}, \ldots, c_{K_j}^{(j)})$, $\boldsymbol{\nu}^{(j)} := (\nu_1^{(j)}, \ldots, \nu_{K_j}^{(j)})$ and $\boldsymbol{\xi}^{(j)} := (\xi_1^{(j)}, \ldots, \xi_{K_j}^{(j)})$. We also assume $\boldsymbol{\omega}_j := (\mathbf{c}^{(j)}, \boldsymbol{\nu}^{(j)}, \boldsymbol{\xi}^{(j)})$ and $\boldsymbol{\psi}_j := (K_j, \boldsymbol{\omega}_j) \in \Omega'$, a complete separable metric space. Then, we can rewrite the

jth basis function from $\mathbf{B}_j(\mathbf{x}_i)$ to $\mathbf{B}_j(\mathbf{x}_i; \boldsymbol{\psi}_j)$.

The mean function of the MLABS model can be formulated by

$$f(\mathbf{x}_i) = \beta_0 + \sum_{j=1}^{J} \mathbf{B}_j(\mathbf{x}_i; \boldsymbol{\psi}_j)\beta_j, \quad \mathbf{x}_i \in \mathbb{R}^p, \tag{3.2}$$

where β_0 is an fixed intercept term, J is a Poisson random variable with mean $M > 0$, and $\{\beta_j, \boldsymbol{\psi}_j\}$ are i.i.d from a distribution $\pi(d\beta, d\boldsymbol{\psi}) := \mathcal{N}(d\beta; 0, \phi^2) \cdot \pi(d\boldsymbol{\psi})$. The main different things are the structure of basis functions and the randomness of degrees of B-spline basis. The prespecified degree numbers of the basis functions in $\boldsymbol{\psi}$ are fixed in the LABS model but random in the MLABS model. The mean function (3.2) can also be expressed as a stochastic integral

$$f(\mathbf{x}) := \int_{\Omega'} \mathbf{B}(\mathbf{x}; \boldsymbol{\psi}) L(d\boldsymbol{\psi}),$$

with respect to a Lévy random measure $L(d\boldsymbol{\psi}) = \sum_j \beta_j \delta_{\boldsymbol{\psi}_j}(d\boldsymbol{\psi})$ with a Lévy measure satisfying $M \equiv \nu(\mathbb{R} \times \Omega') < \infty$.

We follow the priors for $\boldsymbol{\beta}, \boldsymbol{\xi}$, J, M and σ of chapter 2 and have to place priors additionally on parameters in the basis functions including $\mathbf{c}, \boldsymbol{\nu}$, and $\mathbf{K}$. The prior distributions for $\mathbf{c}^{(j)}, \boldsymbol{\nu}^{(j)}$ and $\mathbf{K}_j$, following Nott et al. (2005) are assumed to follow the discrete uniform distribution over some predetermined sets. We also assume independent prior distributions for K_j, $\boldsymbol{\nu}^{(j)}$, and $\mathbf{c}^{(j)}$ In detail, the prior on K_j is uniform on $\{1, \ldots, K_{\max}\}$, where $K_{\max}$ is the maximum degree of interaction for the tensor product basis. We set $K_{\max}$ below 3 in most experiments of section 3.3 and section 3.4. The prior for $\boldsymbol{\nu}^{(j)}$ is uniform distribution that puts equal weight on indices of candidate predictors from one to $\binom{p}{K_j}$ denoted by V (e.g, if $K_j = 1$, then V $= \{1, 2, \ldots, p\}$). The prior for $\mathbf{c}_l^{(j)}$ is uniform on S, the prespecified subset of degree numbers of B-spline basis. Note that the prior for $\xi_l^{(j)}$ is the uniform distribution over $(\mathcal{X}_{\nu_l^{(j)}})^{(c_l^{(j)}+2)}$ since length and support of a knot sequence $\xi_l^{(j)}$ depend on a degree number $\mathbf{c}_l^{(j)}$ and an index $\boldsymbol{\nu}_l^{(j)}$, respectively. Below we summarize the MLABS model:

$$
\begin{gathered}
Y_i|\mathbf{x}_i \overset{ind}{\sim} \mathcal{N}(f(\mathbf{x}_i), \sigma^2), \quad i = 1, 2, \cdots, n, \\
f(\mathbf{x}_i) = \beta_0 + \sum_{j=1}^{J} \mathbf{B}_j(\mathbf{x}_i; K_j, \boldsymbol{\omega}_j)\beta_j, \\
\sigma^2 \sim \mathrm{IG}\left(\frac{r}{2}, \frac{rR}{2}\right), \\
J \sim \mathrm{Poi}(M), \quad M \sim \mathrm{Ga}(a_\gamma, b_\gamma), \\
\beta_j \overset{iid}{\sim} \mathcal{N}(0, \phi^2), \quad j = 1, 2, \cdots, J, \\
K_j \overset{iid}{\sim} \mathcal{DU}(\{1, \ldots, K_{\max}\}), \quad j = 1, 2, \cdots J, \\
\nu^{(j)} \overset{ind}{\sim} \mathcal{DU}(V), \quad j = 1, 2, \cdots, J, \\
c_l^{(j)} \overset{iid}{\sim} \mathcal{DU}(S), \quad l = 1, \cdots, K_j, \; j = 1, 2, \cdots, J, \\
\xi_l^{(j)} \overset{ind}{\sim} \mathcal{U}\left((\mathcal{X}_{\nu_l^{(j)}})^{(c_l^{(j)}+2)}\right), \quad l = 1, \cdots, K_j, \; j = 1, 2, \cdots, J,
\end{gathered}
\tag{3.3}
$$

and we set $\beta_0 = \overline{Y}$ and $\phi = \mathrm{Var}(\mathbf{Y})$ or $0.5 \times (\max_i\{Y_i\} - \min_i\{Y_i\})$.

3.2.2 Comparisons between basis fucntions of MLABS and MARS

The main difference between the basis function of the MLABS model and the (Bayesian) MARS model is the form of univariate basis functions in each element of the tensor product. Thus, their basis functions have very different parameters, too. The tensor product spline basis of the MARS is given by

$$N_j(\mathbf{x}_i) = \prod_{l=1}^{K_j} [s_l^{(j)} \cdot (x_{i,\nu_l^{(j)}} - t_l^{(j)})]_+,$$

where $s_l^{(j)} \in \{-1, +1\}$ is a sign indicator, $t_l^{(j)}$ is a knot point and $[\cdot]_+ = \max(\cdot, 0)$. K_j and $\nu_l^{(j)}$ of the MARS are the same as those of the MLABS.

First, the number and the location of the knot point in the basis functions are quite unlike. The B-spline basis with a degree $c_l^{(j)}$ in the MLABS needs $(c_l^{(j)}+2)$ knot points. The locations of the knots in the MLABS are freely chosen in the domain of $x_{\nu_l^{(j)}}$. In contrast, the univariate basis function of the MARS has only one knot point is set at each data point. In the Bayesian MARS, the prior distribution for $t_l^{(j)}$ is uniform on $\{x_{1,\nu_l^{(j)}}, \dots, x_{n,\nu_l^{(j)}}\}$. We fit the MLABS model and the MARS model to the data generated from a piecewise smooth function with two-dimensional support provided by Imaizumi and Fukumizu (2019) at 50×50 equally spaced points on the unit square. Figure 3.1 reveals that there is a considerable difference between the numbers of knot points used in the two methods and they set knot points with or without data points.

Second, while the degrees of the basis functions in the MARS model is fixed, those in the MLABS model are random and comprised of various combinations of predetermined degree numbers, S. Furthermore, the degree, α is added to the basis functions in the modified Bayesian MARS approach of Francom et al. (2018). Then, in the case of the (Bayesian) MARS, $\alpha = 1$. Figure 3.2 shows that the MLABS model needs more basis functions and uses more diverse types of basis functions than the MARS model to estimate an

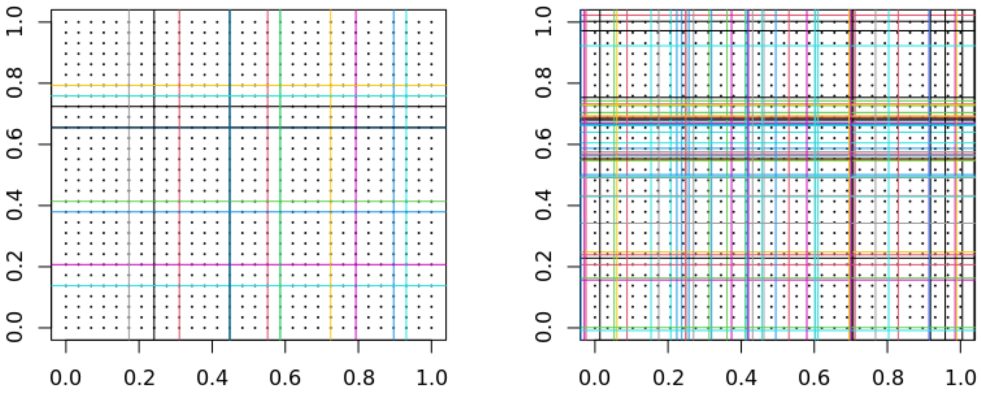


Figure 3.1: Plot for knot points of the Bayesian MARS (left) and the MLABS (right). In each plot the solid lines mean the locations of the knots and the small dots indicate the data points.

unknown surface. Especially, some of the tensor product bases in the MLABS model have very small local support, unlike those of the MARS. These parts will lead to producing accurate estimations for spatially varying surfaces.

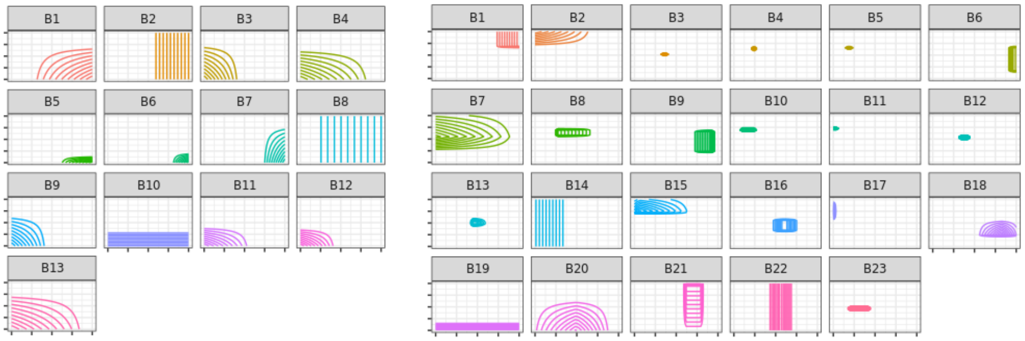


Figure 3.2: Plot for tensor product basis functions constructed by the Bayesian MARS (left) and the MLABS (right) to estimate a non-smooth function of Imaizumi and Fukumizu (2019).

3.2.3 Posterior inference

The structure of the MLABS model is similar to that of the LABS model, although we modified the form of basis function from univariate to the multivariate case. Thus, we follow most posterior computation steps in chapter 2 but incorporate update steps for newly added parameters such as $\mathbf{c}, \boldsymbol{\nu}$, and $\mathbf{K}$ to the existing MCMC algorithm. The joint posterior distribution of the MLABS model (3.3) is given by

$$\begin{aligned}\pi(\boldsymbol{\beta}, \boldsymbol{\xi}, \mathbf{K}, \boldsymbol{\nu}, \mathbf{c}, \boldsymbol{\xi}, J, M, \sigma^2 \,|\, \boldsymbol{Y}) \propto\, & L(\boldsymbol{Y} \,|\, f, \sigma^2) \cdot \pi(\boldsymbol{\beta}|J)\pi(\mathbf{K}|J) \cdot \pi(\boldsymbol{\nu}|\mathbf{K}, J) \\ & \times \pi(\mathbf{c}|\mathbf{K}, J) \cdot \pi(\boldsymbol{\xi}|\mathbf{K}, \boldsymbol{\nu}, \mathbf{c}, J) \\ & \times \pi(\boldsymbol{\xi}|\mathbf{K}, \boldsymbol{\nu}, \mathbf{c}, J) \cdot \pi(J \,|\, M) \cdot \pi(M) \cdot \pi(\sigma^2),\end{aligned}$$

where L is the likelihood function based on data generating mechanism (1.1).

We sum up the posterior sampling schemes of the MLABS model based on the RJMCMC algorithm. Let us denote $\theta_j := (\beta_j, K_j, \boldsymbol{\nu}^{(j)}, \mathbf{c}^{(j)}, \boldsymbol{\xi}^{(j)})$ by an element of $\boldsymbol{\theta} = \{\theta_1, \theta_2, \ldots, \theta_J\}$, where both $\boldsymbol{\nu}^{(j)}$ and $\mathbf{c}^{(j)}$ are K_j dimensional vectors, $\xi_l^{(j)}$ has $(c_l^{(j)} + 2)$ knot points and J is the number of coefficients (or basis functions) in the current model. The RJMCMC algorithm consists of three updating steps to sample posterior distribution. Such move types are called birth step, death step, and relocation step, respectively. The probabilities of exploring the birth, death, and relocation steps are p_b, p_d and p_w with $p_b + p_d + p_w = 1$. Each step is determined with probabilities p_b, p_d and p_w.

The birth step is to decide whether to add a new component θ_{J+1} generated from the proposal distributions or not, i.e., this updating phase is to allow the sampler to move from a current state $\boldsymbol{\theta}$ to a new state $\boldsymbol{\theta}^* := (\theta_1, \ldots, \theta_J, \theta_{(J+1)})$.

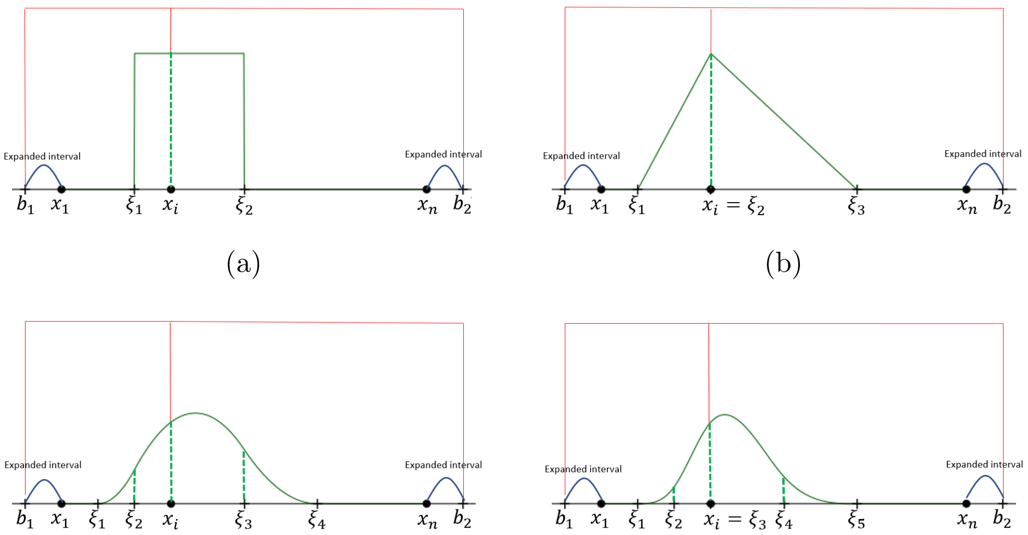
On the contrary, the death step is to decide whether to remove one of the existing components, θ_j or not. Finally, the relocation step is to only update $\boldsymbol{\theta}$ without altering the dimensionality of the parameters. The updating scheme of this step is the same as the standard MCMC methods, including Gibbs sampling or Metropolis-Hastings algorithm. The acceptance ratio in each move step is given by

$$A = \min\left[1, \frac{L(\mathbf{Y}|\boldsymbol{\theta}^*, J^*)\,\pi(\boldsymbol{\theta}^*|J^*)\pi(J^*)q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{L(\mathbf{Y}|\boldsymbol{\theta}\,, J)\,\pi(\boldsymbol{\theta}|J)\pi(J)q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}\right],$$

where $\boldsymbol{\theta}$ and J indicate the current model parameters and the number of tensor product basis functions in the current state. $\boldsymbol{\theta}^*$ and J^* refer to the new model parameters and the number of tensor product basis functions in the new state. $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$ is the jump proposal distribution that proposes a new state $\boldsymbol{\theta}^*$ given a current state $\boldsymbol{\theta}$. We follow the jump proposals of Lee et al. (2020) for each move step. The posterior samples for σ^2 and M are drawn from each full conditional distribution.

In practice, the LABS model had an inefficient sampling for knot points becuase they were uniformly sampled from the domain regardless of the distribution of data points. This caused proposed samples for knot points to locate far from the data points. As a result, the LABS model generated unnecessary B-spline bases and spent many MCMC iterations.

To solve this problem, we introduce new knot proposal schemes to the MLABS model. We illustrate the proposal processes for knot points using Figure 3.3. First, in the case of a degree $k = 0$ (panel (a) of Figure 3.3), a data point x_i is uniformly sampled from $\{x_1, \ldots, x_n\} := I$ and then knot points ξ_1 and ξ_2 are generated from $[b_1, x_i]$ and $[x_i, b_2]$ intervals, respectively. Here, the domain, $[x_1, x_n]$ is expanded to the interval $[b_1, b_2]$ for boundary data points. In practice, we expand by the $E \times (x_n - x_1) = (x_1 - b_1) = (b_2 - x_n)$ from endpoints, where E is a multiplier. Second, if $k = 1$ (panel (b) of Figure 3.3), x_i is uniformly sampled from I and set to ξ_2. Similarly, ξ_1 and ξ_3 are generated from $[b_1, x_i]$ and $[x_i, b_2]$ intervals, respectively. Third, in the case of $k = 2$ (panel (c) of Figure 3.3), ξ_1 and ξ_2 are generated from $[b_1, x_i]$ and ξ_3 and ξ_4 are generated from $[x_i, b_2]$ after x_i is uniformly sampled from I. Finally, for $k = 3$ (panel (d) of Figure 3.3), we generate a point ξ_1 uniformly distributed on I and set to ξ_3. Then, ξ_1 and ξ_2 are generated from $[b_1, x_i]$ and ξ_4 and ξ_5 are generated from $[x_i, b_2]$. These data-dependent knot proposals lead to achieving faster convergence than the LABS model.


Expanded interval
Expanded interval
b_1 x_1 ξ_1 x_i ξ_2 x_n b_2
Expanded interval
Expanded interval
b_1 x_1 ξ_1 $x_i = \xi_2$ ξ_3 x_n b_2
Expanded interval
Expanded interval
b_1 x_1 ξ_1 ξ_2 x_i ξ_3 ξ_4 x_n b_2
Expanded interval
Expanded interval
b_1 x_1 ξ_1 ξ_2 $x_i = \xi_3$ ξ_4 ξ_5 x_n b_2

(a)

(b)

(c)

(d)

Figure 3.3: Proposal schemes for knot points of the B-spline basis function with a degree $k =$ (a) 0, (b) 1, (c) 2, and (d) 3.

3.2.4 Binomial regressions for MLABS

The generalized linear models can cope with the non-Gaussian data. We can further extend the MLABS model (3.3) to generalized linear models by introducing a distribution and link function g into the model as

$$g(\mathbb{E}[\mathbf{Y} \,|\, \mathbf{x}]) := f(\mathbf{x}) = \sum_{j=1}^{J} \mathbf{B}_j(\mathbf{x}; K_j, \boldsymbol{\omega}_j)\beta_j. \tag{3.4}$$

In this subsection, we focus on binary regressions. Thus, the link function will be either the logit or probit function for Binomial distribution. For example, the logit model of the MLABS can be defined as

$$\begin{gathered}
Y_i \,|\, p_i \overset{ind}{\sim} \mathrm{Ber}(p_i), \quad Y_i \in \{0, 1\}, \\
p_i = \mathbb{P}(Y_i = 1|\mathbf{x}_i) = \mathrm{logit}^{-1}(f(\mathbf{x}_i)), \quad i = 1, 2, \cdots, n, \\
J \sim \mathrm{Poi}(M), \quad M \sim \mathrm{Ga}(a_\gamma, b_\gamma), \\
\beta_j \overset{iid}{\sim} \mathcal{N}(0, \tau^{-1}), \quad j = 1, 2, \cdots, J, \\
\tau \sim \mathrm{Ga}(a_\tau, b_\tau),
\end{gathered}$$

where $\mathrm{logit}^{-1}(a) = 1/(1 + \exp(-a))$. The priors for the remaining parameters $\mathbf{K}, \boldsymbol{\nu}, \mathbf{c}$, and $\boldsymbol{\xi}$ are identical with those of the MLABS model (3.3) for regression. For the logit model, the posterior distribution for $\boldsymbol{\beta}$ has no closed-form and is approximated using the Metropolis-Hastings sampler.

In probit link function, model (3.4) takes the form as

$$\mathbb{P}(Y_i = 1|\mathbf{x}_i) = \Phi(f(\mathbf{x}_i)),$$

where $\Phi(\cdot)$ denotes the cumulative distribution function of the standard normal distribution. For posterior inference in the probit model, we use the data augmentation algorithm proposed by Albert and Chib (1993). We introduce the latent variables z_i such that

$$z_i = f(\mathbf{x}_i) + \varepsilon_i, \quad \varepsilon_i \overset{iid}{\sim} \mathcal{N}(0,1),$$

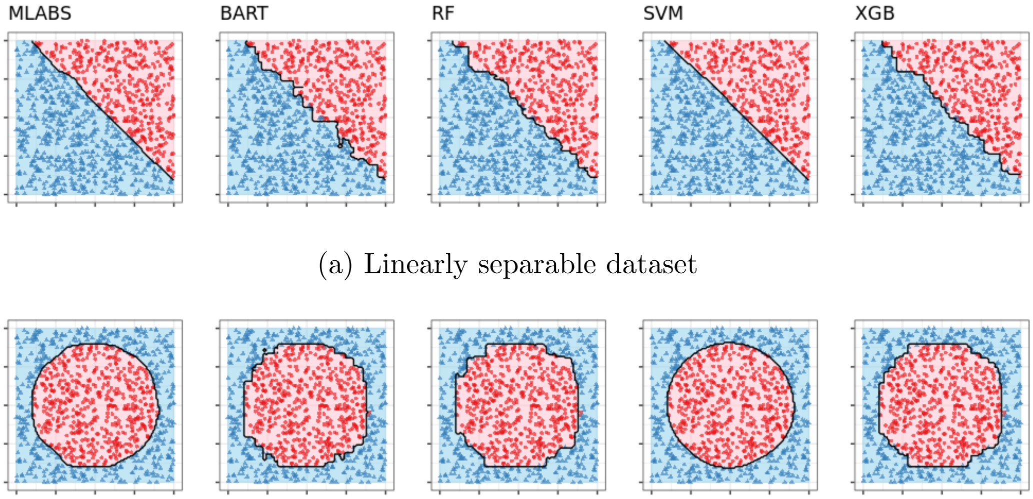
$$Y_i = \begin{cases} 1, & z_i > 0 \\ 0, & z_i \le 0 \end{cases}. \quad i = 1, \ldots, n.$$

Then, the normal prior for $\boldsymbol{\beta}$ gives a conjugate Gibbs-sampling update, unlike the logit model. The full conditional of z_i is given by

$$z_i \,|\, y_i, f(\mathbf{x_i}) = \begin{cases} \mathcal{TN}(f(\mathbf{x}_i), 1, 0, \infty), & \text{if } y_i = 1 \\ \mathcal{TN}(f(\mathbf{x}_i), 1, -\infty, 0), & \text{if } y_i = 0 \end{cases},$$

where $\mathcal{TN}(\mu, \sigma^2, a, b)$ is a truncated normal distribution with mean μ, variance σ^2, and support $[a, b]$. The posterior samples for $z_i, i = 1, \ldots, n$ are drawn from the full conditional after the RJMCMC algorithm as illustrated in subsection 3.2.3. The model parameters $M, J, \mathbf{K}, \boldsymbol{\nu}, \mathbf{c}$, and τ have the same prior distributions of the MLABS model (3.3). We use the MCMC algorithm using the probit link function in terms of the efficient posterior sampling.

We identify the decision boundaries for the probit model of the MLABS on five benchmark datasets: Linearly separable data, Circle data, XOR data, Two moons data, and Two spirals data. Both Figure 3.4 and Figure 3.5 show that the MLABS model produces visually more reasonable decision boundaries than the state-of-the-art classifiers. In other words, the MLABS model can have different and flexible decisions changing the degrees, or interaction orders in the tensor product basis function (3.1).



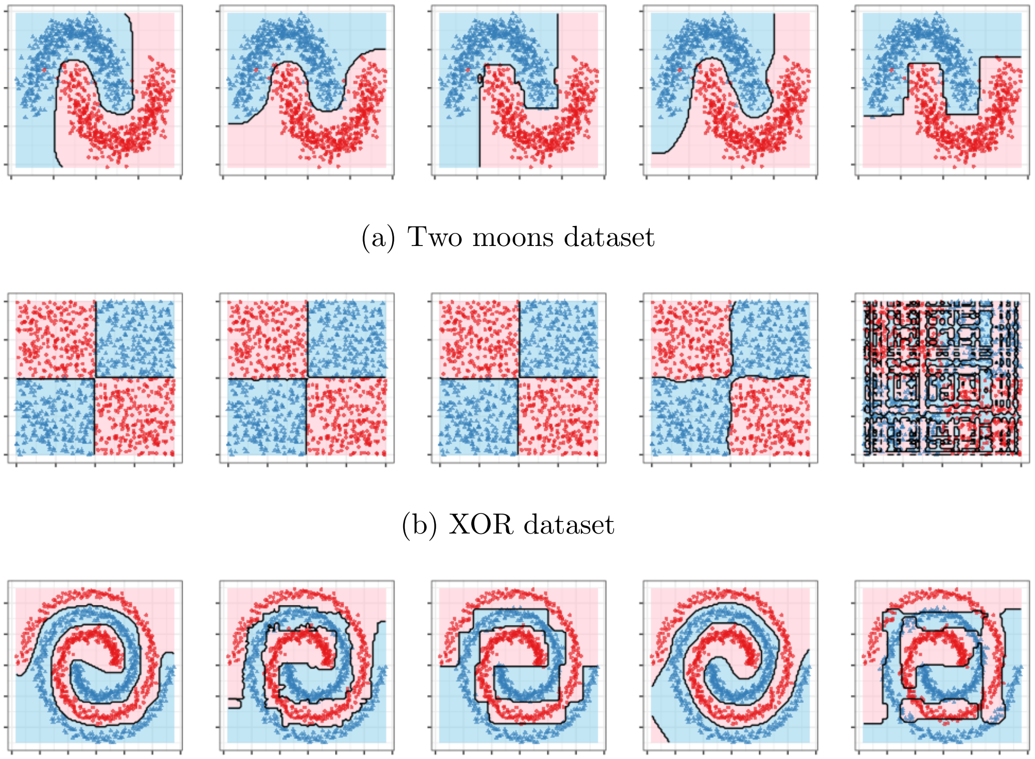
(a) Linearly separable dataset

(b) Circle dataset

Figure 3.4: Comparison of decision boundaries of MLABS and four classifiers on linearly separable and circle data sets

3.3 Simulation studies

In this section, in the regression settings, we measure the performance of the MLABS model (3.3) and competitive methods on simulated data sets. We first consider three test functions with bivariate predictors: the radial and complex interaction functions of Hwang et al. (1994) and the non-smooth test function of Imaizumi and Fukumizu (2019). The two test functions of Hwang et al. (1994) are smooth. Second, we take the examples proposed by Friedman (1991) as benchmark datasets in the multivariate nonparametric regression. One of Friedman's test functions is widely used to assess variable selection



(a) Two moons dataset

(b) XOR dataset

(c) Two Spirals dataset

Figure 3.5: Comparison of decision boundaries of MLABS and four classifiers on two moons, XOR, two spirals data sets

performance in high-dimensional data. For all test functions, we generate 100 pairs of held-in data with independent Gaussian noise and held-out data to evaluate the predictive performance based on root-mean-square error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}_i^\star) - \hat{f}(\mathbf{x}_i^\star))^2},$$

where $\mathbf{x}_i^\star$ is a held-out test set.

For comparison, we consider several competitive alternatives, including the multivariate adaptive regression splines of Friedman (1991) (denoted by MARS), a modified version of Bayesian MARS of Francom et al. (2018) (denoted by BASS), LARK model using multivariate Gaussian kernels of Ouyang (2008) (denoted by BARK), support vector machines with radial basis function (RBF) kernels of Boser et al. (1992); Cortes and Vapnik (1995) (denoted by SVM), a fully connected Neural network with two hidden layers (each 15 nodes) using sigmoid activation (denoted by NN), random forests of Breiman (2001) (denoted by RF), accelerated gradient-boosted decision trees of Chen and Guestrin (2016) (denoted by XGB), and Bayesian decision tree ensembles: Bayesian additive regression trees of Chipman et al. (2010) (denoted by BART) and BART using soft decision trees of Linero and Yang (2018) (denoted by SBART). All competing models were implemented in R packages: `earth`, `BASS` (Francom and Sansó, 2020), `bark`, `e1071` (Meyer and Wien, 2015), `keras`, `randomforest`, `xgboost`, `BayesTree`, and `SoftBart`, respectively.

The hyperparameters for all methods are chosen using grid-search with five-fold cross-validation. The MLABS model have seven tuning parameters such as a_γ, b_γ, r, R, S, $K_{\max}$, and E. We set $a_\gamma = 5$, $b_\gamma = 1$, $r = 0.01$ and $R = 0.01$ as default values. The parameters S, $K_{\max}$ and E are optimized by cross-validated grid-search over parameter grids. The hyperparameter candidates of all methods used in all experiments of this chapter are given in Table 3.1. We also run the MLABS model for 100,000 iterations, with the first 50,000 iterations discarded as burn-in, and retain every 50th sample.

Method	Parameter	Values considerred
MLABS	set of degree numbers: S	0, 1, 2, 3, (0,1), (0,2), (0,3), (1,2), (1,3), (2,3), (0,1,2), (0,1,2,3)
	mmaximum degree of interaction: $K_{\max}$	1, 2, 3
	multiplier for expanded intervals: E	0.1, 1, 2, 3
SBART	number of trees	20, 50, 200
BART	Sigma prior: (ν, q) combinations	(3,0.9), (3,0.99), (10,0.75)
	number of trees: m	50, 20
	μ prior: k value for σ_u	1, 2, 3, 5
BARK	type of prior for the scale parameters	"e", "d","se", "sd"
BASS	degree of splines: α	1, 2, 3
	maximum degree of interaction: $K_{\max}$	1, 2, 3
MARS	maximum number of terms in the pruned model	2, 12, 23, 34, 45, 56, 67, 78, 89, 100
	maximum degree of interaction: $K_{\max}$	1, 2, 3
RF	number of trees	2,. . . , p
SVM	regularization constant: C	0.001, 0.01, 0.1, 1, 5, 10, 100
	kernel hyperparameter: γ	0.5, 1, 2, 3, 4
NN	learning rate: r	0.001, 0.005, 0.01, 0.05, 0.1, 0.5
XGB	max number of boosting iterations	250, 500, 1000
	maximum depth of a tree	4, 8, 12
	learning rate	0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40
	minimum sum of instance weight needed in a child	1, 10, 15
	subsample ratio of columns	0.7, 1

Table 3.1: Hyperparameter grid for various competitive models

3.3.1 Surface examples

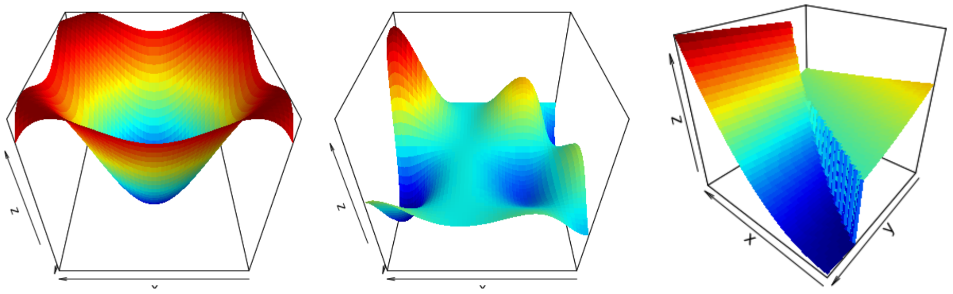
For each surface test function, in-sample data sets are generated from the true function at 30×30 equally spaced grid points on $\mathcal{X} := [0, 1] \times [0, 1]$. We also add independent normally distributed noises $\mathcal{N}(0, \sigma^2)$ to the true target functions. We select the value of σ such that the root signal-to-noise ratio (RSNR) was 1 and 5. We use 2500 additional data points generated independently and uniformly on $[0, 1]$ as out-of-sample data. The three true surfaces are given by

$$f^{(1)}(\mathbf{x}) = 24.234[r^2(0.75 - r^2)],$$

$$f^{(2)}(\mathbf{x}) = 1.9\{1.35 + e^{x_1} \sin[13(x_1 - 0.6)^2] \times e^{-x_2} \sin(7x_2)\},$$

$$f^{(3)}(\mathbf{x}) = \mathbf{1}_{R_1}(0.2 + x_1^2 + 0.1x_2) + \mathbf{1}_{R_1}(0.7 + 0.01|4x_1 + 10x_2 - 9|^{1.5}),$$

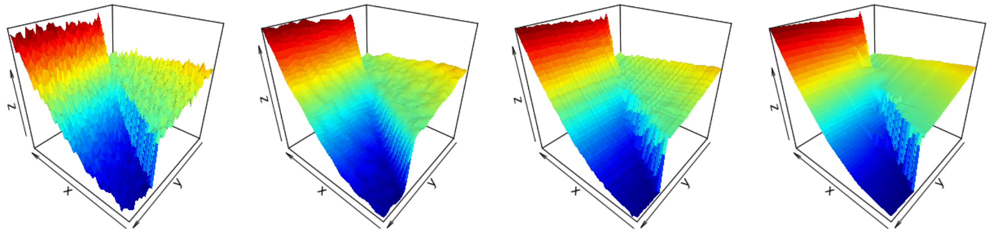
where $r^2 = (x_1 - 0.5)^2 + (x_2 - 0.5)^2$, $R_1 = \{(x_1, x_2) : x_2 \geq -0.6x_1 + 0.75\}$, $R_2 = I^2 \backslash R_1$ and $\mathbf{1}_R$ is the indicator function of R. They are visualized in Figure 3.6.



(a) (b) (c)

Figure 3.6: Three true surfaces: (a) radial (b) complex interaction and (c) non-smooth functions

In this example, we add the thin plate spline (TPS) as a benchmark technique since it is a commonly used tool for the smooth interpolation of two-dimensional data. The TPS is also referred to as a generalization of the smoothing spline. Results of this simulation are presented in Table 3.2. Table 3.2 demonstrates that the MLABS model performs well in most cases with the lowest, the second or the third-lowest average RMSE values across 100 in-sample and out-of-sample sets. According to the average rank of Table 3.2, the MLABS attains a more accurate estimation of the surface test function than the TPS. The tree-based models such as SBART, BART, RF, and XGB have difficulties in estimating smooth surfaces or regions due to their lack of smoothness. The NN doesn't work very well owing to fixed model structures relative to the training data size. The BASS can choose diverse degrees of the spline functions and produce the lowest value on the radial and complex test functions with RSNR = 1, unlike the MARS. One characteristic of the proposed model is smoothness adaptation Figure 3.7 supports that the MLABS model has the advantages of canceling the noise and adapting to the non-smooth function.



(a) (b) (c) (d)

Figure 3.7: Plot of the (a) true non-smooth function with additive Gaussian noise, and estimated surfaces obtained by fitting the (b) TPS, (c) BART, and (d) MLABS model.

3.3.2 Friedman's examples

We conduct additional experiments using Friedman 1, 2, and 3 data sets to assess the practical performance of the proposed method on general p (> 2) dimensional data. The Friedman 1 data set has ten independent uniform random variables on the interval $[0, 1]$. The output is computed using the following formula

$$f_1(\mathbf{x}) = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5.$$

The data set uses only the first five variables out of ten variables. The Friedman 2 and 3 data sets have four independent random variables with uniform distribution on the intervals

$$0 \leq x_1 \leq 100, \quad 40\pi \leq x_2 \leq 560\pi, \quad 0 \leq x_3 \leq 1, \quad 1 \leq x_4 \leq 11.$$

The corresponding responses are created according to the mean functions

$$\begin{aligned} f_2(\mathbf{x}) &= (x_1^2 + (x_2 x_3 - (1/(x_2 x_4)))^2)^{0.5}, \\ f_3(\mathbf{x}) &= \arctan((x_2 x_3 - (1/(x_2 x_4)))/x_1). \end{aligned}$$

These data sets have non-linear and high interaction order terms. For each test function, we create in-sample data sets of 250 observations and add independent Gaussian noise with mean zero and standard deviation σ, so that the root signal-to-noise ratio is set at 1 and 5. We also generate out-of-sample data sets of 1000 observations to measure the predictive accuracy of regression models.

Results of the simulation for Friedman's data sets are given in Table 3.3. The MLABS model has the best performance in almost all cases, as shown in Table 3.3. The feature of this experiment is the tensor product basis-based models, including the MLABS, BASS, and MARS, are superior to others. The results are caused by whether the interaction order terms can be estimated directly or not. Although the SBART and BART have relatively good prediction abilities, the MLABS overwhelms them for all test functions regardless of the RSNR. The average rank in Table 3.3 shows the ensemble models of the RF and XGB, and kernel-based models of the BARK and SVM perform poorly in Friedman's data sets. The NN is not appropriate for handling small datasets, as seen in the previous surface examples.

We evaluate the out-of-sample performance with methods based on the

Function	Noise	MLABS	SBART	BART	BARK	BASS	RF	SVM	MARS	NN	XGB	TPS
Radial	RSNR = 5	0.035 (2)	0.045 (4)	0.071 (7)	**0.03 (1)**	0.053 (6)	0.129 (10)	0.04 (3)	0.1 (8)	0.167 (11)	0.1 (9)	0.047 (5)
	RSNR = 1	0.124 (2)	0.187 (6)	0.216 (8)	0.154 (5)	**0.115 (1)**	0.531 (11)	0.152 (4)	0.203 (7)	0.366 (10)	0.247 (9)	0.135 (3)
Complex	RSNR = 5	**0.055 (1)**	0.067 (5)	0.114 (7)	0.074 (6)	0.059 (4)	0.149 (9)	0.057 (3)	0.34 (11)	0.316 (10)	0.13 (8)	0.056 (2)
	RSNR = 1	0.209 (3)	0.274 (6)	0.325 (8)	0.272 (5)	**0.195 (1)**	0.534 (11)	0.24 (4)	0.386 (9)	0.522 (10)	0.3 (7)	0.196 (2)
Non-smooth	RSNR = 5	**0.029 (1)**	0.036 (5)	0.038 (6)	0.047 (10)	0.04 (8)	0.039 (7)	0.036 (4)	0.058 (11)	0.033 (3)	0.043 (9)	0.032 (2)
	RSNR = 1	**0.059 (1)**	0.063 (2)	0.068 (5)	0.069 (7)	0.066 (3)	0.125 (11)	0.067 (4)	0.072 (9)	0.075 (10)	0.068 (6)	0.07 (8)
Average rank		**1.67 (1)**	4.67 (5)	6.83 (7)	5.67 (6)	3.83 (4)	9.83 (11)	3.67 (2)	9.17 (10)	9 (9)	8 (8)	3.67 (2)

Table 3.2: Average of predictive RMSEs over 100 pairs of held-in and held-out sets for three surface test functions. The rank of the method among the eleven approaches is shown in parentheses. The top-ranked model for each test function is given in bold.

Function	Noise	MLABS	SBART	BART	BASS	BARK	RF	SVM	MARS	NN	XGB
Friedman 1	RSNR = 5	**0.383 (1)**	0.532 (3)	1.077 (6)	0.394 (2)	1.042 (5)	2.173 (8)	3.977 (9)	0.609 (4)	4.075 (10)	1.435 (7)
	RSNR = 1	**1.796 (1)**	1.92 (2)	2.156 (4)	2.117 (3)	2.362 (5)	2.56 (8)	4.067 (9)	2.368 (6)	4.176 (10)	2.407 (7)
Friedman 2	RSNR = 5	**16.679 (1)**	27.731 (5)	44.088 (6)	16.994 (2)	26.553 (4)	50.754 (8)	78.98 (10)	24.106 (3)	53.783 (9)	48.897 (7)
	RSNR = 1	69.585 (2)	90.785 (4)	121.777 (5)	**53.52 (1)**	80.731 (3)	148.693 (8)	183.48 (10)	123.786 (6)	178.673 (9)	128.736 (7)
Friedman 3	RSNR = 5	**0.061 (1)**	0.063 (2)	0.079 (5)	0.066 (3)	0.092 (7)	0.105 (8)	0.125 (10)	0.073 (4)	0.088 (6)	0.105 (9)
	RSNR = 1	**0.11 (1)**	0.116 (2)	0.134 (4)	0.133 (3)	0.146 (7)	0.144 (6)	0.196 (9)	0.146 (8)	0.198 (10)	0.138 (5)
Average rank		**1.17 (1)**	3 (3)	5 (4)	2.33 (2)	5.17 (6)	7.67 (8)	9.5 (10)	5.17 (6)	9 (9)	7 (7)

Table 3.3: Average of predictive RMSEs over 100 pairs of held-in and held-out sets for Friedman's test functions. The rank of the method among the ten approaches is shown in parentheses. The top-ranked model for each test function is given in bold.

Friedman 1 data set in the high-dimensional settings for a detailed comparison. In other words, we check how well the models work as the number of variables increases. We reproduce the simulation scenarios of Linero and Yang (2018). We create five pairs of 250 training and 1000 test samples with p features, which increase from 5 to 1000 along an evenly spaced grid on the scale of $\log(p)$. Independent Gaussian noise with mean zero and standard deviation $\sigma^2 \in \{1, 10\}$ is also added to the training samples generated from the true mean function. Methods are compared by an average of RMSEs over five replications. Every time the number of variables increases, most methods are tuned by using cross-validation.

Results of this simulation are provided in Figure 3.8. An interesting part of Figure 3.8 is that the MLABS achieves the best performance up to about 70-dimensional data irrespective of the noise level. After that point, its error increases gradually in both the low and the high noise settings. Since the MLABS and BASS have the same performance behaviors, unlike MARS, these results seem to come from slowly mixing of the RJMCMC algorithm. In contrast, the SBART and MARS are interestingly invariant to the number of predictors. The SBART is superior to other methods, including the MLABS, for high-dimensional settings where p is large.

3.4 Real data applications

We now compare the MLABS model (3.3) with various competing methods on several real data sets of regression and classification problems.

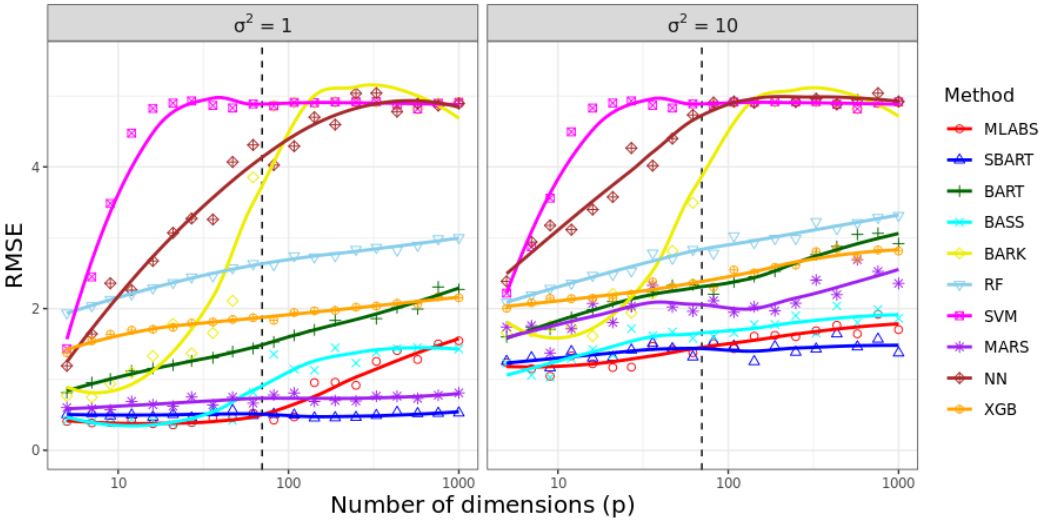


Figure 3.8: Average root-mean-square error of various methods with a smoothing line as a function of the dimension p on the log scale.

3.4.1 Regression examples

We prepare the six real-life datasets from the UCI Machine Learning Repository (UCI) and several R packages: `caret`, `mfp`, `MASS`, and `AppliedPredictiveModeling`. The summary of these data sets is provided in Table 3.6. Since the MLABS model can handle only quantitative variables, we don't consider categorical predictors in the data sets. We also erase missing values. Specifically, case 42 of the bodyfat data seems to be an apparent error, and its height variable is replaced by 69.5. The tecator meat and residential building datasets have multiple responses variables. We choose one of the responses in each data set: the percentages of protein (tecator meat) and actual sales prices (residential building).

We consider the nine competing approaches as illustrated in subsection 3.3.2 and select the best hyperparameters of each method using cross-validation

Dataset	# Samples	# Features	Source
Bodyfat	252	13	`mfp`
Boston housing	506	12	`MASS`
Concrete compressive strength	1030	8	UCI
Residential building	372	103	UCI
Tecator meat	215	100	`caret`
Chemical manufacturing process	152	58	`AppliedPredictiveModeling`

Table 3.4: Information of six data sets for regression analysis.

methods. To gauge the predictive performance among the methods, we make use of 20 times replicated five-fold cross-validations. Thus, we compute an average of 20 estimated CV errors as a measure of accuracy.

Results of the experiment for the regression problem are presented in Table 3.6. Table 3.6 illustrates that the MLABS model has stable predictive abilities by getting the best performance on three data sets. It also produces the third-lowest average RMSE in the remaining three data sets. By the average rank of Table 3.6, the MLABS model generally outperforms state-of-art methods in the fields of machine learning or Bayesian nonparametrics. Furthermore, for the tecator meat data, the tensor product basis based models work much better than the tree-based models do.

In contrast, the tree-based methods perform well for the chemical manufacturing process datasets and rank high among the methods. In practice, the kernel-based methods show bad performance in the regression examples, and the lowest-ranked approach is the SVM. These results are attributed to lacking the flexibility and adaptability to the data sets by using only one type of kernel function.

3.4.2 Classification examples

We choose the seven competitive methods for classification problems and exclude the SBART and BASS because the two models cannot yet analyze the binary data. We compare the MLABS model using the probit link with other methods that optimized their hyperparameters using grid-search with five-fold cross-validation by a classification performance measure: AUC (area under the receiver operating characteristic (ROC) curve). The AUC is the most common metric for classification tasks, and the value lies between 0 to 1, where 1 indicates an excellent classifier. We calculate the average of performance metrics obtained by repeating 5-fold cross-validation 20 times. We collect the seven real data sets for classification from the UCI Machine Learning Repository and two R packages: `mlbench` and `datamicroarray`. The Alon dataset is the high-dimensional microarray data set for colon cancer. The Pima Indian diabetes data set contains zero values of some variables, and we consider the values missing values. The missing values and categorical variables of every real data set for classification are processed in the same way as regression experiments. The real data sets are listed in with the information such as the number of sample size and features, source, and imbalanced ratio (IR) defined as

$$\text{IR} = \frac{\max_{C \in \mathcal{A}} |C|}{\min_{C \in \mathcal{A}} |C|},$$

where $\mathcal{A}$ is the set of all classes.

Results of this experiment are given in Table 3.7. The columns of the methods represent their average of cross-validated AUC values over 20 replicates. As shown in Table 3.7, the MLABS method doesn't show excellent predictive

Dataset	# Samples	# Features	IR	Source
Parkinson	195	22	3.06	UCI
Ionosphere	351	32	1.79	UCI
Breast cancer Wisconsin (Diagnostic)	569	30	1.7	UCI
Sonar	208	61	1.14	UCI
Spambase	4601	57	1.54	UCI
Pima Indian diabetes	392	9	2.02	`mlbench`
Alon	62	2000	1.82	`datamicroarray`

Table 3.5: Information of seven real data sets for classification tasks

performance for classification, but it is comparable to the XGB and RF as gold standard models. Specifically, the MLABS model performs well in most cases except the Ionosphere, Sonar, and Alon data set. It is seen as having difficulties estimating in high-dimensional cases. Here, the XGB model provides the best performance, followed by the RF, MLABS, and BART model. In contrast with the regression problems, tree-based models generally provide better predictive capabilities than the others.

3.5 Discussion

In this chapter, we have introduced a general Bayesian sum-of-bases model named Multivariate Lévy Adaptive B-Spline Regression using the tensor product of B-spline basis function of which parameters are automatically determined by the Lévy random measure. The B-spline basis has nice properties such as local support and differentiability. We have illustrated that it has a powerful predictive ability over the state-of-the-art methods in simulation studies and real data applications of the regression problems. We also proposed a comparable classification model using the data augmentation strategies of Albert and Chib (1993). However, there are drawbacks that the proposed model can treat only continuous variables and is slightly inefficient as it uses the RJMCMC. The MCMC algorithm makes it difficult to deal with high-dimensional data. The classifier based on the MLABS framework also does not work well compared to the tree-based models. Further studies are needed to improve these problems.

Data	MLABS	SBART	BART	BARK	BASS	RF	SVM	MARS	NN	XGB
Bodyfat	**4.08 (1)**	4.1 (2)	4.22 (6)	4.17 (4)	4.13 (3)	4.32 (8)	8.08 (10)	4.36 (9)	4.18 (5)	4.28 (7)
Boston housing	**2.95 (1)**	3.11 (3)	3.14 (4)	3.31 (6)	4.25 (10)	3.19 (5)	3.75 (8)	3.84 (9)	3.71 (7)	3 (2)
Concrete compressive strength	4.3 (3)	4.82 (4)	4.14 (2)	6.66 (9)	5.44 (6)	4.89 (5)	5.9 (7)	6.31 (8)	8.1 (10)	**3.85 (1)**
Residential building	**108.67 (1)**	115.29 (3)	128.93 (5)	140.98 (7)	137.65 (6)	245.83 (9)	909.32 (10)	122.21 (4)	113.7 (2)	196.93 (8)
Tecator meat	1.17 (3)	1.53 (5)	2.18 (9.5)	2.18 (9.5)	1 (2)	2.04 (8)	1.86 (6)	**0.98 (1)**	1.42 (4)	1.93 (7)
Chemical manufacturing process	1.09 (3)	1.11 (5)	1.09 (4)	1.21 (6)	1.24 (7)	1.08 (2)	1.87 (10)	1.26 (8)	1.42 (9)	**1.01 (1)**
Average rank	**2 (1)**	3.67 (2)	5.08 (4)	6.92 (9)	5.67 (5)	6.17 (6.5)	8.5 (10)	6.5 (8)	6.17 (6.5)	4.33 (3)

Table 3.6: Average root-mean-square error of the MLABS and competitve methods with the rank of the method among the ten approaches in parentheses in the real data sets for regression problem. The top-ranked model for each real data set is given in bold.

Data	MLABS	BART	BARK	RF	SVM	MARS	NN	XGB
Parkinson	0.97 (2)	0.962 (4)	0.922 (6)	0.961 (5)	**0.975 (1)**	0.899 (7)	0.797 (8)	0.967 (3)
Ionosphere	0.971 (4)	0.963 (5)	0.95 (6)	**0.978 (1)**	0.978 (2)	0.935 (7)	0.917 (8)	0.976 (3)
Breast cancer Wisconsin	**0.995 (1)**	0.992 (3)	0.984 (7)	0.989 (4)	0.975 (8)	0.988 (5)	0.987 (6)	0.994 (2)
Sonar	0.907 (5)	0.933 (3)	0.79 (8)	**0.941 (1)**	0.909 (4)	0.864 (6)	0.853 (7)	0.935 (2)
Pima Indian Diabetes	0.849 (2)	0.847 (4)	0.846 (5)	0.848 (3)	0.801 (8)	0.816 (7)	0.831 (6)	**0.852 (1)**
Spambase	0.983 (3)	0.982 (4)	0.975 (6)	0.986 (2)	0.949 (8)	0.977 (5)	0.966 (7)	**0.988 (1)**
Alon	0.885 (4)	0.889 (3)	0.836 (6)	0.876 (5)	0.5 (8)	0.771 (7)	0.905 (2)	**0.914 (1)**
Average rank	3.125 (3)	3.5 (4)	6.25 (7)	3 (2)	5.875 (5)	6.375 (8)	6.125 (6)	**1.75 (1)**

Table 3.7: Average of cross-validated AUC of the MLABS and competitve methods with the rank of the method among the eight approaches in parentheses in the real data sets for classfication problem. The top-ranked model for each real data set is given in bold.

Chapter 4

Concluding Remarks

This dissertation proposed two Bayesian nonparametric regression models constructed by a sum of the B-spline bases as elements of an overcomplete system. The first model, the LABS for function estimation, was designed to simultaneously use various B-spline basis functions to capture all parts of functions with locally varying smoothness. We presented that the function spaces for the LABS model rely on the degree of B-spline basis that belongs to the Besov spaces and proved that its prior has full support in the same Besov spaces.

We also proposed the MLABS model for multi-dimensional data analysis by using tensor products of B-spline bases. These basis functions enable it to circumvent the curse of dimensionality. We also considered the efficient knot proposal schemes, and then the algorithm converged faster than an existing proposal method. Finally, we showed that it achieved high predictive accuracy through simulated and real data sets in the regression problems.

Future work will develop a versatile and efficient sampling-based model for the MLABS model. One possibility is to give the Lévy process prior up and use regularization priors to handle the high-dimensional data under a large and fixed number of the basis function. Using a Bayesian backfitting algorithm of Hastie et al. (2000) as a core algorithm in the BART is expected to be more effective to achieve high performance and fast convergence than the inefficient RJMCMC. Moreover, scalable algorithms such as either the Consensus Monte Carlo or variational Bayes can be applied to our model for large and tall data. Another possibility is that the tensor product bases will be to allowed to contain indicators for categorical data.

Appendix A

Appendix

A.1 Appendix for Chapter 2

A.1.1 Proof of Theorem 2.3.1

For simplicity, we assume that $\mathcal{X} = [0,1]$. Since the B-spline basis has local support and is bounded, $\|B_k(x;\boldsymbol{\xi}_k)\|_p$ is finite for all $k \geq 0$. It is enough to show that if the Besov semi-norm, $|B_k(x;\boldsymbol{\xi}_k)|_{\mathbb{B}^{\alpha}_{p,q}}$ is finite for all $k \geq 0$.The definition of the modulus of smoothness and the property that $\omega_k(f,t)_p \leq 2^r \cdot \omega_{k-r}(f,t)_p$, $0 \leq r \leq k$, if $f \in L_p(\mathcal{X})$ imply that

$$\omega_r(B_k(x;\boldsymbol{\xi}_k),t)_p \leq 2^{r-1} \cdot \omega_1(B_k(x;\boldsymbol{\xi}_k),t)_p.$$

Let k be zero. Then, the B-spline basis is piecewise constant with 2 knots, $\boldsymbol{\xi}_0 := (\boldsymbol{\xi}_{01}, \boldsymbol{\xi}_{02})$. By the definition of the B-spline basis (2.6), we divide into two cases to calculate the modulus of continuity.

Case 1. *Assume* $\boldsymbol{\xi}_{01} + h < \boldsymbol{\xi}_{02},\ h > 0$. *Thus,*

$$\|B_0(x+h;\boldsymbol{\xi}_0) - B_0(x;\boldsymbol{\xi}_0)\|_p \leq 2 \cdot h^{\frac{1}{p}}.$$

Case 2. *Assume* $\boldsymbol{\xi}_{01} + h > \boldsymbol{\xi}_{02},\ h > 0$. *Thus,*

$$\|B_0(x+h;\boldsymbol{\xi}_0) - B_0(x;\boldsymbol{\xi}_0)\|_p \leq 2 \cdot h^{\frac{1}{p}}.$$

Therefore, in all cases,

$$\omega_r(B_0(x;\boldsymbol{\xi}_0), t)_p \leq 2^r \cdot h^{\frac{1}{p}}. \tag{A.1}$$

By definition, $|B_0(x;\boldsymbol{\xi}_0)|_{\mathbb{B}^{\alpha}_{p,q}} = \left(\int_0^\infty (t^{-s}\omega_r(B_0(x;\boldsymbol{\xi}_0),t)_p)^p \frac{dt}{t}\right)^{1/q}$, so

$$\begin{aligned} |B_0(x;\boldsymbol{\xi}_0)|_{\mathbb{B}^{\alpha}_{p,q}} &\leq \left[\int_0^1 t^{-sq-1} \cdot 2^{rq} \cdot t^{\frac{1}{p}}\, dt + \int_1^\infty t^{-sq-1} \cdot 2^{rq}\, dt\right]^{1/q} \\ &= 2^r \cdot \left[\int_0^1 t^{-q(s-\frac{1}{p})-1}\, dt + \int_1^\infty t^{-sq-1}\, dt\right]^{1/q}. \end{aligned}$$

The upper bound of $|B_0(x;\boldsymbol{\xi}_0)|_{\mathbb{B}^{\alpha}_{p,q}}$ is finite if and only if $\alpha < \frac{1}{p}$ and $q < \infty$.

Let $k \geq 1$. Since the B-spline basis of degree k is a piecewise polynomial and has $(k-1)$ continuous derivatives at the knots, it falls in $W^k_p(\mathcal{X})$, where $W^k_p(\mathcal{X})$ is the Sobolev space, which is a vector space of functions that have weak derivatives. See the definition of the Sobolev space described in chapter 2.5 of DeVore and Lorentz (1993). We use the following property of the modulus of smoothness,

$$\omega_{r+k}(f,t)_p \leq t^r \cdot \omega_k(f^{(r)},t)_p,\ t > 0,$$

where $f^{(r)}$ is the weak rth derivative of f. For $k \geq 1$, the Besov semi-norm of $B_k(x; \boldsymbol{\xi}_k)$ is bounded by

$$\begin{aligned}
|B_k(x;\boldsymbol{\xi}_k)|_{\mathbb{B}^\alpha_{p,q}} &= \left(\int_0^\infty (t^{-\alpha}\omega_r(B_k(x;\boldsymbol{\xi}_k),t)_p)^q \frac{dt}{t}\right)^{1/q} \\
&\leq \left(\int_0^1 (t^{-\alpha}\cdot(t^k\cdot\omega_{r-k}(B_k^{(k)}(x;\boldsymbol{\xi}_k),t)_p)^q \frac{dt}{t} + \int_1^\infty 2^{rq}\cdot t^{-sq-1}\,dt\right)^{1/q} \\
&\leq \left(\int_0^1 (t^{-\alpha q-1}\cdot(t^k\cdot 2^{r-k-1}\cdot\omega_1(B_k^{(k)}(x;\boldsymbol{\xi}_k),t)_p)^q\,dt + 2^{rq}\cdot\int_1^\infty t^{-\alpha q-1}\,dt\right)^{1/q}
\end{aligned} \tag{A.2}$$

Since $B_k^{(k)}(x;\boldsymbol{\xi}_k)$ is a piecewise constant function, (A.1) implies that

$$\omega_1(B_k^{(k)}(x;\boldsymbol{\xi}_k),t)_p \leq C\cdot h^{\frac{1}{p}}, \quad \text{for some constant } C > 0. \tag{A.3}$$

Using (A.2) and (A.3), it follows that

$$|B_k(x;\boldsymbol{\xi}_k)|_{\mathbb{B}^\alpha_{p,q}} \leq \left(C'\cdot\int_0^1 t^{-\alpha q+kq+\frac{q}{p}-1}\,dt + 2^{rq}\cdot\int_1^\infty t^{-\alpha q-1}\,dt\right)^{1/q}, \quad \text{for some constant } C' > 0.$$

For all $k \geq 1$, $|B_k(x;\boldsymbol{\xi}_k)|_{\mathbb{B}^\alpha_{p,q}}$ is finite if and only if $\alpha < k + \frac{1}{p}$ and $q < \infty$, so the proof is complete.

A.1.2 Proof of Theorem 2.3.2

By Theorem 3 of Wolpert et al. (2011), the L_p norm and Besov semi-norm of η satisfy the following upper bounds, respectively.

$$\|\eta\|_p \leq \sum_{k\in S}\sum_{l} \|B_k(x;\boldsymbol{\xi}_{k,l})\|_p|\beta_{k,l}|,$$
$$|\eta|_{\mathbb{B}^{\alpha}_{p,q}} \leq \sum_{k\in S}\sum_{l} |\beta_{k,l}|\cdot|B_k|_{\mathbb{B}^{\alpha}_{p,q}},$$

Since the condition for (2.11) is satisfied and B-spline basis is bounded and locally supported, $\|\eta\|_p$ is almost surely finite. To obtain finite Besov semi-norms for all $k \in S$, the smoothness parameter α should be $\alpha < \min(S) + \frac{1}{p}$ by Theorem 2.3.1. Therefore, η belongs to $\mathbb{B}^{\alpha}_{p,q}$ with $\alpha < \min(S) + \frac{1}{p}$ almost surely.

A.1.3 Proof of Theorem 2.3.3

For the sake of simplicity we assume $\mathcal{X} = [0,1]$. Fix $\delta > 0$ and $\eta_0 \in \mathbb{B}^{\alpha}_{p,q}([0,1])$ with $\alpha > 0, 1 \leq p,q < \infty$. If $1 \leq p' \leq p < \infty$, then η_0 also belongs to $\mathbb{B}^{\alpha}_{p',q}([0,1])$ by property of the Besov space (Cohen, 2003)[3.2, page 163]. From Theorem 2.1 of Petrushev (1988), we can show that there exists a spline $s \in S(n^\star, q)$ such that

$$\|\eta_0 - s\|_p < C\frac{\|\eta_0\|_{\mathbb{B}^{\alpha}_{p',q}}}{(n^\star)^\alpha} < \frac{\delta}{2},$$

where $S(n^\star, q)$ denotes the set of all splines of degree $(q-1)$ with a sufficiently large number $n^\star$ knots and constant $C = C(\alpha, p, q)$. Since any spline of given degree can be represented as a linear combination of B-spline basis functions with same degree, we can define a spline $s(x)$ by

$$s(x) = \sum_{j=1}^{n^\star} \beta_j^* B^*_{(q-1),j}(x), \tag{A.4}$$

where $B^*_{(q-1),j}(x)$ is the B-spline basis of degree $(q-1)$ with a sequence of knots $\boldsymbol{\xi}^*$ in (2.5).

Set $n^\star := \sum_{k\in S} J_k^\delta$, $A := \sum_{k\in S}\sum_{l=1}^{J_k^\delta} |\beta_{k,l}| < \infty$, $\rho := \sup \|B_k(x, \boldsymbol{\xi}_k)\|_p < \infty$ and $\epsilon := \frac{\delta}{2(A+\rho)}$. We denote the range of a sequence of knots $\boldsymbol{\xi}_{k,l}$ by $r(\boldsymbol{\xi}_{k,l})$, e.g., $r(\boldsymbol{\xi}_{k,l}) = (\xi_{k,l,(k+2)} - \xi_{k,l,1})$. For convenience, we reindex the coefficients and knots of the spline $s(x)$ in (A.4) such that $\beta^*_{k,l}$ and $\boldsymbol{\xi}^*_{k,l}$ for $l = 1, \ldots, J_k^\delta, k \in S$. Then, the spline $s(x)$ can be expressed as follows:

$$s(x) = \sum_{k\in S}\sum_{l=1}^{J_k^\delta} \beta^*_{k,l} B^*_{(q-1),l}(x; \boldsymbol{\xi}^*_{k,l}),$$

where $\boldsymbol{\xi}^*_{k,l} := (\xi^*_l, \ldots, \xi^*_{l+(q-1)+1})$ is a subsequence of given knots $\boldsymbol{\xi}^*$. Using the definitions of B-spline basis in (2.5) and (2.6), we can find a $\zeta > 0$ such that

$$\max(r(\boldsymbol{\xi}_{k,l}), r(\boldsymbol{\xi}^*_{k,l})) < \zeta \Rightarrow \|B_k(x; \boldsymbol{\xi}_{k,l}) - B^*_{(q-1),l}(x; \boldsymbol{\xi}^*_{k,l})\|_p < \epsilon, \quad \forall l, \forall k.$$

Let's define the set

$$\bar{b}'(\eta_0) := \left\{ \eta : \eta(x) = \sum_{k\in S}\sum_{l=1}^{J_k^\delta} \beta_{k,l} B_k(x; \boldsymbol{\xi}_{k,l}),\ \sum_{k\in S}\sum_{l=1}^{J_k^\delta} |\beta_{k,l} - \beta^*_{k,l}| < \epsilon,\ \max(r(\boldsymbol{\xi}_{k,l}), r(\boldsymbol{\xi}^*_{k,l})) < \zeta, \forall l, \forall k \right\}. \tag{A.5}$$

Lemma A.1.1.

$$\bar{b'}_\delta(\eta_0) \subset \bar{b}_\delta(\eta_0)$$

Proof. It suffices to show that $\eta \in \bar{b'}_\delta(\eta_0) \implies \eta \in \bar{b}_\delta(\eta_0)$ to finish the proof of the lemma. For any $\eta \in \bar{b'}_\delta(\eta_0)$,

$$\begin{aligned}\|\eta - s\|_p &\leq \sum_{k\in S}\sum_{l=1}^{J_k^\delta} \|\beta_{k,l}B_k(x;\boldsymbol{\xi}_{k,l}) - \beta_{k,l}^* B_{(q-1),l}^*(x;\boldsymbol{\xi}_{k,l}^*)\|_p \\ &\leq \sum_{k\in S}\sum_{l=1}^{J_k^\delta} |\beta_{k,l}| \cdot \|B_k(x;\boldsymbol{\xi}_{k,l}) - B_{(q-1),l}^*(x;\boldsymbol{\xi}_{k,l}^*)\|_p + \sum_{k\in S}\sum_{l=1}^{J_k^\delta} |\beta_{k,l} - \beta_{k,l}^*| \cdot \|B_k(x;\boldsymbol{\xi}_{k,l})\|_p \\ &\leq \epsilon \cdot \sum_{k\in S}\sum_{l=1}^{J_k^\delta} |\beta_{k,l}| + \rho \cdot \sum_{k\in S}\sum_{l=1}^{J_k^\delta} |\beta_{k,l} - \beta_{k,l}^*| \\ &\leq \epsilon \cdot A + 2\rho \cdot \epsilon = (A+\rho)\cdot\epsilon = \frac{\delta}{2}.\end{aligned}$$

By the triangle inequality,

$$\begin{aligned}\|\eta - \eta_0\|_p &\leq \|\eta - s\|_p + \|s - \eta_0\|_p \\ &< \frac{\delta}{2} + \frac{\delta}{2} = \delta.\end{aligned}$$

Thus, $\eta \in \bar{b}_\delta(\eta_0)$ and this finishes the proof of the lemma. □

To complete the proof of this theorem, we have to show that $\Pi\left(\eta \in \bar{b}'_{\delta}(\eta_0)\right) > 0$ by using the previous lemma. Let $J^\star := \max_{k\in S} J_k^\delta$. By the triangle inequality,

$$
\begin{aligned}
\Pi\left(\eta \in \bar{b}'_{\delta}(\eta_0)\right) &= \Pi\left(\sum_{k\in S}\int\int_{\mathbf{R}\times\mathcal{X}^{(k+2)}} \beta_k B_k(x;\boldsymbol{\xi}_k) N_k(d\beta_k, d\boldsymbol{\xi}_k) \in \bar{b}'_{\delta}(\eta_0)\right) \\
&= \Pi\left(\sum_{k\in S}\sum_{l=1}^{J_k} B_k(x;\boldsymbol{\xi}_{k,l})\beta_{k,l} \in \bar{b}'_{\delta}(\eta_0)\right) \\
&= \mathbb{P}\left[\sum_{k\in S}\sum_{l=1}^{J_k^\delta} |\beta_{k,l}-\beta^*_{k,l}| < \epsilon,\ \max(r(\boldsymbol{\xi}_{k,l}), r(\boldsymbol{\xi}^*_{k,l})) < \zeta,\ J_k = J_k^\delta,\ \forall k \in S\right] \\
&> \prod_{k\in S}\left\{\mathbb{P}\left[|\beta_{k,l}-\beta^*_{k,l}| < \frac{\epsilon}{|S|J^\star}, \max(r(\boldsymbol{\xi}_{k,l}), r(\boldsymbol{\xi}^*_{k,l})) < \zeta,\ l = 1, 2, \ldots, J_k^\delta\right]\right\} \\
&\quad \times \prod_{k\in S}\left[\frac{\nu_k(\mathbb{R}\times\mathcal{X}^{(k+2)})^{J_k^\delta}\cdot\exp(-\nu_k(\mathbb{R}\times\mathcal{X}^{(k+2)}))}{J_k^\delta!}\right] \\
&= \prod_{k\in S}\left\{\prod_{j=1}^{J_k^\delta}\left[\frac{\nu_k(|\beta_{k,l}-\beta^*_{k,l}| < \frac{\epsilon}{|S|J^\star}, \max(r(\boldsymbol{\xi}_{k,l}), r(\boldsymbol{\xi}^*_{k,l})) < \zeta)}{\nu_k(\mathbb{R}\times\mathcal{X}^{(k+2)})}\right]\right\} \\
&\quad \times \prod_{k\in S}\left[\frac{\nu_k(\mathbb{R}\times\mathcal{X}^{(k+2)})^{J_k^\delta}\cdot\exp(-\nu_k(\mathbb{R}\times\mathcal{X}^{(k+2)}))}{J_k^\delta!}\right] \\
&= \prod_{k\in S}\left\{\prod_{j=1}^{J_k^\delta}\left[\int_{|\beta_{k,l}-\beta^*_{k,l}|<\frac{\epsilon}{|S|J^\star}} \pi(\beta_k)d\beta_k \int_{\max(r(\boldsymbol{\xi}_{k,l}), r(\boldsymbol{\xi}^*_{k,l}))<\zeta} \pi(\boldsymbol{\xi}_k)d\boldsymbol{\xi}_k\right]\right. \\
&\quad \left. \times \left[\frac{M_k^{J_k^\delta}\cdot\exp(-M_k)}{J_k^\delta!}\right]\right\}.
\end{aligned}
$$

Since we assume a finite Levy measure and $\pi(\beta_k) = \mathcal{N}(\beta_k; 0, \phi_k^2)$, $\pi(\boldsymbol{\xi}_k) =$

$\mathcal{U}(\mathcal{X}^{(k+2)})$ in the LABS model,

$$\Pi\left(\eta \in \bar{b'}_{\delta}(\eta_0)\right) > 0.$$

Hence, by applying the lemma, we get $\Pi\left(\eta \in \bar{b_\delta}(\eta_0)\right) \geq \Pi\left(\eta \in \bar{b'_\delta}(\eta_0)\right) > 0$ and the theorem is proved.

A.1.4 Full simulation results for Simulation 1

This appendix contains the full simulations results of the four DJ test functions. We simulated two scenarios: (a) small sample size ($n = 128$) and (b) large sample size ($n = 512$) with different noise levels (RSNR = 3, 5, and 10).

Model	Bumps		
	RSNR=3	RSNR=5	RSNR=10
BSP-2	26.904(0.4461)	25.495(0.1606)	24.9(0.0401)
LOESS	47.266(0.1618)	47.163(0.0812)	47.119(0.0366)
SS	43.552(4.6764)	43.377(4.7159)	43.984(4.6074)
NWK	39.892(1.8831)	39.365(1.3862)	39.033(0.7393)
EBW	4.986(1.1761)	1.936(0.601)	0.447(0.0981)
TF-0	47.574(3.6625)	48.449(0.9754)	48.604(0.1229)
TF-1	47.836(1.5952)	47.906(0.5399)	47.954(0.5211)
TF-2	47.585(1.7116)	47.748(0.0384)	47.714(0.0096)
GSP-L	22.99(4.8847)	22.03(5.3556)	20.112(4.4213)
GSP-R	41.626(2.9041)	40.819(2.7234)	40.955(3.102)
BPP-10	4.571(2.3604)	3.668(2.7142)	3.304(2.9789)
BPP-21	15.115(5.9376)	14.674(6.3396)	14.363(6.7395)
BASS-1	2.968(0.4322)	1.206(0.4907)	0.252(0.0421)
BASS-2	47.977(7.4411)	45.988(9.3435)	45.021(9.2185)
LARMuK	2.852(0.426)	1.182(0.678)	0.319(0.0663)
LABS	**2.589(0.5908)**	**0.837(0.3124)**	**0.246(0.0683)**

Table A.1: Average mean squared error with estimated standard error in parentheses from 100 replications for Bumps example with $n = 128$

Model	Blocks		
	RSNR=3	RSNR=5	RSNR=10
BSP-2	5.96(0.4429)	4.53(0.1594)	3.927(0.0399)
LOESS	17.924(0.7218)	17.503(0.3846)	17.332(0.2312)
SS	4.895(0.5145)	3.396(0.241)	2.699(0.1107)
NWK	4.285(0.7336)	1.936(0.36)	0.472(0.0567)
EBW	3.243(1.0747)	0.859(0.2237)	0.21(0.0484)
TF-0	2.553(1.0393)	1.062(0.4313)	0.344(0.126)
TF-1	3.502(0.8151)	1.418(0.3327)	0.387(0.0869)
TF-2	3.862(0.7746)	1.715(0.298)	0.499(0.1615)
GSP-L	7.409(1.3261)	6.637(0.9807)	6.546(1.1115)
GSP-R	15.654(2.0255)	15.323(1.8902)	15.44(2.3391)
BPP-10	2.156(0.789)	0.908(0.2537)	0.465(0.2403)
BPP-21	3.918(0.589)	2.682(0.5399)	2.311(0.451)
BASS-1	2.498(0.6331)	0.696(0.2226)	0.122(0.0381)
BASS-2	7.533(1.7616)	4.253(0.8586)	2.852(0.3863)
LARMuK	1.799(0.5873)	0.682(0.2436)	0.193(0.08)
LABS	**1.305(0.5272)**	**0.365(0.1645)**	**0.072(0.0293)**

Table A.2: Average mean squared error with estimated standard error in parentheses from 100 replications for Blocks example with $n = 128$

Model	Doppler		
	RSNR=3	RSNR=5	RSNR=10
BSP-2	3.896(0.4928)	2.447(0.1774)	1.836(0.0444)
LOESS	8.891(1.506)	6.533(1.0853)	5.344(0.6362)
SS	3.644(0.587)	2.025(0.24)	1.251(0.0812)
NWK	4.045(1.102)	1.864(0.2683)	0.477(0.0624)
EBW	2.979(0.6397)	1.319(0.3142)	0.341(0.0855)
TF-0	4.172(0.9906)	1.891(0.2543)	0.49(0.0664)
TF-1	3.832(1.2397)	1.783(0.3894)	0.49(0.0688)
TF-2	4.122(1.2599)	1.834(0.3309)	0.489(0.0674)
GSP-L	5.845(1.3505)	5.313(1.3217)	4.843(0.9023)
GSP-R	12.164(2.9995)	11.588(3.1093)	12.402(3.5243)
BPP-10	2.911(0.6396)	1.275(0.2471)	0.559(0.2008)
BPP-21	2.575(0.5217)	1.463(0.3399)	1.166(0.3932)
BASS-1	2.865(0.6162)	1.167(0.2607)	0.353(0.0605)
BASS-2	2.841(0.5796)	1.753(0.2702)	1.344(0.1205)
LARMuK	3(0.6708)	1.212(0.3684)	0.364(0.1179)
LABS	**2.273(0.568)**	**0.848(0.2521)**	**0.234(0.0551)**

Table A.3: Average mean squared error with estimated standard error in parentheses from 100 replications for Doppler example with $n = 128$

Model	Heavisine		
	RSNR=3	RSNR=5	RSNR=10
BSP-2	2.399(0.4208)	0.926(0.1515)	0.305(0.0379)
LOESS	0.895(0.2248)	0.548(0.0976)	0.35(0.0436)
SS	0.875(0.2725)	0.484(0.1021)	0.235(0.0299)
NWK	1.022(0.352)	0.521(0.1321)	0.228(0.0472)
EBW	1.29(0.3473)	0.586(0.1365)	0.185(0.0456)
TF-0	1.668(0.4638)	0.831(0.1786)	0.306(0.051)
TF-1	1.129(0.6408)	0.581(0.2344)	0.205(0.0611)
TF-2	1.043(0.6797)	0.543(0.2195)	0.23(0.0692)
GSP-L	2.601(2.0255)	2.217(2.1273)	1.743(1.7561)
GSP-R	1.02(0.2869)	0.756(0.1707)	0.646(0.1152)
BPP-10	1.503(0.4239)	0.674(0.1576)	0.217(0.0616)
BPP-21	0.941(0.2702)	0.444(0.105)	0.147(0.032)
BASS-1	1.022(0.2434)	0.499(0.1047)	0.135(0.033)
BASS-2	**0.802(0.2053)**	0.452(0.0953)	0.176(0.0399)
LARMuK	1.13(0.3235)	0.541(0.16)	0.164(0.0566)
LABS	0.897(0.242)	**0.413(0.1492)**	**0.103(0.0406)**

Table A.4: Average mean squared error with estimated standard error in parentheses from 100 replications for Heavisine example with $n = 128$

Model	Bumps		
	RSNR=3	RSNR=5	RSNR=10
BSP-2	29.359(0.1163)	29.011(0.0419)	28.864(0.0105)
LOESS	43.468(5.4554)	40.385(7.3012)	36.495(7.2734)
SS	16.211(0.255)	15.406(0.123)	15.06(0.0537)
NWK	4.885(0.3559)	1.796(0.1194)	0.482(0.0319)
EBW	2.42(0.3291)	0.914(0.0992)	**0.272(0.0279)**
TF-0	2.921(0.8496)	1.762(0.3298)	0.491(0.0319)
TF-1	4.716(0.9879)	1.965(0.1328)	0.492(0.0326)
TF-2	3.095(0.4119)	3.609(0.9328)	6.917(1.2674)
GSP-L	16.704(2.5691)	16.212(2.324)	16.224(2.5594)
GSP-R	39.297(1.3464)	39.046(1.3386)	38.885(1.2831)
BPP-10	1.9(0.6277)	1.429(0.6619)	1.352(0.8161)
BPP-21	4.698(0.9928)	4.428(1.1195)	4.308(1.1065)
BASS-1	2.497(0.5322)	1.679(0.4936)	1.356(0.559)
BASS-2	24.925(2.6356)	23.806(2.8435)	23.387(2.6943)
LARMuK	3.692(1.7663)	2.465(1.0197)	1.999(0.7813)
LABS	**1.371(0.6845)**	**0.619(0.1769)**	0.341(0.1905)

Table A.5: Average mean square error with estimated standard error in parentheses from 100 replications for Bumps example with $n = 512$

Model	Blocks		
	RSNR=3	RSNR=5	RSNR=10
BSP-2	5.097(0.118)	4.74(0.0425)	4.59(0.0106)
LOESS	3.755(0.2529)	3.095(0.1343)	2.922(0.0252)
SS	2.837(0.1609)	2.197(0.0713)	1.883(0.0256)
NWK	2.34(0.1831)	1.349(0.1149)	0.581(0.1147)
EBW	1.227(0.2009)	0.398(0.0753)	0.088(0.0177)
TF-0	0.65(0.1486)	0.237(0.057)	0.061(0.0199)
TF-1	1.97(0.2187)	0.763(0.0876)	0.193(0.0273)
TF-2	2.234(0.2525)	1.069(0.1013)	0.478(0.0785)
GSP-L	3.089(0.342)	2.613(0.3006)	2.522(0.3358)
GSP-R	13.847(1.3079)	13.783(1.3156)	13.622(1.2962)
BPP-10	0.493(0.137)	0.216(0.0819)	0.182(0.0848)
BPP-21	1.385(0.1974)	0.845(0.127)	0.665(0.1055)
BASS-1	0.942(0.195)	0.436(0.1189)	0.273(0.1078)
BASS-2	3.033(0.2242)	2.613(0.182)	2.426(0.1763)
LARMuK	1.074(0.392)	0.646(0.2521)	0.395(0.1903)
LABS	**0.363(0.1391)**	**0.113(0.0562)**	**0.021(0.009)**

Table A.6: Average mean squared error with estimated standard error in parentheses from 100 replications for Blocks example with $n = 512$

Model	Doppler		
	RSNR=3	RSNR=5	RSNR=10
BSP-2	2.875(0.1038)	2.534(0.0374)	2.39(0.0093)
LOESS	2.756(0.3006)	2.044(0.0474)	1.893(0.0202)
SS	1.993(0.1234)	1.385(0.0531)	1.086(0.0174)
NWK	1.649(0.1233)	0.853(0.105)	0.431(0.0289)
EBW	1.263(0.1618)	0.592(0.0809)	**0.156(0.0209)**
TF-0	1.774(0.1822)	0.888(0.1084)	0.344(0.0595)
TF-1	1.434(0.1904)	0.715(0.1636)	0.263(0.0812)
TF-2	1.504(0.4085)	0.772(0.1816)	0.292(0.0914)
GSP-L	2.167(0.2292)	1.767(0.2376)	1.621(0.2526)
GSP-R	9.389(1.6093)	9.432(1.6343)	9.1(1.577)
BPP-10	1.36(0.1866)	0.632(0.0819)	0.22(0.0283)
BPP-21	1.055(0.1663)	**0.496(0.0768)**	
BASS-1	1.116(0.1587)	0.602(0.064)	0.363(0.024)
BASS-2	**1.051(0.1916)**	0.588(0.0804)	
LARMuK	1.584(0.2401)	1.04(0.1675)	0.652(0.1067)
LABS	1.243(0.2135)	0.66(0.1141)	0.343(0.0843)

Table A.7: Average mean squared error with estimated standard error in parentheses from 100 replications for Doppler example with $n = 512$

Model	Heavisine		
	RSNR=3	RSNR=5	RSNR=10
BSP-2	0.635(0.1112)	0.294(0.04)	0.15(0.01)
LOESS	0.464(0.067)	0.306(0.0523)	0.172(0.0531)
SS	0.384(0.0699)	0.225(0.0299)	0.113(0.0102)
NWK	0.398(0.0789)	0.223(0.0327)	0.106(0.0113)
EBW	0.435(0.1007)	0.202(0.0457)	0.066(0.0127)
TF-0	0.619(0.0906)	0.307(0.0399)	0.118(0.0126)
TF-1	0.391(0.0916)	0.198(0.036)	0.082(0.0107)
TF-2	0.393(0.0839)	0.211(0.0321)	0.095(0.0111)
GSP-L	0.785(0.246)	0.402(0.1555)	0.261(0.1383)
GSP-R	0.422(0.0677)	0.322(0.0409)	0.279(0.0372)
BPP-10	0.431(0.1015)	0.173(0.0392)	0.055(0.0108)
BPP-21	0.308(0.0771)	0.134(0.0328)	0.04(0.0091)
BASS-1	0.365(0.0894)	0.149(0.0348)	0.046(0.011)
BASS-2	0.354(0.0709)	0.169(0.0385)	0.063(0.0131)
LARMuK	0.413(0.1155)	0.19(0.0543)	0.074(0.0206)
LABS	**0.291(0.1185)**	**0.103(0.0508)**	**0.031(0.0128)**

Table A.8: Average mean squared error with estimated standard error in parentheses from 100 replications for Heavisine example with $n = 512$

A.1.5 Derivation of the full conditionals for LABS

In this appendix, we derive the full conditional distributions of some parameters required for Gibbs sampling. The full conditional posterior of each parameter can be easily obtained via conjugacy properties. Let us first find the full conditional posterior for $\beta_{p,q}$.

- Full conditional posterior for $\beta_{p,q}$

For each $q = 1, \ldots, J_p$,

$$\begin{aligned}
[\beta_{p,q} \,|\, \beta_{p,-q}, \text{ others, } \mathbf{Y}] \propto & \left[\exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k\in S}\sum_{l=1}^{J_k} B_{k,l}(x_i;\boldsymbol{\xi}_{k,l})\beta_{k,l})^2\right\}\right] \\
& \times \left[\exp\left\{-\sum_{k\in S}\sum_{l=1}^{J_k}\frac{\beta_{k,l}^2}{2\sigma_k^2}\right\}\right] \\
& \propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k\in S\backslash\{p\}}\sum_{l=1}^{J_k} B_{k,l}(x_i;\boldsymbol{\xi}_{k,l})\beta_{k,l}\right. \\
& \qquad \left. - \sum_{l=1}^{J_p} B_{p,l}(x_i;\boldsymbol{\xi}_{p,l})\beta_{p,l})^2 - \frac{1}{2\sigma_p^2}\sum_{k=1}^{J_p}\beta_{p,l}^2\right\}
\end{aligned}$$

For convenience, we set $c_i = \beta_0 + \sum_{k\in S\backslash\{p\}}\sum_{l=1}^{J_k} B_{k,l}(x_i;\boldsymbol{\xi}_{k,l})\beta_{k,l}$ as a constant term. Then,

$$\begin{aligned}
& \propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - c_i - \sum_{l=1}^{J_p} B_{p,l}(x_i;\boldsymbol{\xi}_{p,l})\beta_{p,l})^2 - \frac{1}{2\sigma_p^2}\sum_{l=1}^{J_p}\beta_{p,l}^2\right\} \\
& \propto \exp\left\{-\frac{1}{2\sigma^2}\sum_{i=1}^{n}(y_i - c_i - \sum_{l\neq q}^{J_p} B_{p,l}(x_i;\boldsymbol{\xi}_{p,l})\beta_{p,l} - B_{p,q}(x_i;\boldsymbol{\xi}_{p,q})\beta_{p,q})^2 - \frac{\beta_{p,q}^2}{2\sigma_p^2}\right\} \\
& \propto \exp\left\{-\frac{1}{2\sigma^2}\left(\beta_{p,l}^2\sum_{i=1}^{n}\left(B_{p,q}(x_i;\boldsymbol{\xi}_{p,q})\right)^2 - 2\beta_{p,q}\sum_{i=1}^{n}\left(y_i - c_i - \sum_{l\neq q}^{J_p} B_{p,l}(x_i;\boldsymbol{\xi}_{p,q})\beta_{p,l}\right)\right.\right. \\
& \qquad \left.\left. \times B_{p,q}(x_i;\boldsymbol{\xi}_{p,q})\right) - \frac{\beta_{p,q}^2}{2\sigma_p^2}\right\} \\
& = \exp\left\{-\frac{1}{2}\left(\left(\frac{\sum_{i=1}^{n}\left(B_{p,q}(x_i;\boldsymbol{\xi}_{p,q})\right)^2}{\sigma^2} + \frac{1}{\sigma_p^2}\right)\beta_{p,q}^2\right.\right. \\
& \qquad \left.\left. -2\frac{\sum_{i=1}^{n}\left(y_i - c_i - \sum_{l\neq q}^{J_p} B_{p,l}(x_i;\boldsymbol{\xi}_{p,q})\beta_{p,l}\right)\left(B_{p,q}(x_i;\boldsymbol{\xi}_{p,q})\right)}{\sigma^2}\beta_{p,q}\right)\right\}
\end{aligned}$$

Thus, the full conditional distribution for $\beta_{p,q}$ is

$$[\beta_{p,q} \,|\, \beta_{p,-q}, \text{ others, } \mathbf{Y}] \sim \mathcal{N}(\mu_{p_0}, \sigma^2_{p_0})$$

with

$$\sigma^2_{p_0} = \left(\frac{\sum_{i=1}^{n} \big(B_{p,q}(x_i; \boldsymbol{\xi}_{p,q})\big)^2}{\sigma^2} + \frac{1}{\sigma^2_p} \right)^{-1}$$
$$\mu_{p_0} = \sigma^2_{p_0} \times \frac{\sum_{i=1}^{n} \Big(y_i - c_i - \sum_{l \neq q}^{J_p} B_{p,l}(x_i; \boldsymbol{\xi}_{p,q})\beta_{p,l}\Big) \big(B_{p,q}(x_i; \boldsymbol{\xi}_{p,q})\big)}{\sigma^2}.$$

- Full conditional posterior of M_k

  For each $k \in S$,

$$\begin{aligned} [M_k \,|\, \text{others}] &\propto M_k^{J_k} \exp\{-M_k\} \times M_k^{a_{\gamma_k}-1} \exp\{-b_{\gamma_k} M_k\} \\ &= M^{J_k + a_{\gamma_k} - 1} \exp\{-(1 + b_{\gamma_k}) M_k\} \end{aligned}$$

The full conditional distribution for M_k is given by

$$[M_k \,|\, \text{others}] \sim \text{Ga}(a_k, b_k)$$

where

$$\begin{aligned} a_k &= a_{\gamma_k} + J_k, \\ b_k &= b_{\gamma_k} + 1. \end{aligned}$$

- Full conditional posterior of σ^2

$$\begin{aligned}
[\sigma^2 \,|\, \text{others}, \mathbf{Y}] \propto & \left[(\sigma^2)^{-\frac{n}{2}} \times \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{k \in S} \sum_{l=1}^{J_k} B_k(x; \boldsymbol{\xi}_{k,l}) \beta_{k,l})^2 \right\} \right] \\
& \times \left[(\sigma^2)^{-\frac{r}{2}+1} \times \exp\left\{ -\frac{rR}{2\sigma^2} \right\} \right] \\
& \propto (\sigma^2)^{-\frac{n+r}{2}+1} \exp\left\{ -\frac{1}{2\sigma^2} \sum_{i=1}^{n} (y_i - \beta_0 - \sum_{k \in S} \sum_{l=1}^{J_k} B_k(x; \boldsymbol{\xi}_{k,l}) \beta_{k,l})^2 - \frac{rR}{2\sigma^2} \right\}
\end{aligned}$$

The full conditional distribution for σ^2 is

$$[\sigma^2 \,|\, \text{others}, \mathbf{Y}] \sim \text{IG}\left(\frac{r_0}{2}, \frac{r_0 R_0}{2} \right)$$

with

$$\begin{aligned}
r_0 &= r + n, \\
R_0 &= \frac{\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{k \in S} \sum_{l=1}^{J_k} B_k(x; \boldsymbol{\xi}_{k,l}) \beta_{k,l})^2 + rR}{r_0}.
\end{aligned}$$

Bibliography

Abramovich, F., Sapatinas, T., and Silverman, B. W. (1998). Wavelet thresholding via a bayesian approach. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(4):725–749.

Albert, J. H. and Chib, S. (1993). Bayesian analysis of binary and polychotomous response data. *Journal of the American statistical Association*, 88(422):669–679.

Angrist, J. D. and Pischke, J.-S. (2014). *Mastering'metrics: The path from cause to effect*. Princeton University Press.

Bakin, S., Hegland, M., and Osborne, M. R. (2000). Parallel mars algorithm based on b-splines. *Computational Statistics*, 15(4):463–484.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152.

Breiman, L. (1996). Bagging predictors. *Machine learning*, 24(2):123–140.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794.

Chipman, H. A., George, E. I., McCulloch, R. E., et al. (2010). Bart: Bayesian additive regression trees. *The Annals of Applied Statistics*, 4(1):266–298.

Chu, J.-H., Clyde, M. A., and Liang, F. (2009). Bayesian function estimation using continuous wavelet dictionaries. *Statistica Sinica*, pages 1419–1438.

Cohen, A. (2003). *Numerical analysis of wavelet methods*. Elsevier.

Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297.

Cox, M. G. (1972). The numerical evaluation of b-splines. *IMA Journal of Applied Mathematics*, 10(2):134–149.

Crainiceanu, C. M., Ruppert, D., Carroll, R. J., Joshi, A., and Goodner, B. (2007). Spatially adaptive bayesian penalized splines with heteroscedastic errors. *Journal of Computational and Graphical Statistics*, 16(2):265–288.

De Boor, C. (1972). On calculating with b-splines. *Journal of Approximation Theory*, 6(1):50–62.

Denison, D., Mallick, B., and Smith, A. (1998a). Automatic bayesian curve fitting. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 60(2):333–350.

Denison, D. G., Mallick, B. K., and Smith, A. F. (1998b). Bayesian mars. *Statistics and Computing*, 8(4):337–346.

DeVore, R. A. and Lorentz, G. G. (1993). *Constructive Approximation*, volume 303. Springer Science & Business Media.

DiMatteo, I., Genovese, C. R., and Kass, R. E. (2001). Bayesian curve-fitting with free-knot splines. *Biometrika*, 88(4):1055–1071.

Donoho, D. L. and Johnstone, I. M. (1995). Adapting to unknown smoothness via wavelet shrinkage. *Journal of the american statistical association*, 90(432):1200–1224.

Donoho, D. L. and Johnstone, J. M. (1994). Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455.

Feng, D. (2013). miscf: Miscellaneous functions.

Francom, D. and Sanso, B. (2016). Bass: Bayesian adaptive spline surfaces.

Francom, D. and Sansó, B. (2020). Bass: An r package for fitting and performing sensitivity analysis of bayesian adaptive spline surfaces. *Journal of Statistical Software*, 94(1):1–36.

Francom, D., Sansó, B., Kupresanin, A., and Johannesson, G. (2018). Sensitivity analysis and emulation for functional data using bayesian adaptive splines. *Statistica Sinica*, 28:791–816.

Freund, Y., Schapire, R., and Abe, N. (1999). A short introduction to boosting. *Journal-Japanese Society For Artificial Intelligence*, 14(771-780):1612.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The annals of statistics*, pages 1–67.

Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Gijbels, I., Lambert, A., and Qiu, P. (2007). Jump-preserving regression and smoothing using local linear fitting: a compromise. *Annals of the Institute of Statistical Mathematics*, 59(2):235–272.

Green, P. J. (1995). Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732.

Hastie, T., Tibshirani, R., et al. (2000). Bayesian backfitting (with comments and a rejoinder by the authors. *Statistical Science*, 15(3):196–223.

Hwang, J.-N., Lay, S.-R., Maechler, M., Martin, R. D., and Schimert, J. (1994). Regression modeling in back-propagation and projection pursuit learning. *IEEE Transactions on neural networks*, 5(3):342–353.

Imaizumi, M. and Fukumizu, K. (2019). Deep neural networks learn non-smooth functions effectively. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 869–878. PMLR.

Johnstone, I. M. and Silverman, B. W. (2005). Empirical bayes selection of wavelet thresholds. *Annals of Statistics*, pages 1700–1752.

Karatzoglou, A., Smola, A., Hornik, K., Maniscalco, M. A., Teo, C. H., and (NICTA), N. I. A. (2004). kernlab: Kernel-based machine learning lab.

Kimeldorf, G. and Wahba, G. (1971). Some results on tchebycheffian spline functions. *Journal of mathematical analysis and applications*, 33(1):82–95.

Koo, J.-Y. (1997). Spline estimation of discontinuous regression functions. *Journal of Computational and Graphical Statistics*, 6(3):266–284.

Krivobokova, T., Crainiceanu, C. M., and Kauermann, G. (2008). Fast adaptive penalized splines. *Journal of Computational and Graphical Statistics*, 17(1):1–20.

Lee, J. (2007). Sampling methods of neutral to the right processes. *Journal of Computational and Graphical Statistics*, 16(3):656–671.

Lee, J. and Kim, Y. (2004). A new algorithm to generate beta processes. *Computational Statistics & Data Analysis*, 47(3):441–453.

Lee, T. C. (2002). Automatic smoothing for discontinuous regression functions. *Statistica Sinica*, pages 823–842.

Lee, Y., Mano, S., and Lee, J. (2020). Bayesian curve fitting for discontinuous functions using an overcomplete system with multiple kernels. *Journal of the Korean Statistical Society*, pages 1–21.

Lewicki, M. S. and Sejnowski, T. J. (2000). Learning overcomplete representations. *Neural computation*, 12(2):337–365.

Linero, A. R. (2018). Bayesian regression trees for high-dimensional prediction and variable selection. *Journal of the American Statistical Association*, 113(522):626–636.

Linero, A. R. and Yang, Y. (2018). Bayesian regression tree ensembles that adapt to smoothness and sparsity. *Journal of the royal statistical society: series B (statistical methodology)*, 80(5):1087–1110.

Liu, Z. and Guo, W. (2010). Data driven adaptive spline smoothing. *Statistica Sinica*, pages 1143–1163.

Luo, Z. and Wahba, G. (1997). Hybrid adaptive splines. *Journal of the American Statistical Association*, 92(437):107–116.

Meyer, D. and Wien, F. T. (2015). Support vector machines. *The Interface to libsvm in package e1071*, 28.

Nott, D. J., Kuk, A. Y., and Duc, H. (2005). Efficient sampling schemes for bayesian mars models with many predictors. *Statistics and Computing*, 15(2):93–101.

Ouyang, Z. (2008). *Bayesian Additive Regression Kernels*. PhD dissertation, Duke University.

Petrushev, P. P. (1988). Direct and converse theorems for spline and rational approximation and besov spaces. In *Function spaces and applications*, pages 363–377. Springer.

Pillai, N. S. (2008). *Lévy random measures: Posterior consistency and applications*. PhD dissertation, Duke University.

Pillai, N. S., Wu, Q., Liang, F., Mukherjee, S., and Wolpert, R. L. (2007). Characterizing the function space for bayesian kernel models. *Journal of Machine Learning Research*, 8(Aug):1769–1797.

Pintore, A., Speckman, P., and Holmes, C. C. (2006). Spatially adaptive smoothing splines. *Biometrika*, 93(1):113–125.

Qiu, P. (2003). A jump-preserving curve fitting procedure based on local piecewise-linear kernel estimation. *Journal of Nonparametric Statistics*, 15(4-5):437–453.

Qiu, P. and Yandell, B. (1998). Local polynomial jump-detection algorithm in nonparametric regression. *Technometrics*, 40(2):141–152.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ruppert, D. and Carroll, R. J. (2000). Theory & methods: Spatially-adaptive penalties for spline fitting. *Australian & New Zealand Journal of Statistics*, 42(2):205–223.

Silverman, B. W., Evers, L., Xu, K., Carbonetto, P., and Stephens, M. (2005). Ebayesthresh: Empirical bayes thresholding and related methods.

Simoncelli, E. P., Freeman, W. T., Adelson, E. H., and Heeger, D. J. (1992). Shiftable multiscale transforms. *IEEE Transactions on Information Theory*, 38(2):587–607.

Smith, M. and Kohn, R. (1996). Nonparametric regression using bayesian variable selection. *Journal of Econometrics*, 75(2):317–343.

Tibshirani, R. J. et al. (2014). Adaptive piecewise polynomial estimation via trend filtering. *Annals of statistics*, 42(1):285–323.

Tipping, M. E. (2000). The relevance vector machine. In *Advances in neural information processing systems*, pages 652–658.

Tu, C. (2006). *Bayesian nonparametric modeling using Lèvy process priors with applications for function estimation, time series modeling and spatio-temporal modeling.* PhD dissertation, Duke University.

Wahba, G. (1990). *Spline models for observational data*, volume 59. Siam.

Wang, X., Du, P., and Shen, J. (2013). Smoothing splines with varying smoothing parameter. *Biometrika*, 100(4):955–970.

Wang, X.-F. (2010). fancova: Nonparametric analysis of covariance.

Wolpert, R. L., Clyde, M. A., Tu, C., et al. (2011). Stochastic expansions using continuous dictionaries: Lévy adaptive regression kernels. *The Annals of Statistics*, 39(4):1916–1962.

Xia, Z. and Qiu, P. (2015). Jump information criterion for statistical inference in estimating discontinuous curves. *Biometrika*, 102(2):397–408.

Yang, L. and Hong, Y. (2017). Adaptive penalized splines for data smoothing. *Computational Statistics & Data Analysis*, 108:70–83.

Yang, Y. and Song, Q. (2014). Jump detection in time series nonparametric regression models: a polynomial spline approach. *Annals of the Institute of Statistical Mathematics*, 66(2):325–344.

Zhou, S. and Shen, X. (2001). Spatially adaptive regression splines and accurate knot selection schemes. *Journal of the American Statistical Association*, 96(453):247–259.

국문초록

본 학위 논문에서는 함수의 변화하는 부드러움을 추정하기 위해 LARK 모형을 확장한 “레비 적응 B-스플라인 회귀 모형” (LABS) 을 제안한다. 즉, 제안한 모형은 B-스플라인 기저들이 생성 커널로 갖는 LARK 모형이다. 제안한 모형은 B-스플라인 기저의 차수를 조정하면서 불연속하거나 최고점 등을 지닌 함수의 부드러움에 체계적으로 적응한다. 모의 실험들과 실제 자료 분석을 통해서 제안한 모형이 불연속점, 최고점, 곡선 부분을 모두 잘 추정하고 있음을 입증하고, 거의 모든 실험에서 최고의 성능을 발휘한다. 또한, B-스플라인 차수에 따라 LABS 모형의 평균 함수가 특정 베소프 공간에 존재하고, LABS 모형의 사전분포가 해당 베소프 공간에 상당히 넓은 받침을 갖는다는 것을 밝힌다.

추가적으로, 텐서곱 B-스플라인 기저를 도입하여 다차원 자료를 분석할 수 있는 LABS 모형을 개발한다. 제안한 모형을 “다차원 레비 적응 B-스플라인 회귀 모형” (MLABS) 이라고 명명한다. MLABS 모형은 회귀 및 분류 문제들에서 최신 모형들과 필적할만한 성능을 갖추고 있다. 특히, MLABS 모형이 저차원 회귀 문제들에서 최신 비모수 회귀 모형들보다 안정적이고 정확한 예측 능력을 지니고 있음을 실험들을 통해 보인다.

주요어: 레비 랜덤 측도; 베소프 공간; 텐서곱 B-스플라인 기저; 가역 점프 마르코프 체인 몬테 카를로

학번: 2015-20297