stats

MDPI

Article

Precise Tensor Product Smoothing via Spectral Splines

Nathaniel E. Helwig [1,2]

[1] Department of Psychology, University of Minnesota, 75 E River Road, Minneapolis, MN 55455, USA; helwig@umn.edu
[2] School of Statistics, University of Minnesota, 224 Church Street SE, Minneapolis, MN 55455, USA

Abstract: Tensor product smoothers are frequently used to include interaction effects in multiple nonparametric regression models. Current implementations of tensor product smoothers either require using approximate penalties, such as those typically used in generalized additive models, or costly parameterizations, such as those used in smoothing spline analysis of variance models. In this paper, I propose a computationally efficient and theoretically precise approach for tensor product smoothing. Specifically, I propose a spectral representation of a univariate smoothing spline basis, and I develop an efficient approach for building tensor product smooths from marginal spectral spline representations. The developed theory suggests that current tensor product smoothing methods could be improved by incorporating the proposed tensor product spectral smoothers. Simulation results demonstrate that the proposed approach can outperform popular tensor product smoothing implementations, which supports the theoretical results developed in the paper.

Keywords: generalized additive model; linear mixed model; multiple nonparametric regression; penalized spline; smoothing spline analysis of variance

1. Introduction

Consider a multiple nonparametric regression model [1] of the form

$$Y = f(\boldsymbol{X}) + \epsilon \tag{1}$$

where $Y \in \mathbb{R}$ is the observed response variable, $\boldsymbol{X} = (X_1, \dots, X_p)^\top \in \mathcal{X}$ is the observed predictor vector, $\mathcal{X} = \mathcal{X}_{(1)} \times \cdots \times \mathcal{X}_{(p)}$ is the product domain with $\mathcal{X}_{(j)}$ denoting the domain of the j-th predictor, $f : \mathcal{X} \to \mathbb{R}$ is the (unknown) real-valued function connecting the response and predictors, and ϵ is an error term that satisfies $E(\epsilon) = 0$ and $E(\epsilon^2) = \sigma^2 < \infty$. Note that this implies that $E(Y|\boldsymbol{X}) = f(\boldsymbol{X})$, i.e., the function $f(\cdot)$ is the conditional expectation of the response variable Y given the predictor vector $\boldsymbol{X}$. Given a sample of training data, the goal is to estimate the unknown mean function f without having any a priori information about the parametric nature of the functional relationship (e.g., without assuming linearity).

Let $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ denote a sample of n independent observations from the model in Equation (1), where $y_i \in \mathbb{R}$ is the i-th observation's realization of the response variable, and $\mathbf{x}_i = (x_{i1}, \dots, x_{ip})^\top \in \mathcal{X}$ is the i-th observation's realization of the predictor vector. To estimate f, it is typical to minimize a penalized least squares functional of the form

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2 + \lambda P(f) \tag{2}$$

where $P(f) \geq 0$ denotes some non-negative penalty that describes the complexity of f, i.e., if $P(f) > P(g)$, then the function f is more complex (less smooth) than the function g, and the tuning parameter $\lambda \geq 0$ controls the influence of the penalty. To find a reasonable balance between fitting (the data) and smoothing (the function), λ is often chosen via cross-validation, information theory, or maximum likelihood estimation [2].

check for updates

Citation: Helwig, N.E. Precise Tensor Product Smoothing via Spectral Splines. *Stats* **2024**, *7*, 34–53. https://doi.org/10.3390/stats7010003

Academic Editors: Wei Zhu and Eddy Kwessi

Received: 9 December 2023
Revised: 27 December 2023
Accepted: 3 January 2024
Published: 10 January 2024

When the penalty P is a semi-norm in a (tensor product) reproducing kernel Hilbert space (RKHS), the minimizer of Equation (2) is referred to as a (tensor product) smoothing spline [1,3–7]. Note that (tensor product) smoothing splines are used within multiple nonparametric regression frameworks, such as generalized additive models (GAMs) [7,8] and smoothing spline analysis of variance (SSANOVA) models [3,5,6]. Such methods have proven powerful for nonparametric (multivariate) function estimation for a variety of different types of data, such as oceanography [9], social media [10], clinical biomechanics [11], self-esteem development [12], smile perception [13], clinical neuroimaging [14], psychiatry [15], and demography [16].

To find the $\hat{f}_\lambda$ that minimizes Equation (2), it is first necessary to specify the assumed model form. For example, with $p = 2$ predictors, we could consider one of two forms:

$$\begin{aligned} \text{additive}: &\quad f(\boldsymbol{X}) = f_0 + f_1(X_1) + f_2(X_2) \\ \text{interactive}: &\quad f(\boldsymbol{X}) = f_0 + f_1(X_1) + f_2(X_2) + f_{12}(X_1, X_2) \end{aligned}$$

where $\boldsymbol{X} = (X_1, X_2)^\top \in \mathcal{X} = \mathcal{X}_{(1)} \times \mathcal{X}_{(2)}$ is the bidimensional predictor, $f_0 \in \mathbb{R}$ is an intercept, $f_1 : \mathcal{X}_{(1)} \to \mathbb{R}$ is the main effect of the first predictor, $f_2 : \mathcal{X}_{(2)} \to \mathbb{R}$ is the main effect of the second predictor, and $f_{12} : \mathcal{X} \to \mathbb{R}$ is the two-way interaction effect. Note that these models are nested given that the additive model is equivalent to the interaction model if $f_{12} = 0$.

For additive models, f_j is typically represented by a spline basis of rank r_j, such as $f_j(X_j) = \mathbf{Z}_j^\top \boldsymbol{\beta}_j$. Note that $\mathbf{Z}_j = \left(Z_1^{(j)}(X_j), \dots, Z_{r_j}^{(j)}(X_j)\right)^\top \in \mathbb{R}^{r_j}$ denotes the known spline basis vector that depends on the chosen knots (later described), and $\boldsymbol{\beta}_j \in \mathbb{R}^{r_j}$ is the unknown coefficient vector. To define the complexity of each (additive) effect, it is typical to consider penalties of the form $P_j(f_j) = \boldsymbol{\beta}_j^\top \mathbf{Q}_j \boldsymbol{\beta}_j$, where $\mathbf{Q}_j$ is a semi-positive definite matrix. Using these representations of the function evaluation and penalty, Equation (2) can be written as

$$\frac{1}{n} \sum_{i=1}^{n} \left(y_i - f_0 - \sum_{j=1}^{p} \mathbf{z}_{ij}^\top \boldsymbol{\beta}_j \right)^2 + \sum_{j=1}^{p} \lambda_j \boldsymbol{\beta}_j^\top \mathbf{Q}_j \boldsymbol{\beta}_j \tag{3}$$

where $\mathbf{z}_{ij} = \left(Z_1^{(j)}(x_{ij}), \dots, Z_{r_j}^{(j)}(x_{ij})\right)^\top$ is the i-th observation's realization of the $\boldsymbol{Z}_j$ vector, and $\lambda_j \geq 0$ are tuning parameters that control the influence of each penalty.

When the model contains interaction effects, different approaches can be used to represent and penalize the interaction terms. In GAMs, it is typical to (i) represent interaction effects by taking an outer (Kronecker) product of marginal basis vectors, and (ii) penalize interaction effects using an equidistant (grid) approximation (see [7] (pp. 227–237)). In SSANOVA models, it is typical to represent and penalize interaction effects using a tensor product RK function (see [5] (pp. 40–48)). For a thorough comparison of the two approaches, see Helwig [1]. In both frameworks, estimation of interaction effects can be costly when using a moderate to large number of knots, which is true even when using scalable parameterizations and algorithms [9,17,18]. This is because efficient computational tools for exact tensor product function representation and penalization are lacking from the literature, which hinders the widespread application of tensor product smoothing splines.

To address this practical issue, this paper (i) proposes a spectral representer theorem for univariate smoothing spline estimators, and (ii) develops efficient computational strategies for constructing tensor product smoothing splines from marginal spectral representations. The marginal spectral spline representation that I propose is similar to that proposed by [19]; however, the version that I consider penalizes all of the non-constant functions of each predictor. The tensor product basis construction approach that I propose generally follows the idea proposed by [20], where tensor products are built from outer (Kronecker) products of marginal bases. However, unlike this approach, I leverage reproducing kernel theory to develop exact analytical penalties for tensor product smooth terms. The proposed approach makes it possible to fit tensor product smoothing spline models (a) with interaction effects between any combination of predictors, and (b) using any linear mixed modeling software.

The remainder of this paper is organized as follows: Section 2 provides background on the reproducing kernel Hilbert space theory relevant to univariate smoothing splines; Section 3 provides background on the tensor product smoothing splines; Section 4 proposes an alternative tensor product smoothing spline (like) framework that penalizes all non-constant functions of the predictor; Section 5 develops the spectral representer theories necessary for efficiently computing exact tensor product penalties; Section 6 conducts a simulation study to compare the proposed approach to an existing (comparable) method; Section 7 demonstrates the proposed approach using a real dataset; and Section 8 discusses potential extensions of the proposed approach.

2. Smoothing Spline Foundations

2.1. Reproducing Kernel Hilbert Spaces

Consider a single predictor (i.e., $p = 1$) that satisfies $x \in \mathcal{X}$, and let $\mathcal{H}$ denote a RKHS of functions on $\mathcal{X}$. The unknown function f from Equation (2) is assumed to be an element of $\mathcal{H}$, which will be denoted by $f \in \mathcal{H}$. Suppose that the space $\mathcal{H}$ can be decomposed into two orthogonal subspaces, such as $\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1$, where $\oplus$ denotes the tensor summation. Note that $\mathcal{H}_0 = \{f : P(f) = 0, f \in \mathcal{H}\}$ is the null space, which contains all functions (in $\mathcal{H}$) that have zero penalty, and $\mathcal{H}_1 = \{f : P(f) > 0, f \in \mathcal{H}\}$ is the contrast space, which contains all functions (in $\mathcal{H}$) that have a non-zero penalty. In some cases, the null space can be further decomposed, such as $\mathcal{H}_0 = \mathcal{H}_{00} \oplus \mathcal{H}_{01}$, where $\mathcal{H}_{00} = \{f : f \in \mathcal{H}_0, f(x) \propto 1\ \forall x \in \mathcal{X}\}$ is a space of constant functions (intercept), and $\mathcal{H}_{01} = \{f : f \in \mathcal{H}_0, f(x) \not\propto 1\ \forall x \in \mathcal{X}\}$ is a space of non-constant functions (unpenalized). For example, when using a cubic smoothing spline, $\mathcal{H}_{01}$ contains the linear effect of X, which is unpenalized.

The inner product of $\mathcal{H}$ will be denoted by $\langle f, g\rangle$ for any $f, g \in \mathcal{H}$, and the corresponding norm will be written as $\|f\| = \sqrt{\langle f, f\rangle}$ for any $f \in \mathcal{H}$. Given the tensor sum decomposition of $\mathcal{H}$, the inner product can be written as a summation of the corresponding subspaces' inner products, such as $\langle f, g\rangle = \langle f, g\rangle_0 + \langle f, g\rangle_1$. Note that $\langle f, g\rangle_0$ is the null space inner product for any $f, g \in \mathcal{H}_0$, and $\langle f, g\rangle_1$ is the contrast space inner product for any $f, g \in \mathcal{H}_1$. The corresponding norms will be denoted by $\|f\|_0 = \sqrt{\langle f, f\rangle_0}$ (norm of $\mathcal{H}_0$) and $\|f\|_1 = \sqrt{\langle f, f\rangle_1}$ (norm of $\mathcal{H}_1$). When the null space consists of non-constant functions, i.e, when $\mathcal{H}_0 = \mathcal{H}_{00} \oplus \mathcal{H}_{01}$, the null space inner product can be written as $\langle f, g\rangle_0 = \langle f, g\rangle_{00} + \langle f, g\rangle_{01}$, where $\langle f, g\rangle_{00}$ is the inner product of $\mathcal{H}_{00}$ for any $f, g \in \mathcal{H}_{00}$, and $\langle f, g\rangle_{01}$ is the inner product of $\mathcal{H}_{01}$ for any $f, g \in \mathcal{H}_{01}$. The corresponding norm can be written as $\|f\|_0 = \sqrt{\|f\|_{00}^2 + \|f\|_{01}^2}$, where $\|f\|_{00} = \sqrt{\langle f, f\rangle_{00}}$ and $\|f\|_{01} = \sqrt{\langle f, f\rangle_{01}}$ denote the norms of $\mathcal{H}_{00}$ and $\mathcal{H}_{01}$, respectively.

The RK of $\mathcal{H}$ will be denoted by $R(x, z) = R_z(x) = R_x(z)$ for any $x, z \in \mathcal{X}$. Note that the RK is an element of the RKHS, i.e., $R \in \mathcal{H}$ for any $x, z \in \mathcal{X}$. By definition, the RK is the representer of the evaluation functional in $\mathcal{H}$, which implies that the RK satisfies $f(x) = \langle R_x(z), f(z)\rangle$ for any $f \in \mathcal{H}$ and any $x, z \in \mathcal{X}$. This important property, which is referred to as the "reproducing property" of the (reproducing) kernel function, implies that any function in $\mathcal{H}$ can be evaluated through the inner product and RK function. Following the decompositions of the inner product, the RK function can be written as $R(x, z) = R_0(x, z) + R_1(x, z)$, where $R_0 \in \mathcal{H}_0$ and $R_1 \in \mathcal{H}_1$ denotes the RKs of $\mathcal{H}_0$ and $\mathcal{H}_1$, respectively. Furthermore, when $\mathcal{H}_0 = \mathcal{H}_{00} \oplus \mathcal{H}_{01}$, the null space RK can be decomposed such as $R_0(x, z) = R_{00}(x, z) + R_{01}(x, z)$, where $R_{00} \in \mathcal{H}_{00}$ and $R_{01} \in \mathcal{H}_{01}$ denotes the RKs of $\mathcal{H}_{00}$ and $\mathcal{H}_{01}$, respectively. By definition, $R_{00}(x, z) = \beta_0$ for all $x, z \in \mathcal{X}$, where $\beta_0 \in \mathbb{R}$ is some constant.

The tensor sum decomposition $\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1$ implies that any function $f \in \mathcal{H}$ can be written as a summation of two components, such as

$$f(x) = f_0(x) + f_1(x)$$

where $f_0 \in \mathcal{H}_0$ is the null space contribution and $f_1 \in \mathcal{H}_1$ is the contrast space contribution. Furthermore, when $\mathcal{H}_0 = \mathcal{H}_{00} \oplus \mathcal{H}_{01}$, the null space component can be further decomposed into its constant and non-constant contributions, such as $f_0(x) = f_{00}(x) + f_{01}(x)$, where $f_{00}(x) \propto 1$ for all $x \in \mathcal{X}$. Let $\mathcal{P}_0$ denote the projection operator for the null space, such that $\mathcal{P}_0 f = f_0$ for any $f \in \mathcal{H}$. Similarly, let $\mathcal{P}_1$ denote the projection operator for the contrast space, such that $\mathcal{P}_1 f = f_1$ for any $f \in \mathcal{H}$. Note that $f_0 \in \mathcal{H}_0$ is referred to as the "parametric component" of f, given that $\mathcal{H}_0$ is a finite dimensional subspace. In contrast, $f_1 \in \mathcal{H}_1$ is the "nonparametric component" of f, given that $\mathcal{H}_1$ is an infinite dimensional subspace.

2.2. Representer Theorem

Still consider a single predictor (i.e., $p = 1$) that satisfies $x \in \mathcal{X}$ with $\mathcal{H} = \mathcal{H}_0 \oplus \mathcal{H}_1$ denoting a RKHS of functions on $\mathcal{X}$. Now, suppose that the penalty functional in Equation (2) is defined to be the squared norm of the function's projection into the contrast space, i.e., $P(f) = \|\mathcal{P}_1 f\|^2 = \|f_1\|_1^2$. Note that the second equality is due to the fact that $\|f_1\|_0^2 = 0$ for any $f_1 \in \mathcal{H}_1$, which is a consequence of the orthogonality of $\mathcal{H}_0$ and $\mathcal{H}_1$. More specifically, given $\{(x_i, y_i)\}_{i=1}^n$, consider the problem of finding the function

$$\hat{f}_\lambda = \underset{f \in \mathcal{H}}{\arg\min} \left[\frac{1}{n} \sum_{i=1}^{n} (y_i - f(x_i))^2 + \lambda \|\mathcal{P}_1 f\|^2 \right] \tag{4}$$

where $\mathcal{P}_1$ is the projection operator for the contrast space $\mathcal{H}_1$. Note that the solution is subscripted with λ to emphasize the dependence on the tuning parameter.

Suppose that the null space has dimension $m \geq 1$. Note that $m = 1$ when $\mathcal{H}_0$ only consists of the constant (intercept) subspace, whereas $m \geq 2$ when $\mathcal{H}_0 = \mathcal{H}_{00} \oplus \mathcal{H}_{01}$. Let $\{N_0, N_1, \ldots, N_{m-1}\}$ denote a basis for the null space $\mathcal{H}_0$, such that any $f_0 \in \mathcal{H}_0$ can be written as $f_0(x) = \sum_{j=0}^{m-1} \beta_j N_j(x)$ for some coefficient vector $\boldsymbol{\beta} = (\beta_0, \ldots, \beta_{m-1})^\top \in \mathbb{R}^m$. The representer theorem of Kimeldorf and Wahba [21] reveals that the optimal smoothing spline estimator from Equation (4) has the form

$$f_\lambda(x) = \sum_{j=0}^{m-1} \beta_j N_j(x) + \sum_{i=1}^{n} \alpha_i R_1(x, x_i) \tag{5}$$

where $R_1 \in \mathcal{H}_1$ is the RK of the contrast space, and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{R}^n$ is the coefficient vector that combines the training data RK evaluations.

The representer theorem in Equation (5) reveals that the smoothing spline estimator can be written as $f_\lambda(x) = f_{0\lambda}(x) + f_{1\lambda}(x)$, where $f_{0\lambda}(x) = \sum_{j=0}^{m-1} \beta_j N_j(x)$ is the null space contribution and $f_{1\lambda}(x) = \sum_{i=1}^{n} \alpha_i R_1(x, x_i)$ is the contrast space contribution. Using the optimal representation from Equation (5), the penalty has the form

$$\begin{aligned} \|\mathcal{P}_1 f_\lambda\|^2 &= \| \textstyle\sum_{i=1}^{n} \alpha_i R_1(x, x_i) \|_1^2 \\ &= \textstyle\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \langle R_1(x, x_i), R_1(x, x_j) \rangle_1 \\ &= \boldsymbol{\alpha}^\top \mathbf{Q} \boldsymbol{\alpha} \end{aligned} \tag{6}$$

where $\mathbf{Q} = [R_1(x_i, x_{i'})]$ evaluates the RK function at all combinations of $i, i' \in \{1, \ldots, n\}$. Note that the first line is due to the fact that $\mathcal{P}_1 f_\lambda = f_{1\lambda}$ for any $f_\lambda \in \mathcal{H}$, the second line is due to the bilinear nature of the inner product, and the third line is due to the reproducing property of the RK function.

2.3. Scalable Computation

The optimal solution given by the representer theorem in Equation (5) uses all training data points to represent $f_{1\lambda}$, which could be computationally costly when n is large. For more scalable computation, it is typical to approximate $f_{1\lambda}$ by evaluating the contrast space RK at all combinations of $r < n$ knots, which are typically placed at the quantiles of the training data predictor scores. Using this type of (low-rank) smoothing spline approximation, the approximation to the representer theorem becomes

$$f_\lambda(x) \approx \sum_{j=0}^{m-1} \beta_j N_j(x) + \sum_{\ell=1}^{r} \alpha_\ell R_1(x, x_\ell^*) \tag{7}$$

where $\{x_\ell^*\}_{\ell=1}^r$ are the chosen knots. As long as enough knots are used in the representation, the approximate representer theorem in Equation (7) can produce theoretically optimal function estimates [22,23]. For optimal asymptotic properties, the number of knots should be on the order of $r \asymp O(n^{2/(4\delta+1)})$, where $\delta \in [1,2]$ depends on the smoothness of the unknown true function. Note that $\delta = 1$ is necessary when $P(f) < \infty$ is barely satisfied, whereas $\delta = 2$ can be used when f is sufficiently smooth (see [5,22,23]).

Using the approximate representer theorem in Equation (7), the penalized least squares functional from Equation (4) becomes a penalized least squares problem of the form

$$\begin{bmatrix} \hat{\boldsymbol{\beta}}_\lambda \\ \hat{\boldsymbol{\alpha}}_\lambda \end{bmatrix} = \underset{\boldsymbol{\beta} \in \mathbb{R}^m, \boldsymbol{\alpha} \in \mathbb{R}^n}{\arg\min} \left[\frac{1}{n} \sum_{i=1}^{n} \left(y_i - \boldsymbol{N}_i^\top \boldsymbol{\beta} - \boldsymbol{R}_i^\top \boldsymbol{\alpha} \right)^2 + \lambda \boldsymbol{\alpha}^\top \mathbf{Q} \boldsymbol{\alpha} \right] \tag{8}$$

where $\boldsymbol{N}_i = (N_0(x_i), \ldots, N_{m-1}(x_i))^\top$ is the i-th observation's null space basis function vector, and $\boldsymbol{R}_i = \left(R_1(x_i, x_1^*), \ldots, R_1(x_i, x_r^*)\right)^\top$ is the i-th observation's contrast space basis function vector. Note that $\mathbf{Q} = [R_1(x_k^*, x_\ell^*)]$ evaluates the contrast space RK at all combinations of knots. Given a choice of the smoothing parameter λ, the solution has the form

$$\begin{bmatrix} \hat{\boldsymbol{\beta}}_\lambda \\ \hat{\boldsymbol{\alpha}}_\lambda \end{bmatrix} = \begin{bmatrix} \mathbf{N}^\top \mathbf{N} & \mathbf{N}^\top \mathbf{R} \\ \mathbf{R}^\top \mathbf{N} & \mathbf{R}^\top \mathbf{R} + n\lambda \mathbf{Q} \end{bmatrix}^\dagger \begin{bmatrix} \mathbf{N}^\top \\ \mathbf{R}^\top \end{bmatrix} \mathbf{y} \tag{9}$$

where $\mathbf{N} = (\boldsymbol{N}_1, \ldots, \boldsymbol{N}_n)^\top$ is the null space design matrix with $\boldsymbol{N}_i$ as rows, $\mathbf{R} = (\boldsymbol{R}_1, \ldots, \boldsymbol{R}_n)^\top$ is the contrast space design matrix with $\boldsymbol{R}_i$ as rows, $\mathbf{y} = (y_1, \ldots, y_n)^\top$ is the response vector, and $(\cdot)^\dagger$ denotes the Moore–Penrose pseudoinverse [24,25].

3. Tensor Product Smoothing

3.1. Marginal Function Space Notation

Now, consider the multiple nonparametric regression model in Equation (1), where $\boldsymbol{X} = (X_1, \ldots, X_p)^\top \in \mathcal{X}$ is the observed predictor vector. Note that $\mathcal{X} = \mathcal{X}_{(1)} \times \cdots \times \mathcal{X}_{(p)}$ is the product domain with $\mathcal{X}_{(j)}$ denoting the domain of X_j. Following the discussion from Section 2.1, let $\mathcal{H}_{(j)}$ denote a RKHS of functions on $\mathcal{X}_{(j)}$ for $j = 1, \ldots, p$. Suppose that the complexity (i.e., lack of smoothness) for each predictor's marginal RKHS is defined according to some non-negative penalty functional P_j. This implies that each RKHS can be decomposed such as $\mathcal{H}_{(j)} = \mathcal{H}_{0(j)} \oplus \mathcal{H}_{1(j)}$, where $\mathcal{H}_{0(j)} = \{f : P_j(f) = 0, f \in \mathcal{H}_{(j)}\}$ is the j-th predictor's null space, which contains all functions (in $\mathcal{H}_{(j)}$) that have zero penalty, and $\mathcal{H}_{1(j)} = \{f : P_j(f) > 0, f \in \mathcal{H}_{(j)}\}$ is the j-th predictor's contrast space, which contains all functions (in $\mathcal{H}_{(j)}$) that have a non-zero penalty. When relevant, the j-th predictor's null space can be further decomposed such as $\mathcal{H}_{0(j)} = \mathcal{H}_{00(j)} \oplus \mathcal{H}_{01(j)}$, where $\mathcal{H}_{00(j)}$ is a constant (intercept) subspace, and $\mathcal{H}_{01(j)}$ contains non-constant functions that are unpenalized.

The inner product of $\mathcal{H}_{(j)}$ will be denoted by $\langle f, g \rangle_{(j)}$ for any $f, g \in \mathcal{H}_{(j)}$, and the corresponding norm will be written as $\|f\|_{(j)} = \sqrt{\langle f, f \rangle_{(j)}}$ for any $f \in \mathcal{H}_{(j)}$. Each inner product can be decomposed into its null and contrast contributions, such as $\langle f, g \rangle_{(j)} = \langle f, g \rangle_{0(j)} + \langle f, g \rangle_{1(j)}$, and the corresponding norms will be denoted by $\|f\|_{0(j)} = \sqrt{\langle f, f \rangle_{0(j)}}$ (norm of $\mathcal{H}_{0(j)}$) and $\|f\|_{1(j)} = \sqrt{\langle f, f \rangle_{1(j)}}$ (norm of $\mathcal{H}_{1(j)}$). When $\mathcal{H}_{0(j)} = \mathcal{H}_{00(j)} \oplus \mathcal{H}_{01(j)}$, the null space inner product can be written as $\langle f, g \rangle_{0(j)} = \langle f, g \rangle_{00(j)} + \langle f, g \rangle_{01(j)}$, where $\langle f, g \rangle_{00(j)}$ is the inner product of $\mathcal{H}_{00(j)}$ for any $f, g \in \mathcal{H}_{00(j)}$, and $\langle f, g \rangle_{01(j)}$ is the inner product of $\mathcal{H}_{01(j)}$ for any $f, g \in \mathcal{H}_{01(j)}$. The corresponding norm can be written as $\|f\|_{0(j)} = \sqrt{\|f\|^2_{00(j)} + \|f\|^2_{01(j)}}$, where $\|f\|_{00(j)} = \sqrt{\langle f, f\rangle_{00(j)}}$ and $\|f\|_{01(j)} = \sqrt{\langle f, f\rangle_{01(j)}}$ denote the norms of $\mathcal{H}_{00(j)}$ and $\mathcal{H}_{01(j)}$, respectively.

The RK of $\mathcal{H}_{(j)}$ will be denoted by $R_{(j)}(x, z)$ for any $x, z \in \mathcal{X}_{(j)}$, and note that the RK is an element of the j-th predictor's RKHS, i.e., $R_{(j)} \in \mathcal{H}_{(j)}$ for any $x, z \in \mathcal{X}_{(j)}$. By definition, the RK is the representer of the evaluation functional in $\mathcal{H}_{(j)}$, which implies that the RK satisfies $f(x) = \langle R_{(j)}(x, z), f(z)\rangle$ for any $f \in \mathcal{H}_{(j)}$ and any $x, z \in \mathcal{X}_{(j)}$. Note that $R_{(j)}(x, z) = R_{0(j)}(x, z) + R_{1(j)}(x, z)$, where where $R_{0(j)} \in \mathcal{H}_{0(j)}$ is the null space RK and $R_{1(j)} \in \mathcal{H}_{1(j)}$ is the contrast space RK. Furthermore, when $\mathcal{H}_{0(j)} = \mathcal{H}_{00(j)} \oplus \mathcal{H}_{01(j)}$, the null space RK can be decomposed such as $R_{0(j)}(x, z) = R_{00(j)}(x, z) + R_{01(j)}(x, z)$, where $R_{00(j)} \in \mathcal{H}_{00(j)}$ and $R_{01(j)} \in \mathcal{H}_{01(j)}$ denotes the RKs of $\mathcal{H}_{00(j)}$ and $\mathcal{H}_{01(j)}$, respectively. Note that $R_{00(j)}(x, z) = \beta_{0(j)}$ for all $x, z \in \mathcal{X}_{(j)}$, where $\beta_{0(j)} \in \mathbb{R}$ is some constant, given that $\mathcal{H}_{00(j)}$ is assumed to be a constant (intercept) subspace for all p predictors.

3.2. Tensor Product Function Spaces

Consider the construction of a tensor product function space $\mathcal{H}$ that is formed by combining the marginal spaces $\{\mathcal{H}_{(1)}, \dots, \mathcal{H}_{(p)}\}$. The largest space that could be constructed includes all possible main and interaction effects, such as

$$\begin{aligned} \mathcal{H} &= \mathcal{H}_{(1)} \otimes \cdots \otimes \mathcal{H}_{(p)} \\ &= \mathcal{H}_{\{0\}} \oplus \mathcal{H}_{\{1\}} \oplus \cdots \oplus \mathcal{H}_{\{p\}} \end{aligned} \tag{10}$$

where $\mathcal{H}_{\{0\}} = \{f : f(\boldsymbol{X}) \propto 1\ \forall \boldsymbol{X} \in \mathcal{X}\}$ is the tensor product constant (intercept) space, and each $\mathcal{H}_{\{j\}}$ consists of $\binom{p}{j}$ orthogonal subspaces that capture different main and/or interaction effects of the predictors. For example, $\mathcal{H}_{\{1\}} = \oplus_{j=1}^{p} \mathcal{H}_{(j)}$ consists of p main effect subspaces, $\mathcal{H}_{\{2\}} = \oplus_{k=2}^{p} \oplus_{j=1}^{k-1} \mathcal{H}_{(j)} \otimes \mathcal{H}_{(k)}$ consists of $\binom{p}{2} = \frac{p(p-1)}{2}$ two-way interaction effect subspaces, etc. Note that different (more parsimonious) statistical models can be formed by excluding subspaces from the tensor product RKHS defined in Equation (10). For example, the tensor product space corresponding to the additive model has the form $\mathcal{H} = \mathcal{H}_{\{0\}} \oplus \mathcal{H}_{\{1\}}$. For the model that includes all main effects and two-way interactions, the tensor product RKHS has the form $\mathcal{H} = \mathcal{H}_{\{0\}} \oplus \mathcal{H}_{\{1\}} \oplus \mathcal{H}_{\{2\}}$.

Let $\boldsymbol{X} = (X_1, \dots, X_p)^\top \in \mathcal{X}$ and $\boldsymbol{Z} = (Z_1, \dots, Z_p)^\top \in \mathcal{X}$ denote two arbitrary predictor vectors. To evaluate functions in $\mathcal{H}$, the tensor product RK can be defined as

$$\begin{aligned} R(\boldsymbol{X}, \boldsymbol{Z}) &= \prod_{j=1}^{p} R_{(j)}(X_j, Z_j) \\ &= R_{\{0\}}(\boldsymbol{X}, \boldsymbol{Z}) + R_{\{1\}}(\boldsymbol{X}, \boldsymbol{Z}) + \cdots + R_{\{p\}}(\boldsymbol{X}, \boldsymbol{Z}) \end{aligned} \tag{11}$$

where $R_{\{0\}}(\boldsymbol{X}, \boldsymbol{Z}) = 1$ is the constant (intercept) term, and each $R_{\{j\}}$ consists of a summation of $\binom{p}{j}$ RKs from orthogonal subspaces that capture different main and/or interaction effects of the predictors. For example, $R_{\{1\}} = \sum_{j=1}^{p} R_{(j)}$ consists of p main effect RKs, $R_{\{2\}} = \sum_{k=2}^{p} \sum_{j=1}^{k-1} R_{(j)} R_{(k)}$ consists of $\binom{p}{2} = \frac{p(p-1)}{2}$ two-way interaction effect RKs, etc. When different (more parsimonious) models are formed by excluding subspaces of the tensor product RKHS, the corresponding components of the tensor product RK are also excluded. For example, the tensor product RK corresponding to the additive model has the form $R = R_{\{0\}} + R_{\{1\}}$, and the tensor product RK for the model that includes all main effects and two-way interactions has the form $R = R_{\{0\}} + R_{\{1\}} + R_{\{2\}}$.

The inner product of the tensor product RKHS $\mathcal{H}$ can be written as

$$\langle f, g\rangle = \langle f, g\rangle_{\{0\}} + \langle f, g\rangle_{\{1\}} + \cdots + \langle f, g\rangle_{\{p\}} \tag{12}$$

where $\langle f,g\rangle_{\{0\}}$ is the inner product of $\mathcal{H}_{\{0\}}$, and $\langle f,g\rangle_{\{j\}}$ consists of a summation of $\binom{p}{j}$ inner products corresponding to orthogonal subspaces that capture different main and/or interaction effects of the predictors. For example, $\langle f,g\rangle_{\{1\}}$ consists of the summation of p main effect inner products, and $\langle f,g\rangle_{\{2\}}$ consists of the summation of $\frac{p(p-1)}{2}$ two-way interaction effect inner products. The specifics of each subspace's inner product will depend on the type of spline used for each predictor. This is because each subspace's inner product (and, consequently, penalty) aggregates information across the penalized components after "averaging out" information from unpenalized components (see [5] (pp. 40–48)).

3.3. Representation and Computation

Given an assumed model form, the tensor product RKHS can be written as

$$\mathcal{H} = \mathcal{H}_0^\star \oplus \mathcal{H}_1 \oplus \cdots \oplus \mathcal{H}_K \tag{13}$$

where $\mathcal{H}_0^\star = \{f : P(f) = 0, f \in \mathcal{H}\}$ is the tensor product null space with $P(\cdot)$ denoting the tensor product penalty (later defined), and $\mathcal{H}_k$ is the k-th orthogonal subspace of the tensor product contrast space $\mathcal{H}_1^\star = \mathcal{H}_1 \oplus \cdots \oplus \mathcal{H}_K$. Note that $\mathcal{H}_k$ corresponds to the different main and/or interaction effect subspaces that are included in the assumed model form. The corresponding inner product and RK can be written as

$$\begin{aligned}\langle f,g\rangle &= \langle f,g\rangle_0^\star + \sum_{k=1}^{K} \theta_k^{-1}\langle f,g\rangle_k \\ R(\boldsymbol{X},\boldsymbol{Z}) &= R_0^\star(\boldsymbol{X},\boldsymbol{Z}) + \sum_{k=1}^{K} \theta_k R_k(\boldsymbol{X},\boldsymbol{Z})\end{aligned} \tag{14}$$

where $\langle f,g\rangle_0^\star$ and $R_0^\star(\boldsymbol{X},\boldsymbol{Z})$ denote the inner product and RK of $\mathcal{H}_0^\star$ (the tensor product null space), $\langle f,g\rangle_k$ and $R_k(\boldsymbol{X},\boldsymbol{Z})$ denote the inner product and RK of $\mathcal{H}_k$ for $k = 1,\ldots,K$, and the $\theta_k > 0$ are additional non-negative tuning parameters that control the influence of each subspace's contribution. Note that including the θ_k parameters is essential given that the different subspaces do not necessarily have comparable metrics.

Suppose that the tensor product penalty $P(f)$ is defined to be the squared norm of the function's projection into the (tensor product) contrast space, i.e.,

$$P(f) = \langle f,f\rangle_1^\star = \sum_{k=1}^{K} \theta_k^{-1}\|f\|_k^2 \tag{15}$$

where $\|f\|_k = \sqrt{\langle f,f\rangle_k}$ is the norm for $\mathcal{H}_k$ (the k-th orthogonal subspace of $\mathcal{H}_1^\star$). Using this definition of the penalty, the function minimizing the penalized least squares functional in Equation (2) can be written according to the representer theorem in Equation (5). In this case, the set of functions $\{N_0, N_1, \ldots, N_{m-1}\}$ forms a basis for the tensor product null space $\mathcal{H}_0^\star$, and the RK of the contrast space is defined as $R_1^\star = \sum_{k=1}^K \theta_k R_k$. Using this optimal representation, the penalty can be written according to Equation (6) with the penalty matrix defined as $\mathbf{Q} = \sum_{k=1}^K \theta_k \mathbf{Q}_k$ where $\mathbf{Q}_k = [R_k(\mathbf{x}_i, \mathbf{x}_{i'})]$ evaluates the k-th subspace's RK function at all combinations of training data points.

For scalable computation as n becomes large, the approximate representer theorem in Equation (7) can be applied using the knots $\{\mathbf{x}_\ell^*\}_{\ell=1}^r$, where $\mathbf{x}_\ell^* = (x_{\ell 1}^*, \ldots, x_{\ell p}^*)^\top \in \mathcal{X}$ for all $\ell = 1,\ldots,r$. Using the approximately optimal representation from Equation (7), the penalized least squares problem can be written according to Equation (8), and the optimal coefficients can be written according to Equation (9). In the tensor product case, the optimal coefficients should really be subscripted with $\boldsymbol{\lambda} = (\lambda, \theta_1, \ldots, \theta_K)$, given that these estimates depend on the overall tuning parameter λ, as well as the K tuning (hyper)parameters for each of the contrast subspaces. Note that the penalty only depends on $(\lambda_1, \ldots, \lambda_K)$ where $\lambda_k = \lambda/\theta_k$ for $k = 1,\ldots,K$. However, it is often helpful (for tuning purposes) to separate the overall tuning parameter λ from the tuning parameters that control the individual effect functions, i.e., the θ_k tuning parameters.

4. Refined Tensor Product Smoothing

4.1. Smoothing Spline Like Estimators

Consider a single predictor (i.e., $p = 1$) that satisfies $x \in \mathcal{X}$, and let $\mathcal{H}$ denote a RKHS of functions on $\mathcal{X}$. Consider a decomposition of the function space such as $\mathcal{H} = \mathcal{H}_{00} \oplus \mathcal{H}_{11}$, where $\mathcal{H}_{11} = \mathcal{H}_{01} \oplus \mathcal{H}_1$ is a space of non-constant functions that either sum to zero (for categorical x) or integrate to zero (for continuous x) across the domain $\mathcal{X}$. The inner product of $\mathcal{H}$ can be written as $\langle f, g \rangle = \langle f, g \rangle_{00} + \langle f, g \rangle_{11}$ for any $f, g \in \mathcal{H}$, where $\langle f, g \rangle_{11} = \langle f, g \rangle_{01} + \langle f, g \rangle_1$ is the inner product of $\mathcal{H}_{11}$. The corresponding RK can be written as $R(x,z) = R_{00}(x,z) + R_{11}(x,z)$ for any $x, z \in \mathcal{X}$, where $R_{11} = R_{01} + R_1$ is the RK for $\mathcal{H}_{11}$. Given a sample of n observations $\{(x_i, y_i)\}_{i=1}^n$, consider finding the $f \in \mathcal{H}$ that satisfies

$$\hat{f}_\lambda = \underset{f \in \mathcal{H}}{\arg\min} \left\{ \frac{1}{n} \sum_{i=1}^n (y_i - f(x_i))^2 + \lambda \|f\|_{11}^2 \right\} \tag{16}$$

where $\|f\|_{11}^2 = \langle f, f \rangle_{11}$ is the squared norm of the projection of f into $\mathcal{H}_{11}$. The $\hat{f}_\lambda$ defined in Equation (16) is a smoothing spline if $\mathcal{H}_0 = \mathcal{H}_{00}$, which will be the case for nominal, ordinal, and linear smoothing splines. However, for cubic (and higher-order) smoothing splines, the $\mathcal{H}_{01}$ subspace consists of non-constant lower-order polynomial terms, which are unpenalized. Note that the $\hat{f}_\lambda$ in Equation (16) penalizes all non-constant terms, so it will not be equivalent to a cubic smoothing spline—even when $\mathcal{H} = \{f : \int |f^2(x)|^2 dx < \infty, \forall x \in \mathcal{X}\}$ is the same RKHS used for cubic smoothing spline estimation.

Theorem 1 (Representer Theorem)**.** *The $f \in \mathcal{H}$ that minimizes Equation (16) has the form*

$$f_\lambda(x) = \beta + \sum_{i=1}^n \alpha_i R_{11}(x, x_i)$$

where $\beta \in \mathbb{R}$ is an intercept parameter and $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top \in \mathbb{R}^n$ is a vector of coefficients that combine the reproducing kernel function evaluations.

Proof. The theorem is simply a version of the representer theorem from Equation (5) where the null space has dimension one. □

Corollary 1 (Low-Rank Approximation)**.** *The function $f \in \mathcal{H}$ that minimizes Equation (16) can be well-approximated via*

$$f_\lambda(x) \approx \beta + \sum_{\ell=1}^r \alpha_\ell^* R_{11}(x, x_\ell^*)$$

where $\{x_\ell^*\}_{\ell=1}^r$ are the selected knots with $r \asymp O(n^{2/(4\delta+1)})$ for some $\delta \in [1,2]$.

Proof. The corollary is simply a version of the approximate representer theorem from Equation (7) where the null space has dimension one. □

These results imply that the penalized least squares functional from Equation (16) can be rewritten as the penalized least squares problem in Equation (8) where (i) the null space only contains the intercept column, i.e., $\boldsymbol{N}_i = 1$ and $\boldsymbol{\beta} = \beta$, and (ii) the contrast space RK R_1 is replaced by R_{11} in the function and penalty representation, i.e., $\boldsymbol{R}_i = \left(R_{11}(x, x_1^*), \ldots, R_{11}(x, x_r^*)\right)^\top$ and $\mathbf{Q} = [R_{11}(x_\ell^*, x_{\ell'}^*)]$. Using these modifications the optimal coefficients can be written according to Equation (9).

4.2. Tensor Product Formulation

Now, consider the model in Equation (1) with $p \geq 2$ predictors. Given an assumed model form, the tensor product RKHS $\mathcal{H}$ can be written according to the tensor sum decomposition in Equation (13) with $\mathcal{H}_0^\star = \{f : f(\boldsymbol{X}) \propto 1 \; \forall \boldsymbol{X} \in \mathcal{X}\}$ denoting the constant (intercept) subspace. Similarly, the inner product and RK of $\mathcal{H}$ can be written according to Equation (14), and the tensor product penalty can be written according to Equation (15). Unlike the previous tensor product treatment, this tensor product formulation assumes that $\mathcal{H}_0^\star$ contains only the constant (intercept) subspace, which implies that the $\mathcal{H}_k$ subspaces contain all non-constant functions of the predictors. Furthermore, this implies that the proposed formulation of the tensor product penalty in Equation (15) penalizes all non-constant functions of the predictors. Note that if all p predictors have a null space dimension of one, i.e., if $\mathcal{H}_{0(j)} = \mathcal{H}_{00(j)}$ for all $j = 1, \ldots, p$, then the proposed formulation will be equivalent to the classic formulation. However, if $\mathcal{H}_{01(j)}$ exists for any predictor, then the proposed formulation will differ from the classic formulation because the functions in $\mathcal{H}_{01(j)}$ will be penalized using the proposed formulation.

Given a sample of n observations $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ with $\mathbf{x}_i = (x_{i1}, \ldots, x_{ip}) \in \mathcal{X}$ and $y_i \in \mathbb{R}$, consider the problem of finding the function $f \in \mathcal{H}$ that satisfies

$$\hat{f}_\lambda = \underset{f \in \mathcal{H}}{\arg\min} \left\{ \frac{1}{n} \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2 + \lambda \sum_{k=1}^K \omega_k \|f\|_k^2 \right\} \tag{17}$$

where the $\omega_k \geq 0$ are additional tuning parameters (penalty weights) that control the influence of each component function's penalty contribution.

Theorem 2 (Tensor Product Representer Theorem)**.** *The minimizer of Equation (17) has the form $f_\lambda = \sum_{k=0}^K f_{k\lambda}$, where $f_{0\lambda} \in \mathbb{R}$ is an intercept, and $f_{k\lambda} \in \mathcal{H}_k$ is the k-th effect function for $k = 1, \ldots, K$. The optimal effect functions can be expressed as*

$$f_{k\lambda}(\boldsymbol{x}) = \sum_{i=1}^n \alpha_{ik} R_k(\boldsymbol{x}, \mathbf{x}_i)$$

for all $\boldsymbol{x} \in \mathcal{X}$, where the coefficient vector $\boldsymbol{\alpha}_k = (\alpha_{1k}, \ldots, \alpha_{nk})^\top \in \mathbb{R}^n$ depends on the chosen hyperparameters (i.e., λ and ω_k) for $k = 1, \ldots, K$.

Proof. The result in Theorem 2 can be considered a generalization of the typical result used in tensor product smoothing spline estimators (see [3,5]). More specifically, the SSANOVA approach assumes that the function can be represented according to the form in Theorem 2 with the coefficients defined as $\alpha_{ik} = \alpha_i \theta_k$, where the vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^\top$ is common to all K terms. □

Compared to the tensor product representation used in the SSANOVA modeling approach, the proposed approach combines the marginal RK information in a more flexible manner, such as

$$\text{SSANOVA}: \; f_\lambda(\boldsymbol{x}) = f_{0\lambda} + \sum_{i=1}^n \sum_{k=1}^K \alpha_i \theta_k R_k(\boldsymbol{x}, \mathbf{x}_i)$$

$$\text{Proposed}: \; f_\lambda(\boldsymbol{x}) = f_{0\lambda} + \sum_{i=1}^n \sum_{k=1}^K \alpha_{ik} R_k(\boldsymbol{x}, \mathbf{x}_i)$$

Clearly, the two representations are equivalent when $\alpha_{ik} = \alpha_i \theta_k$ for all $i \in \{1, \ldots, n\}$ and all $k \in \{1, \ldots, K\}$. However, such a constraint is not necessary in practice. At first glance, it may appear that the proposed approach has made the estimation problem more challenging, given that the number of parameters has increased from $n + K$ to nK. However, for estimation and inference purposes, it is beneficial to allow each term to have unique coefficients, given that this makes is possible to treat the tuning parameters as variance components in a linear mixed effects modeling framework [2,26,27].

4.3. Scalable Computation

The tensor product representer theorem in Theorem 2 is computationally costly for large n and/or K, given that it requires estimation of nK coefficients. For more practical computation, it is possible to apply knot-based approximations in a tensor product function space, as described in the following corollary.

Corollary 2 (Tensor Product Low-Rank Approximation)**.** *The minimizer of Equation (17) has the form $f_\lambda = \sum_{k=0}^{K} f_{k\lambda}$, and the effect functions can be approximated via*

$$f_{k\lambda}(\boldsymbol{x}) \approx \sum_{\ell=1}^{r_k} \alpha_{\ell k} R_k(\boldsymbol{x}, \mathbf{x}_{\ell k}^*)$$

where $\{\mathbf{x}_{\ell k}^*\}_{\ell=1}^{r_k}$ are the selected knots for the k-th effect with $\mathbf{x}_{\ell k}^* \in \mathcal{X}_k \subset \mathcal{X}\ \forall \ell, k$.

The proposed representation also allows for more flexible knot placement within each of the K subspaces of the tensor product contrast space. In particular, each of the K contrast subspaces is permitted to have a different number of knots r_k using this formulation. Furthermore, note that $\mathbf{x}_{\ell k}^*$ only needs to contain knot values for the predictors that are included in the k-th effect, e.g., $\mathbf{x}_{\ell k}^*$ is a scalar for main effects, a vector of length two for two-way interactions, etc. For main effects, it is typical to place the knots at the (univariate) data quantiles for each predictor. For two-way interactions, many different knot placement strategies are possible, e.g., fixed grid, random sample, bivariate quantiles, strategic placement, etc. In this paper, I only consider multivariate knot placements that involve taking combinations of univariate knots [as in 20], but my ideas are easily applicable to other knot placement schemes.

Theorem 3 (Tensor Product Penalties)**.** *Suppose that $K \geq p$ and $\mathcal{H}_k$ captures the k-th predictor's main effect for $k = 1, \ldots, p$. Given any $\boldsymbol{x} = (x_1, \ldots, x_p) \in \mathcal{X}$, the k-th basis vector is defined as $\boldsymbol{R}_k = (R_{11(k)}(x_k, x_{1k}^*), \ldots, R_{11(k)}(x_k, x_{r_k k}^*))^\top$ and the k-th penalty matrix is $\mathbf{Q}_k = [R_{11(k)}(x_{\ell k}^*, x_{\ell' k}^*)]$, where $R_{11(k)} = R_{01(k)} + R_{1(k)}$ is the non-constant portion of each predictor's marginal RK function for $k = 1, \ldots, p$. Now, suppose that $\mathcal{H}_k$ (for some $k > p$) captures the interaction effect between X_a and X_b for some $a, b \in \{1, \ldots, p\}$. If the basis vector is defined as $\boldsymbol{R}_k = \boldsymbol{R}_a \tilde{\otimes} \boldsymbol{R}_b$, where $\tilde{\otimes}$ denotes the Kronecker product, then the penalty matrix has the form $\mathbf{Q}_k = \mathbf{Q}_a \tilde{\otimes} \mathbf{Q}_b$. Now, suppose that $\mathcal{H}_k$ (for some $k > p$) captures the three-way interaction between (X_a, X_b, X_c) for some $a, b, c \in \{1, \ldots, p\}$. If the basis vector is defined as $\boldsymbol{R}_k = \boldsymbol{R}_a \tilde{\otimes} \boldsymbol{R}_b \tilde{\otimes} \boldsymbol{R}_c$, then the penalty matrix has the form $\mathbf{Q}_k = \mathbf{Q}_a \tilde{\otimes} \mathbf{Q}_b \tilde{\otimes} \mathbf{Q}_c$. Basis vectors and penalty matrices for higher-order interactions can be efficiently constructed in a similar fashion.*

Proof. To prove the theorem, it suffices to prove the result for two-way interactions, given that three-way (and higher-order) interactions can be built by recursively applying the results from the two-way interaction scenario. Specifically, it suffices to show that $\mathbf{Q}_k = \mathbf{Q}_a \tilde{\otimes} \mathbf{Q}_b$ is the penalty matrix corresponding to $\boldsymbol{R}_k = \boldsymbol{R}_a \tilde{\otimes} \boldsymbol{R}_b$. First note that the vector $\boldsymbol{R}_k = \boldsymbol{R}_a \tilde{\otimes} \boldsymbol{R}_b = (R_{1k}, \ldots, R_{r_k k})^\top$ has length $r_k = r_a r_b$ for any $a, b \in \{1, \ldots, p\}$. The ℓ-th entry $R_{\ell k}$ can be written in terms of the corresponding entries of $\boldsymbol{R}_a$ and $\boldsymbol{R}_b$, such as

$$R_{\ell k} = R_k(\boldsymbol{x}, \mathbf{x}_{\ell k}^*) = R_{11(a)}(x_a, x_{ua}^*) R_{11(b)}(x_b, x_{vb}^*) \tag{18}$$

where $\boldsymbol{x} = (x_a, x_b)$ is the bivariate vector at which the RK is evaluated, and $\mathbf{x}_{\ell k}^* = (x_{ua}^*, x_{vb}^*)$ is the bivariate knot. Note that $\ell = v + r_b(u - 1)$ indexes the tensor product vector $\boldsymbol{R}_k$, and $u \in \{1, \ldots, r_a\}$ and $v \in \{1, \ldots, r_b\}$ index the marginal $\boldsymbol{R}_a$ and $\boldsymbol{R}_b$ vectors. Letting $\boldsymbol{\alpha}_k = (\alpha_{1k}, \ldots, \alpha_{r_k k})^\top \in \mathbb{R}^{r_k}$ denote an arbitrary coefficient vector, the penalty for the k-th term has the form

$$\begin{aligned} \|f\|_k^2 &= \textstyle\sum_{\ell=1}^{r_k} \sum_{\ell'=1}^{r_k} \alpha_{\ell k} \alpha_{\ell' k} \langle R_{\ell k}, R_{\ell' k} \rangle_k \\ &= \textstyle\sum_{\ell=1}^{r_k} \sum_{\ell'=1}^{r_k} \alpha_{\ell k} \alpha_{\ell' k} R_k(\mathbf{x}_{\ell k}^*, \mathbf{x}_{\ell' k}^*) \\ &= \boldsymbol{\alpha}_k^\top (\mathbf{Q}_a \tilde{\otimes} \mathbf{Q}_b) \boldsymbol{\alpha}_k \end{aligned} \tag{19}$$

where the first line is due to the bilinearity of the inner product, the second line is due to the reproducing property of the RK function, and the third line is a straightforward (algebraic) simplification of the second line. □

5. Tensor Product Spectral Smoothing

5.1. Spectral Representater Theorem

For a more convenient representation of univariate smoothing spline (like) estimators, I introduce the spectral version of the representer theorem from Theorem 1, which will be particularly useful for tensor product function building.

Theorem 4 (Spectral Representer Theorem)**.** *Let $\boldsymbol{R} = (R_{11}(x, x_1), \dots, R_{11}(x, x_n))^\top$ denote the vector of RK evaluations at the training data for an arbitrary $x \in \mathcal{X}$, and let $\mathbf{Q} = [R_{11}(x_i, x_{i'})]$ denote the corresponding penalty matrix. Consider an eigen-decomposition of $\mathbf{Q}$ of the form $\mathbf{Q} = \mathbf{V}\mathbf{D}^2\mathbf{V}^\top$, where $\mathbf{V} = (\mathbf{v}_1, \dots, \mathbf{v}_n)$ is the matrix of eigenvectors, and $\mathbf{D}^2 = \mathrm{diag}(d_1^2, \dots, d_n^2)$ is the diagonal matrix of eigenvalues ($d_i > 0$ is the i-th singular value). The function $f \in \mathcal{H}$ that minimizes Equation (16) can be written as*

$$f_\lambda(x) = \beta + \sum_{i=1}^{n} \gamma_i S_i(x)$$

where $\boldsymbol{\gamma} = (\gamma_1, \dots, \gamma_n)^\top \in \mathbb{R}^n$ is a vector of coefficients, and $S_i(x) = d_i^{-1}\mathbf{v}_i^\top \boldsymbol{R}$. The spectral basis functions satisfy $\langle S_i, S_{i'} \rangle_{11} = \delta_{ii'}$, where $\delta_{ii'}$ is Kronecker's delta, which implies that $\|\boldsymbol{S}^\top \boldsymbol{\gamma}\|_{11}^2 = \sum_{i=1}^n \gamma_i^2$ for any $\boldsymbol{\gamma} \in \mathbb{R}^n$, where $\boldsymbol{S} = (S_1(x), \dots, S_n(x))^\top$.

Proof. To prove the first part of the theorem, we need to prove that $\boldsymbol{R}^\top \boldsymbol{\alpha} = \boldsymbol{S}^\top \boldsymbol{\gamma}$, where $\boldsymbol{S} = (S_1(x), \dots, S_n(x))^\top$. To establish the connection between the classic and spectral representations, first note that we can write the transformed (spectral) basis as $\boldsymbol{S} = \mathbf{D}^{-1}\mathbf{V}^\top \boldsymbol{R}$, and the corresponding transformed coefficients as $\boldsymbol{\gamma} = \mathbf{D}\mathbf{V}^\top \boldsymbol{\alpha}$. This implies that

$$\boldsymbol{S}^\top \boldsymbol{\gamma} = \boldsymbol{R}^\top \mathbf{V}\mathbf{D}^{-1}\mathbf{D}\mathbf{V}^\top \boldsymbol{\alpha} = \boldsymbol{R}^\top \boldsymbol{\alpha}$$

given that $\mathbf{V}\mathbf{D}^{-1}\mathbf{D}\mathbf{V}^\top = \mathbf{I}_n$, which completes the proof of the first part of the theorem. The prove the second part of the theorem, note that $\langle S_i, S_{i'} \rangle_1 = d_i^{-1} d_{i'}^{-1} \mathbf{v}_i^\top \mathbf{V}\mathbf{D}^2\mathbf{V}^\top \mathbf{v}_{i'} = \delta_{ii'}$, which is a consequence of the fact that $\langle \boldsymbol{R}, \boldsymbol{R} \rangle_{11} = \mathbf{Q}$ due to the reproducing property, and the fact that $\mathbf{v}_i^\top \mathbf{v}_{i'} = \delta_{ii'}$ due to the orthonormality of the eigenvectors. □

Note that Theorem 4 reveals that modified representation in Theorem 1 can be equivalently expressed in terms of the empirical eigen-decomposition of the penalty matrix, which we refer to as the *spectral representation* of the smoothing spline. Furthermore, note that the theorem reveals that the spectral basis functions $\{S_1, S_2, \dots, S_n\}$ serve as empirical eigenfunctions for $\mathcal{H}$, in the sense that these functions are a sample dependent basis that is orthonormal with respect to the contrast space inner-product. These eigenfunctions have the typical sign-changing behavior that is characteristic of spectral representations, such that S_{i+1} has more sign changes than S_i for $i = 1, \dots, n$, see Figure 1. Note that the (scaled) ordinal and linear smoothing spline spectra are nearly identical to one another, which is not surprising given the asymptotic equivalence of these kernel functions [28]. Furthermore, note that the (scaled) cubic and quintic smoothing spline spectra are rather similar in appearance, especially for the first four empirical eigenfunctions.

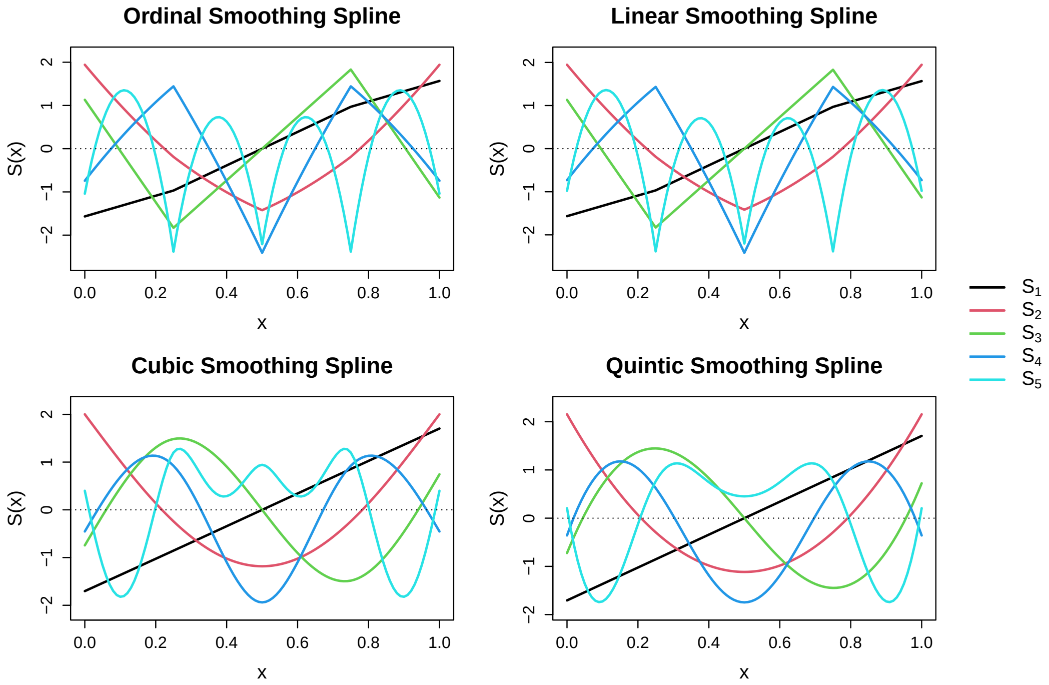

Figure 1. Spectral basis functions for different types of reproducing kernel functions using five equidistant knots. Basis functions were evaluated at $x_i = \frac{i-1}{100}$ for $i = 1, \ldots, 101$ and were scaled for visualization purposes. Produced by R [29] using the `rk()` function in the **grpnet** package [30].

5.2. Tensor Product Formulation

For a more convenient representation of tensor product smoothing spline (like) estimators, I introduce the spectral version of the representer theorem from Theorem 2, which will be particularly useful for tensor product function building.

Theorem 5 (Spectral Tensor Product Representer Theorem)**.** *The minimizer of Equation (17) has the form $f_\lambda = \sum_{k=0}^{K} f_{k\lambda}$, where $f_{0\lambda} \in \mathbb{R}$ is an intercept, and $f_{k\lambda} \in \mathcal{H}_k$ is the k-th effect function for $k = 1, \ldots, K$. The optimal effect functions can be expressed as*

$$f_{k\lambda}(\boldsymbol{x}) = \sum_{i=1}^{n} \gamma_{ik} S_{ik}(\boldsymbol{x})$$

for all $\boldsymbol{x} \in \mathcal{X}$, where $\gamma_k = (\gamma_{1k}, \ldots, \gamma_{nk})^\top \in \mathbb{R}^n$ is the coefficient vector and $\{S_{ik}\}_{i=1}^{n}$ are the spectral basis functions for $k = 1, \ldots, K$. The spectral basis functions can be defined to satisfy $\langle S_{ik}, S_{i'k} \rangle_k = \delta_{ii'}$, where $\delta_{ii'}$ is Kronecker's delta, which implies that $\|S_k \gamma_k\|_k^2 = \sum_{i=1}^{n} \gamma_{ik}^2$ for any $\gamma_k \in \mathbb{R}^n$, where $\boldsymbol{S}_k = (S_{1k}(\boldsymbol{x}), \ldots, S_{nk}(\boldsymbol{x}))^\top$.

Proof. The result in Theorem 5 is essentially a combination of the results in Theorem 2 and Theorem 4. To prove the result, let $\boldsymbol{R}_k = (R_k(\boldsymbol{x}, \mathbf{x}_1), \ldots, R_k(\boldsymbol{x}, \mathbf{x}_n))^\top$ denote the vector of RK evaluations at the training data for an arbitrary $\boldsymbol{x} \in \mathcal{X}$, and let $\mathbf{Q}_k = [R_k(\mathbf{x}_i, \mathbf{x}_{i'})]$ denote the corresponding penalty matrix. Furthermore, let $\mathbf{Q}_k = \mathbf{V}_k \mathbf{D}_k^2 \mathbf{V}_k^\top$ denote the eigen-decomposition of the penalty matrix, where $\mathbf{V}_k = (\mathbf{v}_{1k}, \ldots, \mathbf{v}_{nk})$ is the matrix of eigenvectors, and $\mathbf{D}_k^2 = \mathrm{diag}(d_{1k}^2, \ldots, d_{nk}^2)$ is the diagonal matrix of eigenvalues ($d_{ik} > 0$ is the i-th singular value). Then the spectral basis functions can be defined as $S_{ik} = d_{ik}^{-1} \mathbf{v}_{ik}^\top \boldsymbol{R}_k$, which ensures that $\|\boldsymbol{S}_k \gamma_k\|_k^2 = \sum_{i=1}^{n} \gamma_{ik}^2$ for any $\gamma_k \in \mathbb{R}^n$. □

Using the spectral tensor products, multiple and generalized nonparametric regression models can be easily fit using standard mixed effects modeling software, such as **lme4** [31]. See Figure 2 for a visualization of the spectral tensor product basis functions.

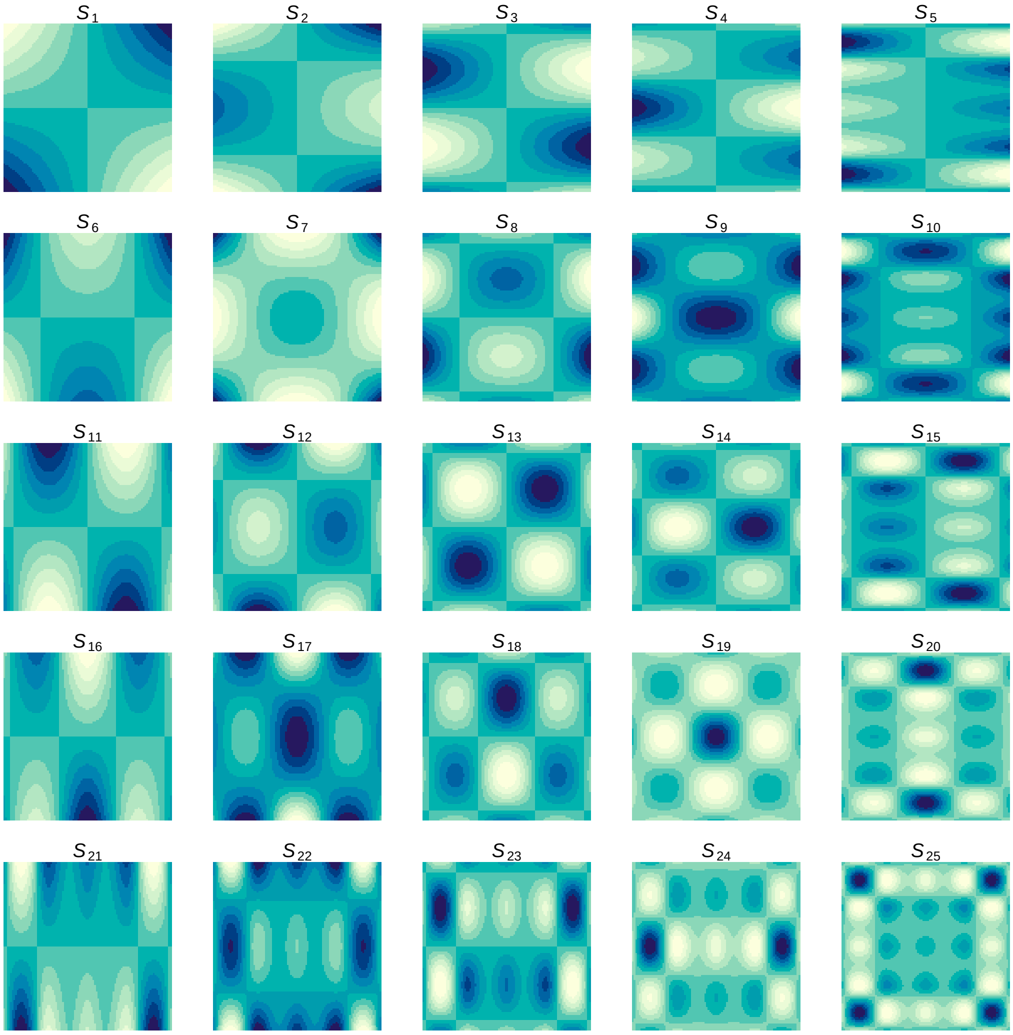

Figure 2. Spectral tensor product basis functions formed from $p = 2$ cubic smoothing spline marginals with $r_k = 5$ equidistant knots for each predictor. From left to right, the basis functions become less smooth with respect to X_2. From top to bottom, the basis functions become less smooth with respect to X_1. Produced by R [29] using the `rk()` function in the **grpnet** package [30].

5.3. Scalable Computation

For large n, the spectral basis functions defined in Theorem 5 are not computationally feasible, given that computing the eigen-decomposition of the penalty requires $O(n^3)$ flops. For more scalable computation, I present a spectral version of Corollary 2.

Corollary 3 (Spectral Tensor Product Low-Rank Approximation)**.** *The minimizer of Equation (17) has the form $f_\lambda = \sum_{k=0}^{K} f_{k\lambda}$, and the effect functions can be approximated via*

$$f_{k\lambda}(\boldsymbol{x}) \approx \sum_{\ell=1}^{r_k} \gamma_{\ell k} S_{\ell k}(\boldsymbol{x})$$

where $\mathbf{S}_k = \left(S_{1k}(\boldsymbol{x}), \ldots, S_{r_k k}(\boldsymbol{x})\right)^\top$ is the vector of spectral basis functions corresponding to $\{\mathbf{x}_{\ell k}^*\}_{\ell=1}^{r_k}$, which are the selected knots for the k-th effect with $\mathbf{x}_{\ell k}^* \in \mathcal{X}_k \subset \mathcal{X}\ \forall \ell, k$.

Using the low-rank approximation, the penalty matrix $\mathbf{Q}_k = [R_k(\mathbf{x}_{\ell k}^*, \mathbf{x}_{\ell' k}^*)]$ evaluates the RK function at all combinations of the selected knots $\{\mathbf{x}_{\ell k}^*\}_{\ell=1}^{r_k}$. Note that the eigen-decomposition of $\mathbf{Q}_k$ only requires $O(nr_k^2)$ flops, which is a substantial improvement if

$r_k \ll n$. For the main effects, the spectral basis functions can be defined as $S_{\ell k} = d_{\ell k}^{-1} \mathbf{v}_{\ell k}^\top \boldsymbol{R}_k$, where $\mathbf{Q}_k = \mathbf{V}_k \mathbf{D}_k^2 \mathbf{V}_k^\top$ is the eigen-decomposition of the penalty matrix with $(d_{\ell k}^2, \mathbf{v}_{\ell k})$ denoting the ℓ-th eigenvalue/vector pair. As will be demonstrated in the subsequent theorem, spectral basis functions for interaction effects can be defined in a more efficient fashion via the computational tools from Theorem 3.

Theorem 6 (Spectral Tensor Product Penalties). *Suppose that $K \geq p$ and $\mathcal{H}_k$ captures the k-th predictor's main effect for $k = 1, \dots, p$. Given any $\boldsymbol{x} = (x_1, \dots, x_p) \in \mathcal{X}$, the k-th basis vector is defined as $\boldsymbol{R}_k = (R_{11(k)}(x_k, x_{1k}^*), \dots, R_{11(k)}(x_k, x_{r_k k}^*))^\top$ and the k-th penalty matrix is $\mathbf{Q}_k = [R_{11(k)}(x_{\ell k}^*, x_{\ell' k}^*)]$, where $R_{11(k)} = R_{01(k)} + R_{1(k)}$ is the non-constant portion of each predictor's marginal RK function for $k = 1, \dots, p$. Then the k-th spectral basis vector is defined as $\boldsymbol{S}_k = \mathbf{D}_k^{-1} \mathbf{V}_k^\top \boldsymbol{R}_k$, and the corresponding penalty matrix is the identity matrix. Now, suppose that $\mathcal{H}_k$ (for some $k > p$) captures the interaction effect between X_a and X_b for some $a, b \in \{1, \dots, p\}$. If the basis vector is defined as $\boldsymbol{S}_k = \boldsymbol{S}_a \tilde{\otimes} \boldsymbol{S}_b$, where $\tilde{\otimes}$ denotes the Kronecker product, then the penalty matrix is the identity matrix. Now, suppose that $\mathcal{H}_k$ (for some $k > p$) captures the three-way interaction between (X_a, X_b, X_c) for some $a, b, c \in \{1, \dots, p\}$. If the basis vector is defined as $\boldsymbol{S}_k = \boldsymbol{S}_a \tilde{\otimes} \boldsymbol{S}_b \tilde{\otimes} \boldsymbol{S}_c$, then the penalty matrix is the identity matrix. Basis vectors for higher-order interactions can be efficiently constructed in a similar fashion.*

Proof. To prove the theorem, it suffices to prove the result for two-way interactions, given that three-way (and higher-order) interactions can be built by recursively applying the results from the two-way interaction scenario. Specifically, it suffices to show that the penalty matrix corresponding to $\boldsymbol{S}_k = \boldsymbol{S}_a \tilde{\otimes} \boldsymbol{S}_b$ is the identity matrix. Letting $\boldsymbol{\alpha}_k \in \mathbb{R}^{r_k}$ and $\gamma_k \in \mathbb{R}^{r_k}$ denote arbitrary coefficient vectors, the representation for the k-th term is

$$f_k(\boldsymbol{x}) = \boldsymbol{R}_k^\top \boldsymbol{\alpha}_k = \boldsymbol{S}_k^\top \gamma_k \tag{20}$$

where the reparameterized basis and coefficient vector can be written as

$$\begin{aligned} \boldsymbol{S}_k &= \left(\left(\mathbf{D}_a^{-1} \mathbf{V}_a^\top \right) \tilde{\otimes} \left(\mathbf{D}_b^{-1} \mathbf{V}_b^\top \right) \right) \boldsymbol{R}_k \\ \gamma_k &= \left(\left(\mathbf{D}_a \mathbf{V}_a^\top \right) \tilde{\otimes} \left(\mathbf{D}_b \mathbf{V}_b^\top \right) \right) \boldsymbol{\alpha}_k \end{aligned} \tag{21}$$

Now, note that the squared Euclidean norm of the reparameterized coefficients is

$$\begin{aligned} \sum_{\ell=1}^{r_k} \gamma_{\ell k}^2 &= \boldsymbol{\alpha}_k^\top \left(\left(\mathbf{D}_a \mathbf{V}_a^\top \right) \tilde{\otimes} \left(\mathbf{D}_b \mathbf{V}_b^\top \right) \right)^\top \left(\left(\mathbf{D}_a \mathbf{V}_a^\top \right) \tilde{\otimes} \left(\mathbf{D}_b \mathbf{V}_b^\top \right) \right) \boldsymbol{\alpha}_k \\ &= \boldsymbol{\alpha}_k^\top ((\mathbf{V}_a \mathbf{D}_a) \tilde{\otimes} (\mathbf{V}_b \mathbf{D}_b)) \left(\left(\mathbf{D}_a \mathbf{V}_a^\top \right) \tilde{\otimes} \left(\mathbf{D}_b \mathbf{V}_b^\top \right) \right) \boldsymbol{\alpha}_k \\ &= \boldsymbol{\alpha}_k^\top \left(\left(\mathbf{V}_a \mathbf{D}_a \mathbf{D}_a \mathbf{V}_a^\top \right) \tilde{\otimes} \left(\mathbf{V}_b \mathbf{D}_b \mathbf{D}_b \mathbf{V}_b^\top \right) \right) \boldsymbol{\alpha}_k \\ &= \boldsymbol{\alpha}_k^\top (\mathbf{Q}_a \tilde{\otimes} \mathbf{Q}_b) \boldsymbol{\alpha}_k \end{aligned} \tag{22}$$

where the first line plugs in the definition of the squared Euclidean norm, the second line uses the fact that $(A \tilde{\otimes} B)^\top = (A^\top \tilde{\otimes} B^\top)$, the third line uses the fact that $(A \tilde{\otimes} B)(C \tilde{\otimes} D) = (AC) \tilde{\otimes} (BD)$, and the fourth line plugs in the definition of the penalty matrices. □

6. Simulated Example

To demonstrate the potential of the proposed approach, I designed a simple simulation study to compare the performance of the proposed tensor product smoothing approach with the approach of Wood et al. [20], which is implemented in the popular **gamm4** package [32] in R [29]. The **gamm4** package [32] uses the **mgcv** package [33] to build the smooth basis matrices, and then uses the **lme4** package [31] to tune the smoothing parameters (which are treated as variance parameters). For a fair comparison, I have implemented the proposed tensor product spectral smoothing (TPSS) approach using the

lme4 package to tune the smoothing parameters, which I refer to as **tpss4**. This ensures that any difference in the results is due to the employed (reparameterized) basis functions instead of due to differences in the tuning procedure.

Given $p = 2$ predictors with $\mathcal{X} = [0,1] \times [0,1]$, the true mean function is defined as

$$f(x_1, x_2) = f_1(x_1) + f_2(x_2) + f_{12}(x_1, x_2)$$

where $f_1(x_1) = 4\cos(2\pi[x_1 - \pi])$ is the main effect of the first predictor, and $f_2(x_2) = 120(x_2 - 0.6)^5$ is the main effect of the second predictor. The interaction effect is defined as $f_{12}(x_1, x_2) = 0$ for the additive function, and $f_{12}(x_1, x_2) = 4\sin(\pi[x_1 - x_2])$ for the interaction function. Note that this interaction function has been used in previous simulation work that explored tensor product smoothers (see [9,34]). Two different sample sizes were considered $n \in \{1000, 2000\}$. For each sample size and data-generating mean function, n observations were (independently) randomly sampled from $\mathcal{X}$, and the response was defined as $y_i = f(x_{i1}, x_{i2}) + \epsilon_i$, where ϵ_i follows a standard normal distribution.

For both the **gamm4** package and the proposed **tpss4** implementation, (i) I fit the model using $r_k \in \{5, 6, \ldots, 10\}$ marginal knots for each predictor, and (ii) I used restricted maximum likelihood (REML) to tune the smoothing parameters. For the **gamm4** package, the tensor product smooth was formed using the `t2()` function, which allows for main and interaction effects of the predictors. For the **tpss4** method, the implementation in the `smooth2d()` function (see Supplementary Materials) allows for both main and interaction effects. Thus, for both methods, the fit model is misspecified for additive models and correctly specified for interaction models.

I compared the quality of the solutions using the root mean squared error (RMSE)

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\Big(f(x_{i1}, x_{i2}) - \hat{f}_\lambda(x_{i1}, x_{i2})\Big)^2}$$

and the mean absolute error (MAE)

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}\Big|f(x_{i1}, x_{i2}) - \hat{f}_\lambda(x_{i1}, x_{i2})\Big|$$

where $f(x_{i1}, x_{i2})$ is the data-generating mean function and $\hat{f}_\lambda$ is the estimated function. The data generation and analysis procedure was repeated 100 times for each sample size.

Box plots of the RMSE and MAE for each method under each combination of $n \in \{1000, 2000\}$ and $r_k \in \{5, 6, \ldots, 10\}$ are displayed in Figures 3 and 4. As expected, both the RMSE and MAE decrease as the number of knots increases for both methods. For each r_k, the proposed **tpss4** method tends to result in smaller RMSE and MAE values compared to the **gamm4** implementation. For the (misspecified) additive function, the benefit of the proposed approach is noteworthy and persists across all r_k. For the interaction model, the benefit of the proposed **tpss4** approach is particularly noticeable for small r_k, but is still existent for larger numbers of knots.

The runtime for each method is displayed in Figure 5. The proposed **tpss4** method produces runtimes that are slightly larger than the **gamm4** method in most situations. Despite using the same number of marginal knots for each predictor, the **gamm4** approach uses an approximation that estimates slightly fewer coefficients, which is likely causing the timing differences. However, it is possible that these timing differences could be due to running compiled code (in **gamm4**) versus uncompiled code (in **tpss4**). Regardless of the source of the differences, the timing differences are rather small and disappear as r_k increases, which reveals the practicality of the proposed approach.

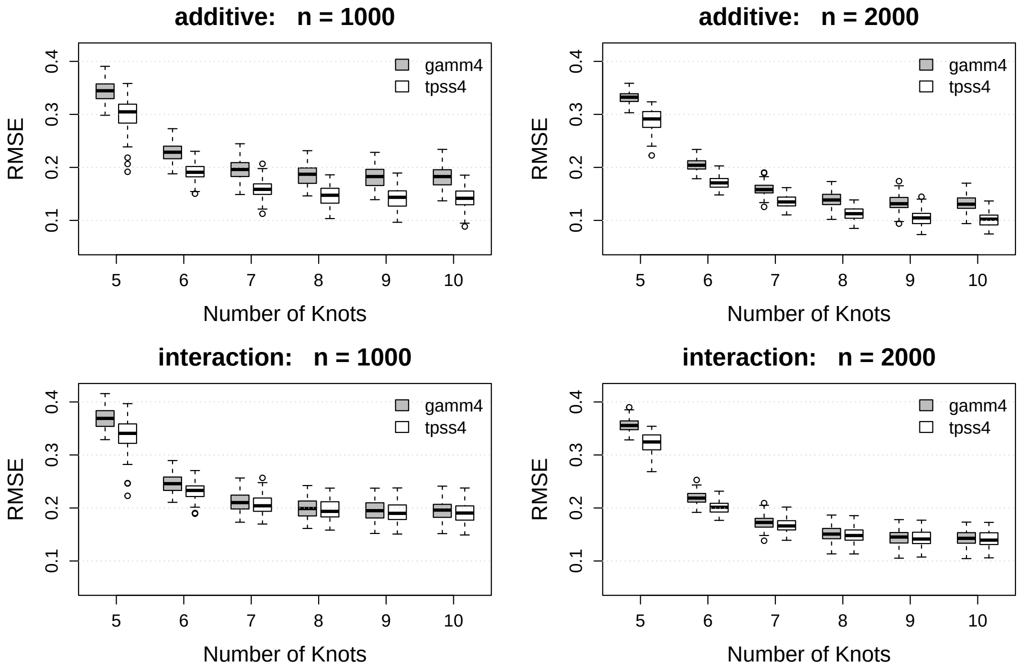

Figure 3. Box plots of the root mean squared error (RMSE) of the function estimate for each method. Rows show results for the additive function (**top**) and interaction function (**bottom**). Columns show the results as a function of the number of knots for $n = 1000$ (**left**) and $n = 2000$ (**right**). Gray boxes denote the results using the **gamm4** packages, whereas white boxes denote the results using the proposed **tpss4** approach. Each box plot summarizes the results across the 100 simulation replications.

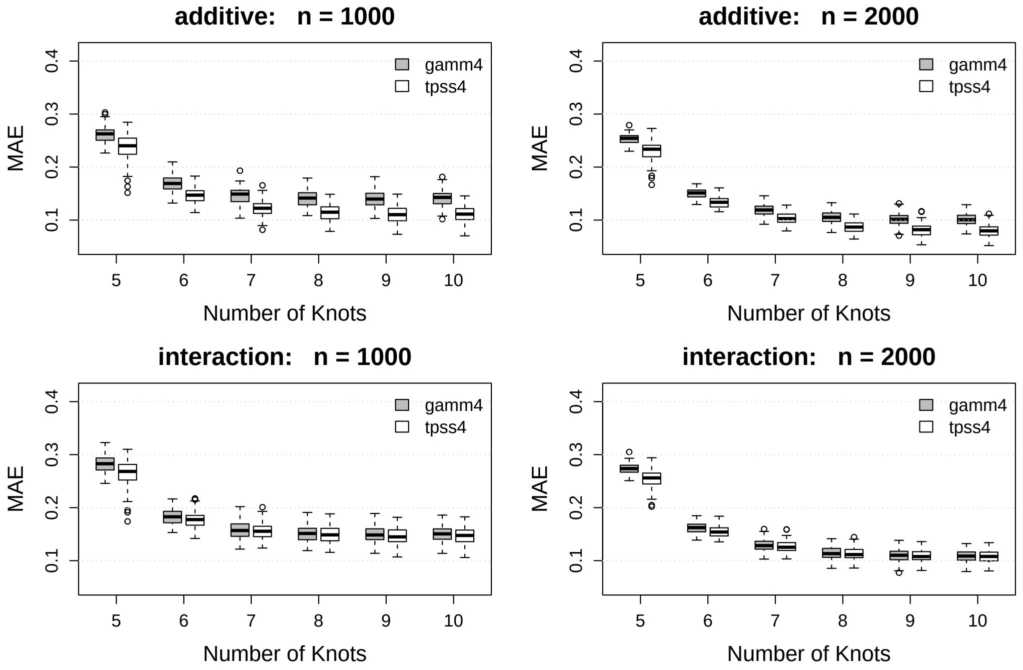

Figure 4. Box plots of the mean absolute error (MAE) of the function estimate for each method. Rows show results for the additive function (**top**) and interaction function (**bottom**). Columns show the results as a function of the number of knots for $n = 1000$ (**left**) and $n = 2000$ (**right**). Gray boxes denote the results using the **gamm4** packages, whereas white boxes denote the results using the proposed **tpss4** approach. Each box plot summarizes the results across the 100 simulation replications.

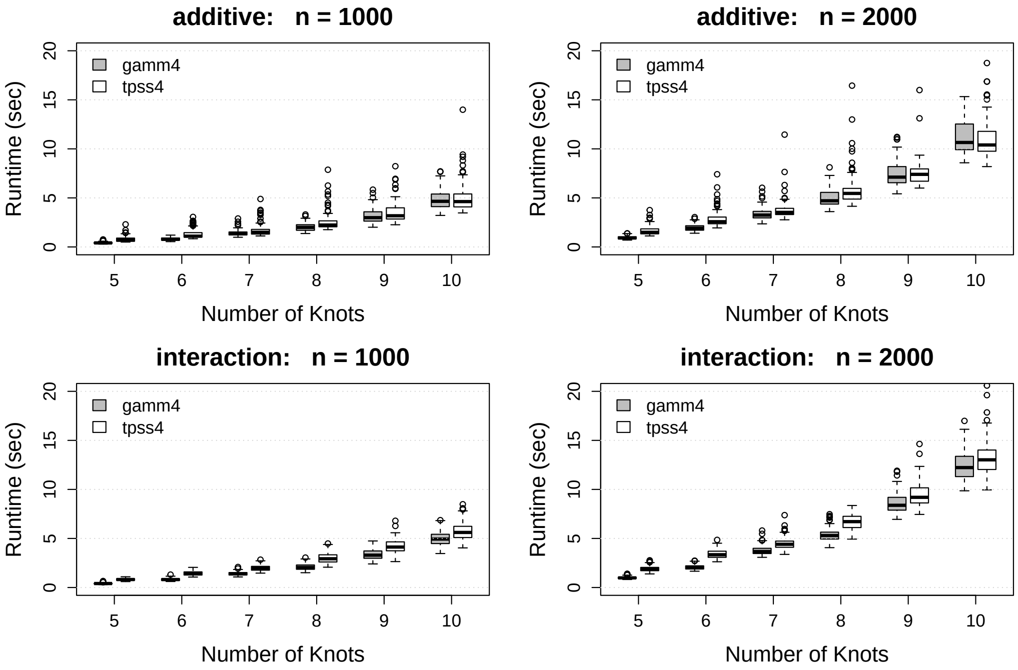

Figure 5. Box plots of the algorithm runtime (in seconds) for each method. Rows show results for the additive function (**top**) and interaction function (**bottom**). Columns show the results as a function of the number of knots for $n = 1000$ (**left**) and $n = 2000$ (**right**). Gray boxes denote the results using the **gamm4** packages, whereas white boxes denote the results using the proposed **tpss4** approach. Each box plot summarizes the results across the 100 simulation replications.

7. Real Data Example

To demonstrate the proposed approach using real data, I make use of the Bike Sharing Dataset [35] from the UCI Machine Learning Repository [36]. This dataset contains the number (count) of bikes rented from the Capital Bike Share system in Washington DC. The rental counts are recorded by the hour from the years 2011 and 2012, which produced a dataset with n = 17,379 observations. In addition to the counts, the dataset contains various situational factors that might affect the number of rented bikes. In this example, I will focus on modeling the number of bike rentals as a function of the hour of the day (which takes values $0, 1, \ldots, 23$) and the month of the year (which takes values $1, 2, \ldots, 12$).

The proposed approach was used to fit a tensor product spectral smoother (TPSS) to the data using 12 knots for the hour variable and 6 knots for the month variable. The counts were modeled on the log10 scale, and then transformed back to the original (data) scale for visualization purposes. As in the simulation study, the smoothing parameters were tuned using the REML method in the **lme4** package. Figure 6 displays the average number of bike rentals by hour and month, as well as the TPSS model predictions. As is evident from the figure, the TPSS solution closely resembles the average data; however, the model predictions are substantially smoother, which improves the interpretation.

Looking at the bike rental patterns by hour of the day, it is evident that there are two surges in the number of rentals: (1) during the morning rush hour (∼8:00–9:00) and (2) during the evening rush hour (∼17:00–18:00). The results also reveal another (smaller) surge that occurs during the lunch hour (∼12:00–13:00). Interestingly, the predictions in Figure 6 reveal that the bike rental surge during the morning rush hour is more localized in time (lasting about one hour), whereas the evening surge is more temporally diffuse (lasting 2–3 h). The bike rentals tend to peak during the afternoon rush hour, and are at their lowest expected value during the evening hours (∼23:00–06:00).

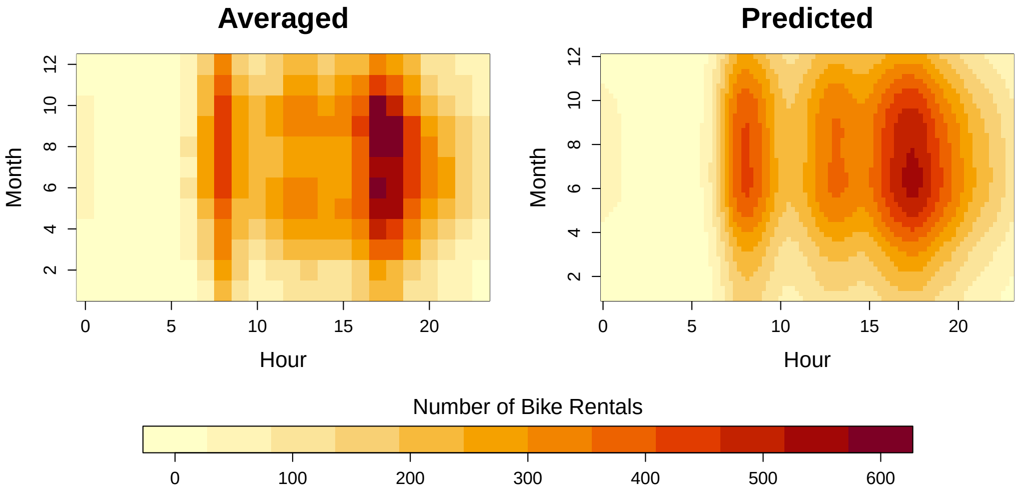

Figure 6. Real data results. (**left**) average number of bike rentals by hour and month. (**right**) predicted number of bike rentals by hour and month.

The month effect is less pronounced than the hour effect, but it still produces some interpretable insights. In particular, we see that there are fewer people using the bikes during the winter months (Dec, Jan, Feb), which is expected. The drops in the number of rentals during the winter are particularly noticeable during the lunch surge, which suggests that fewer people use the bikes to compute for lunch during the winter. The peak in the rentals occurs during the summer months (Jun, Jul, Aug). Combining the hour and month information suggests that the evening rush hour during the summer months is when the Capital Bike Share system sees it greatest surge in demand.

8. Discussion

This paper proposes efficient and flexible approaches for fitting tensor product smoothing spline-like models. The refined smoothing spline approach developed in Section 4 offers an alternative approach for tensor product smoothing splines that penalizes all non-constant effects of the predictors. In particular, Theorem 1 proposes a representer theorem for univariate smoothing spline-like estimators that penalizes all non-constant functions, Theorem 2 provides a tensor product extension of the proposed estimator, and Theorem 3 develops efficient computational tools for forming tensor product penalties. Furthermore, the spectral tensor product approach developed in Section 5 makes it possible to use exact (instead of approximate) tensor product penalties, which can be easily implemented in any standard mixed effects modeling software. In particular, Theorem 4 presents a spectral representer theorem for univariate smoothing, Theorem 5 provides a tensor product extension of the spectral representation, and Theorem 6 develops efficient computational tools for forming tensor product penalties.

The principal results in this paper reveal that if basis functions are formed by taking Kronecker products of spectral spline representations, then the resulting (exact) penalty matrix is the identity matrix. This implies that it is no longer necessary to choose between approximate penalties or costly parameterizations. Note that the results in this paper provide some theoretical support for the tensor product approach of Wood et al. [20], which uses a similar approach with different basis functions. The simulation results support the theoretical results given that the proposed approach (which uses the exact penalty) outperforms the approach of Wood et al. [20] in **gamm4** [32]. As a result, I expect that the proposed approach will be quite useful for fitting (generalized) nonparametric models using modern mixed effects and penalized regression modeling softwares such as **lme4** or **grpnet**. Furthermore, I expect that the proposed approach will be useful for conducting inference with tensor product smoothing splines, e.g., using nonparametric permutation tests [37] or standard hypothesis tests for variance components [38].

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/stats7010003/s1.

Name	Type	Description
smooth2d	R function (.R)	Function for 2-dimensional smoothing
tpss_ex	R script (.R)	Script for the bike sharing analyses and Figure 6
tpss_figs	R script (.R)	Script for reproducing Figures 1 and 2
tpss_sim	R script (.R)	Script for the simulation study and Figures 3–5

Funding: This research was funded by National Institutes of Health (NIH) grants R01EY030890, U01DA046413, and R01MH115046.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: R code to reproduce the results is included as Supplementary Materials. The analyzed data are open source and publicly available.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Helwig, N.E. Multiple and Generalized Nonparametric Regression. In *SAGE Research Methods Foundations*; Atkinson, P., Delamont, S., Cernat, A., Sakshaug, J.W., Williams, R.A., Eds.; SAGE Publications Ltd.: London UK, 2020. [CrossRef]
2. Berry, L.N.; Helwig, N.E. Cross-validation, information theory, or maximum likelihood? A comparison of tuning methods for penalized splines. *Stats* **2021**, *4*, 701–724. [CrossRef]
3. Wahba, G. *Spline Models for Observational Data*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1990.
4. de Boor, C. *A Practical Guide to Splines*, revised ed.; Springer: New York, NY, USA, 2001.
5. Gu, C. *Smoothing Spline ANOVA Models*, 2nd ed.; Springer: New York, NY, USA, 2013. [CrossRef]
6. Wang, Y. *Smoothing Splines: Methods and Applications*; CRC Press: Boca Raton, FL, USA, 2011.
7. Wood, S.N. *Generalized Additive Models: An Introduction with R*, 2nd ed.; Chapman & Hall: Boca Raton, FL, USA, 2017.
8. Hastie, T.; Tibshirani, R. *Generalized Additive Models*; Chapman and Hall/CRC: New York, NY, USA, 1990.
9. Helwig, N.E.; Ma, P. Fast and stable multiple smoothing parameter selection in smoothing spline analysis of variance models with large samples. *J. Comput. Graph. Stat.* **2015**, *24*, 715–732. [CrossRef]
10. Helwig, N.E.; Gao, Y.; Wang, S.; Ma, P. Analyzing spatiotemporal trends in social media data via smoothing spline analysis of variance. *Spat. Stat.* **2015**, *14*, 491–504. [CrossRef]
11. Helwig, N.E.; Shorter, K.A.; Hsiao-Wecksler, E.T.; Ma, P. Smoothing spline analysis of variance models: A new tool for the analysis of cyclic biomechaniacal data. *J. Biomech.* **2016**, *49*, 3216–3222. [CrossRef] [PubMed]
12. Helwig, N.E.; Ruprecht, M.R. Age, gender, and self-esteem: A sociocultural look through a nonparametric lens. *Arch. Sci. Psychol.* **2017**, *5*, 19–31. [CrossRef]
13. Helwig, N.E.; Sohre, N.E.; Ruprecht, M.R.; Guy, S.J.; Lyford-Pike, S. Dynamic properties of successful smiles. *PLoS ONE* **2017**, *12*, e0179708. [CrossRef]
14. Helwig, N.E.; Snodgress, M.A. Exploring individual and group differences in latent brain networks using cross-validated simultaneous component analysis. *NeuroImage* **2019**, *201*, 116019. [CrossRef] [PubMed]
15. Hammell, A.E.; Helwig, N.E.; Kaczkurkin, A.N.; Sponheim, S.R.; Lissek, S. The temporal course of over-generalized conditioned threat expectancies in posttraumatic stress disorder. *Behav. Res. Ther.* **2020**, *124*, 103513. [CrossRef]
16. Almquist, Z.W.; Helwig, N.E.; You, Y. Connecting Continuum of Care point-in-time homeless counts to United States Census areal units. *Math. Popul. Stud.* **2020**, *27*, 46–58. [CrossRef]
17. Helwig, N.E. Efficient estimation of variance components in nonparametric mixed-effects models with large samples. *Stat. Comput.* **2016**, *26*, 1319–1336. [CrossRef]
18. Helwig, N.E.; Ma, P. Smoothing spline ANOVA for super-large samples: Scalable computation via rounding parameters. *Stat. Its Interface* **2016**, *9*, 433–444. [CrossRef]
19. Demmler, A.; Reinsch, C. Oscillation matrices with spline smoothing. *Numer. Math.* **1975**, *24*, 375–382. [CrossRef]
20. Wood, S.N.; Scheipl, F.; Faraway, J.J. Straightforward intermediate rank tensor product smoothing in mixed models. *Stat. Comput.* **2013**, *23*, 341–360. [CrossRef]
21. Kimeldorf, G.; Wahba, G. Some results on Tchebycheffian spline functions. *J. Math. Anal. Appl.* **1971**, *33*, 82–95. [CrossRef]
22. Gu, C.; Kim, Y.J. Penalized likelihood regression: General formulation and efficient approximation. *Can. J. Stat.* **2002**, *30*, 619–628. [CrossRef]
23. Kim, Y.J.; Gu, C. Smoothing spline Gaussian regression: More scalable computation via efficient approximation. *J. R. Stat. Soc. Ser. B* **2004**, *66*, 337–356. [CrossRef]

24. Moore, E.H. On the reciprocal of the general algebraic matrix. *Bull. Am. Math. Soc.* **1920**, *26*, 394–395. [CrossRef]
25. Penrose, R. A generalized inverse for matrices. *Math. Proc. Camb. Philos. Soc.* **1955**, *51*, 406–413. [CrossRef]
26. Wang, Y. Mixed effects smoothing spline analysis of variance. *J. R. Stat. Soc. Ser. B* **1998**, *60*, 159–174. [CrossRef]
27. Wang, Y. Smoothing spline models with correlated random errors. *J. Am. Stat. Assoc.* **1998**, *93*, 341–348. [CrossRef]
28. Helwig, N.E. Regression with ordered predictors via ordinal smoothing splines. *Front. Appl. Math. Stat.* **2017**, *3*, 15. [CrossRef]
29. R Core Team. *R: A Language and Environment for Statistical Computing*; R Foundation for Statistical Computing: Vienna, Austria, 2023; R version 4.3.1.
30. Helwig, N.E. *grpnet: Group Elastic Net Regularized GLM*; R package version 0.2; Comprehensive R Archive Network: Vienna, Austria, 2023.
31. Bates, D.; Mächler, M.; Bolker, B.M.; Walker, S.C. Fitting Linear Mixed-Effects Models Using lme4. *J. Stat. Softw.* **2015**, *67*, 1–48. [CrossRef]
32. Wood, S.; Scheipl, F. *gamm4: Generalized Additive Mixed Models Using 'mgcv' and 'lme4'*; R package version 0.2-6; Comprehensive R Archive Network: Vienna, Austria, 2020.
33. Wood, S.N. *mgcv: Mixed GAM Computation Vehicle with GCV/AIC/REML Smoothness Estimation and GAMMs by REML/PQL*; R package version 1.9-1; Comprehensive R Archive Network: Vienna, Austria, 2023.
34. Helwig, N.E. Spectrally sparse nonparametric regression via elastic net regularized smoothers. *J. Comput. Graph. Stat.* **2021**, *30*, 182–191. [CrossRef]
35. Fanaee-T, H.; Gama, J. Event labeling combining ensemble detectors and background knowledge. *Prog. Artif. Intell.* **2013**, *2*, 1–15. [CrossRef]
36. Kelly, M.; Longjohn, R.; Nottingham, K. The University of California Irvine (UCI) Machine Learning Repository. Available online: https://archive.ics.uci.edu/ (accessed on 26 December 2023).
37. Helwig, N.E. Robust Permutation Tests for Penalized Splines. *Stats* **2022**, *5*, 916–933. [CrossRef]
38. Kuznetsova, A.; Brockhoff, P.B.; Christensen, R.H.B. lmerTest Package: Tests in Linear Mixed Effects Models. *J. Stat. Softw.* **2017**, *82*, 1–26. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.