TBASS: A Robust Adaptation of Bayesian Adaptive Spline Surfaces

Andy A. Shen, Kellin N. Rumsey, Devin C. Francom

Statistical Sciences Group (CCS-6): Los Alamos National Laboratory

Abstract

The R package ‘TBASS‘ is an extension of the ‘BASS‘ package created by Francom et. al (2016). The package is used to fit a Bayesian adaptive spline surface to a dataset that follows a Student’s t-distribution or has outliers. Much of the framework for ‘TBASS‘ is adapted from the concepts of Bayesian Multivariate Adaptive Regression Splines (BMARS), specifically the work done from Denison, Mallick, and Smith (1998). By including a more robust generalization, a dataset with outliers can now be accurately fit using the BMARS model, without the possibility of overfitting or variance inflation.

Keywords: splines, robust regression, Bayesian inference, nonparametric regression, sensitivity analysis

1 Introduction

Splines are a commonly used regression tool for fitting nonlinear data. Splines can act as basis functions, where all of the basis combine to form the $\boldsymbol{X}$ matrix. The simplest way to create the ith basis functions can be represented as

$$X_{ij} = [s_i(x_j - t_i)]_+ \tag{1}$$

Equation (1) is used to calculate the ith column of the X matrix of basis functions, where $s_i \in -1, 1$, t_i is called a **knot** and $[a]_+ = max(0, a)$.

For example, given the nonlinear data shown in Figure 1 below, we can use (1) to fit a spline model shown in Figure 2.

Figure 1 : Univariate Nonlinear Data

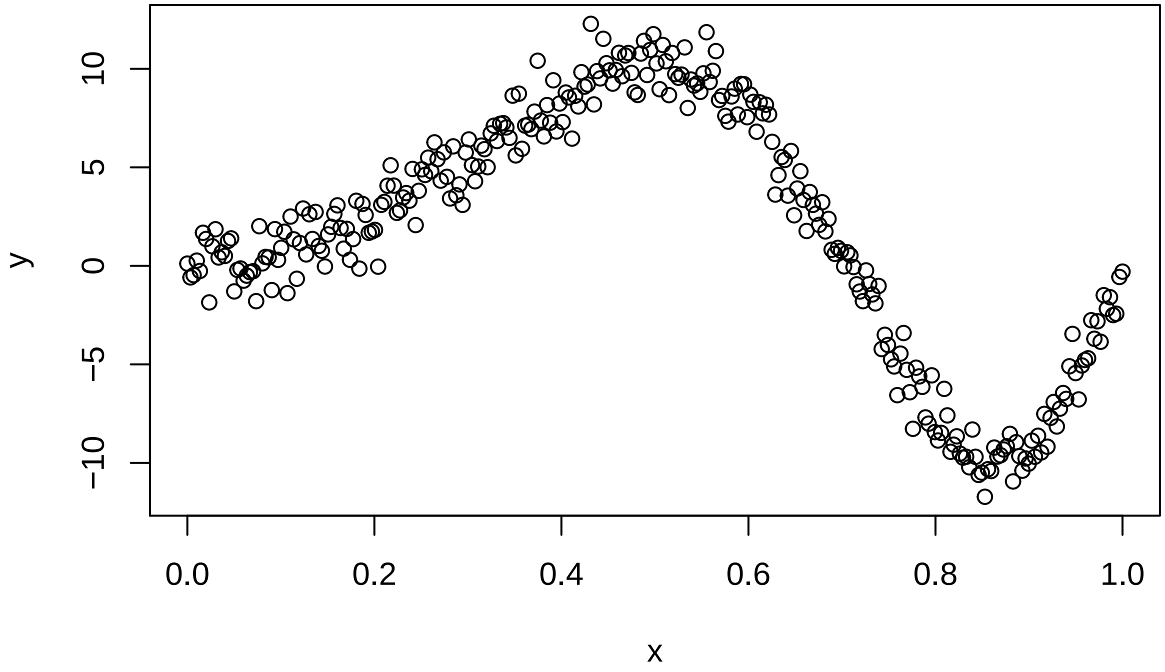
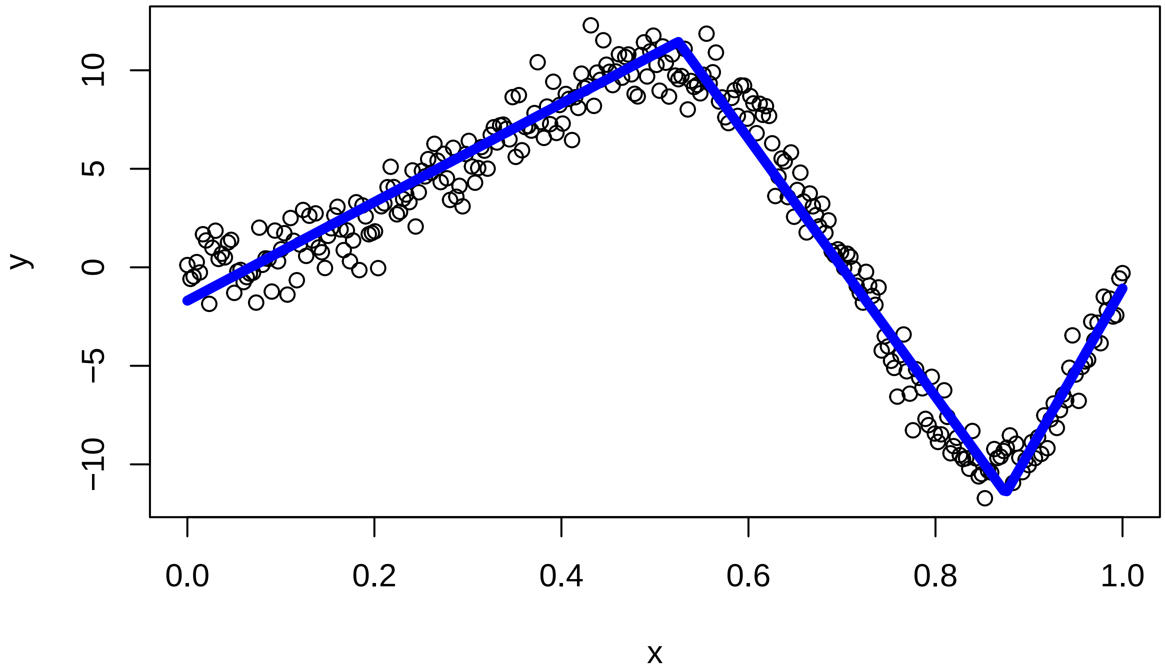


Figure 2 : Univariate Spline Function



2 Robust BMARS

2.1 Overview

We want to extend the theory behind frequentist univariate spline regression to a multivariate Bayesian framework. Moreover, we want to be able to fit nonlinear data that has outliers. We adopt the Gaussian BMARS framework based on the standard framework used by Dennison et al. (1998) and Francom (2018) when deriving our likelihood and full conditional distributions for the parameters.

In the presence of outliers, Gaussian BMARS will attempt to capture the excess noise by adding basis functions (overfitting) or inflating the variance term (σ^2). The Robust BMARS model accounts for this sensitivity to outliers by avoiding overfitting or variance inflation when the degrees of freedom (ν) are low. When ν is high, the t-distribution closely mimics a normal distribution, so the Robust BMARS model behaves in a similar way.

The function used to fit the Robust BMARS model in the `TBASS` package is the `tbass()` command (see section 4).

2.2 Auxiliary Variables

In creating the Robust BMARS model, we introduce new auxiliary parameters based on the work done by Gelman et al. (2014).

Similar to Gaussian BMARS, we let y_i be our dependent variable and $\mathbf{x}_i$ be our independent variable representing a single basis function with $i = 1, ..., n$. Without loss of generality, all independent variables $\mathbf{x}_i$ are scaled from zero to one (Francom, 2016).

In the robust case, y_i is modeled as

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \epsilon, \quad \epsilon \sim t_\nu \left(0,\ \sigma^2 \boldsymbol{V}^{-1}\right) \tag{2}$$

and

$$y_i | V_i \sim \mathcal{N}\left(\boldsymbol{X}\boldsymbol{\beta},\ \frac{\sigma^2}{V_i}\right) \tag{3}$$

where $\boldsymbol{X}$ represents the matrix of basis functions by column, $\boldsymbol{\beta}$ is the vector of regression coefficients, ϵ is the error term, ν represents the degrees of freedom in a Student's t-distribubtion, σ^2 is the variance term for y_i, and $V^{-1} = diag(1/V_1 ... 1/V_n)$ where V_i is the variance estimate of y_i.

The basis functions themselves are produced the same way as in the `BASS` package by Francom, et al. (2016).

2.3 Priors

We assume an Inverse-Gamma prior for σ^2 with default shape $\gamma_1 = 0$ and default rate $\gamma_2 = 0$, and a Gamma prior with shape and rate $\frac{\nu}{2}$ for V_i, such that

$$\sigma^2 \sim IG\ (\gamma_1,\ \gamma_2) \tag{4}$$

$$V_i \sim \Gamma\left(\frac{\nu}{2},\ \frac{\nu}{2}\right) \tag{5}$$

From there, we obtain the full conditional of V_i as

$$V_i|\cdot \sim \Gamma \left\{ \frac{\nu+1}{2},\ \frac{1}{2\sigma^2} \sum_{i=1}^{n} (y - X\beta)^2 \right\} \tag{6}$$

For the parameter governing the number of basis functions λ, we have that

$$\lambda|\cdot \sim \Gamma \left(h_1 + M,\ h_2 + 1 \right) \tag{7}$$

where $h_1 = h_2 = 10$ are the default hyperparameters for λ and M is the current number of basis functions. It follows that λ follows a gamma prior.

2.4 Regression Coefficients

Finally, our regression coefficients $\boldsymbol{\beta}$ follow a Gaussian prior such that

$$\beta|\cdot \sim \mathcal{N} \left(0,\ \tau^2 \boldsymbol{I} \right) \tag{8}$$

In the Student's t-distribution, we can marginalize the posterior for $\boldsymbol{\beta}$ and σ^2 to obtain the regression estimate $\hat{\boldsymbol{\beta}}$:

$$\hat{\boldsymbol{\beta}} = \frac{1}{\sigma^2} \left(\frac{1}{\sigma^2} \boldsymbol{X'VX} + \tau^{-2} \boldsymbol{I} \right)^{-1} \boldsymbol{X'Vy} \tag{9}$$

where τ^2 is the prior variance for β_i.

2.5 Reversible-Jump Markov Chain Monte Carlo (RJ-MCMC)

Like Gaussian BMARS, the robust algorithm builds basis functions adaptively, sampling from candidate knot locations, signs, interaction degrees, and accepting or rejecting the basis functions using a RJ-MCMC algorithm to sample from the full posterior.

RJ-MCMC is an generalization of the traditional Metropolis-Hastings algorithm in the sense that RJ-MCMC allows for parameter dimension change, allowing for simulation when the number of parameters is unknown. This is important for BMARS because we want to learn where the knots should be placed and how many basis functions to have in our model, along with the degree of interaction for our basis functions. There can exist multiple basis functions in a multivariate setting. We also want to know if certain basis functions should be added, deleted, or changed.

The BMARS model has three possible move types, which are sampled using a discrete uniform:

- **Birth**: adding a basis function
- **Death**: deleting a basis function
- **Change**: changing a knot, sign, and values of a basis function

Once the move type is sampled, the RJ-MCMC algorithm is used to determine acceptance of that move type.

Our acceptance ratio α is denoted by

$$\alpha = min \left\{ 1,\ \frac{L(D|\theta')\ p(\theta')\ S(\theta' \to \theta)}{L(D|\theta)\ p(\theta)\ S(\theta \to \theta')} \right\} \tag{10}$$

where θ' represents the candidate model parameters and θ represent the current state model parameters, L is the Gaussian likelihood, p is the prior, and S is the proposal to jump from one model to another.

Section 3 details the RJ-MCMC algorithm for the birth step in detail, and the death and change steps are very similar.

2.6 Gibbs Sampling

Once the basis function move type is complete, the model parameter values can then be sampled using Gibbs Sampling, since the full conditionals are all closed-form.

3 Birth Step

3.1 Likelihood Function

Estimate (9) allows us to achieve our t-distributed likelihood function for the birth step:

$$(\tau^2)^{\frac{M+1}{2}}|V|^{-1/2}\ \left|(X^tW^{-1}X+\tau^{-2}I)^{-1}\right|^{-1/2} exp\bigg\{-\frac{1}{2}\big(y'V^{-1}y-\hat{\boldsymbol{\beta}}^t(X^tW^{-1}X+\tau^{-2}I)^{-1}\hat{\boldsymbol{\beta}}\big)\bigg\} \tag{11}$$

4 Simulation with tbass()

We now demonstrate the capabilities of the TBASS package using the main command, tbass(). For all parameter values of this function, please refer to the help documentation by running ?tbass after loading the package.

We begin by loading in the package and setting the seed for reproducibility. The package can be installed using the following command: devtools::install_github("aashen12/TBASS").

```
set.seed(12)
library(TBASS)
```

4.1 Friedman Function

Our first example fits the Robust BMARS model to the infamous Friedman Function (reference here):

$$10\ sin(x_1\ x_2\ \pi)\ +\ 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 \tag{12}$$