A sandwich smoother for spatio-temporal functional data

Joshua P. French[a], Piotr S. Kokoszka[b]

[a]*Department of Mathematical and Statistical Sciences, University of Colorado Denver, Campus Box 170, PO Box 173364, Denver, CO 80217-3364*
[b]*Department of Statistics, Colorado State University*

Abstract

Statistical analysis of spatio-temporal data has been evolving to handle increasingly large data sets. For example, the North American CORDEX program is producing daily values of climate-related variables on spatial grids with approximately 100,000 locations over 150 years. Smoothing of such massive and noisy data is essential to understanding their spatio-temporal features. It also reduces the size of the data by representing them in terms of suitable basis functions, which facilitates further computations and statistical analysis. Traditional tensor-based methods break down under the size of such massive data. We develop a penalized spline method for representing such data using a generalization of the sandwich smoother proposed by Xiao et al. (2013). Unlike the original method, our generalization treats the spatial and temporal dimensions distinctly and allows the methodology to be directly applied to non-gridded data. We demonstrate the practicality of the methodology using both simulated and real data. The new smoother, as well as the original sandwich smoother, are implemented in the `hero` R package.

Email address: `joshua.french@ucdenver.edu` (Joshua P. French)

Keywords: Massive climate data, Smoothing, Spatio-temporal data

1. Introduction

Statistical analysis of spatio-temporal data has changed over time as the data sets have grown increasingly large. For example, the North American component of the Coordinated Regional Climate Downscaling Experiment (CORDEX, Giorgi et al., 2009) program is producing daily values of climate-related variables on (approximately) 0.44°, 0.22°, and 0.11° native rotated-pole grids (50 km, 25 km, and 12.5 km resolutions, respectively), which correspond to spatial grids with approximately, 18,000, 100,000, and 400,000 locations, respectively. Our goal is to develop computationally feasible and statistically meaningful methodology for describing the spatio-temporal features of very large spatio-temporal data sets such as those generated by the North American Regional Climate Change Assessment Program (NARCCAP, Mearns et al., 2009, 2012) or the North American CORDEX (NA-CORDEX) programs. Thus, our goals relate more to *describing* the spatio-temporal structure of our data rather than *predicting* unobserved values, and we focus on methodology for representing and compressing the data using smooth functions.

Smoothing and dimension reduction by means of basis representations play a fundamental role in functional data analysis (FDA, Ramsay et al. 2009). With the emerging applications of FDA to spatio-temporal data, e.g., Aston et al. (2016), Liu et al. (2017), Constantinou et al. (2017), Gromenko et al. (2017a), Gromenko et al. (2017b), French et al. (2019), comes the need to develop effective and meaningful smoothing and dimension reduction tools

for such data.

Methods have been proposed for investigating data of similar magnitude. Medical imaging data frequently have a magnitude similar to the NACORDEX data. However, it seems that the research goals of methods for analysis of that data are quite different (e.g., modeling curves of event-related potential (Zhu et al., 2018) or representing two-dimensional manifolds (Lila et al., 2016)) and are not suitable for our purpose. Methodology directly related to modeling massive spatio-temporal data sets tends to focus on prediction (e.g., Ma and Kang (2019); Jurek and Katzfuss (2018)). Additionally, user-friendly software or code are frequently not provided with these methods, making them difficult to apply to other data.

We develop a penalized spline method for representing continuous spatio-temporal data with a dominant smooth component. Our approach builds on the original sandwich smoother (OSS) proposed by Xiao et al. (2013). Multivariate splines are frequently constructed using tensor products of univariate splines associated with each dimension, and a penalized fitting criterion is typically used to estimate the relevant model parameters. Xiao et al. (2013) proposed a special form of the penalty matrix that allows for highly efficient computation using linear array operations. To deal meaningfully with the spatio-temporal data introduced above, we develop a framework that inherits the computational advantages of the OSS while treating the spatial and temporal dimensions differently. We recommend utilizing compactly-supported radial basis splines to represent the spatial dimension instead of the tensor product B-splines recommended by Xiao et al. (2013). This modification allows us to apply the generalized sandwich smoother to non-gridded spatio-

temporal data, extending the applicability of the methodology to our desired context. Our approach is computationally faster than standard tensor-based methods and more widely and meaningfully applicable in the context of climate data than the OSS.

The paper is organized as follows. In Section 2, we describe some relevant smoothing methods and explain why they are not ideal for the data that motivate this research. The proposed spatio-temporal sandwich smoother is introduced in Section 3. We demonstrate its performance in Section 4 using synthetic data constructed from a large-scale climate experiment, and in Section 5 by applying it to two large climate data sets to which the OSS cannot be applied. We summarize our results and discuss future work in Section 6.

2. Traditional splines and fast p-splines

Consider a d-dimensional smoothly-varying process $\{\mu(\mathbf{t}), \mathbf{t} \in D\}$, where $\mathbf{t} = (t_1, t_2, \ldots, t_d)$ is the d-dimensional coordinate vector, and D is a bounded d-dimensional region in $\mathbb{R}^d$. In what follows, we generally focus on spatial and spatio-temporal data, so $d = 2$ or $d = 3$.

We will adapt our notation slightly depending on the context. In general, a process in two dimensions will be denoted $\mu(u, v)$, with $(u, v) \in D \subset \mathbb{R}^2$. The two spatial dimensions will be referred to as the u- and v-dimensions. In the special case that the process is observed in geographical space, we let $\mathbf{s} := (u, v)$ refer to a spatial location. In this case, the u- and v-dimensions would typically be associated with something like longitude and latitude in the geographic coordinate system or easting and northing in the Universal

Transverse Mercator (UTM) coordinate system. A generic three-dimensional process will be denoted $\mu(u, v, t)$, where $(u, v, t) \in D \times [0, \infty)$. We will specifically focus on data observed in two-dimensional geographic space and one-dimensional time, and in that case, we will utilize the notation $\mu(\mathbf{s}, t)$, where $\mathbf{s} \in D$ and $t \in [0, \infty)$.

The data are assumed to be observed with error or noise, so that

$$y(\cdot) = \mu(\cdot) + \epsilon(\cdot), \tag{1}$$

where $(\cdot)$ is replaced by the appropriate coordinates in d-dimensional space, and $\epsilon(\cdot)$ is an error process.

In the remainder of this section, we provide some background that helps understand the need for a new approach.

2.1. Standard multivariate tensor splines

We first consider smoothing a two-dimensional surface. Suppose we observe a partial realization of a two-dimensional process $\{y(u, v) = \mu(u, v) + \epsilon(u, v), (u, v) \in D\}$. The n observed values of y are denoted by the column vector $\mathbf{y} = (y_1, y_2, \ldots, y_n)$, with $y_i := y(u_i, v_i)$, $i = 1, 2, \ldots, n$.

A standard way to smooth the observed data, from a functional data perspective, is to use a tensor product of univariate basis functions. Let $\{b_k^1, k = 1, 2, \ldots, c_1\}$ and $\{b_l^2, l = 1, 2, \ldots, c_2\}$ denote a set of univariate basis functions (e.g., natural cubic splines, B-splines, monomials (Ramsey and Silverman, 2005; Hastie et al., 2009)) associated with dimensions u and v, respectively. The tensor product spline is defined by

$$\sum_{1 \le k \le c_1, 1 \le l \le c_2} \theta_{k,l} b_k^1(u) b_l^2(v),$$

where $\{\theta_{k,l}, 1 \leq k \leq c_1, 1 \leq l \leq c_2\}$ is a set of associated coefficients. The original data are then modeled as

$$y(u,v) = \sum_{1 \leq k \leq c_1, 1 \leq l \leq c_2} \theta_{k,l} b_k^1(u) b_l^2(v) + \epsilon(u,v).$$

The smoothed data are obtained as

$$\hat{\mu}(u,v) = \sum_{1 \leq k \leq c_1, 1 \leq l \leq c_2} \hat{\theta}_{k,l} b_k^1(u) b_l^2(v),$$

and $\hat{\mathbf{y}} := (\hat{y}_1, \ldots, \hat{y}_n) := (\hat{\mu}(u_1, v_1), \ldots, \hat{\mu}(u_n, v_n))$. Let $\boldsymbol{\theta} := \text{vec}(\theta_{k,l})_{k=1,\ldots,c_1, l=1,\ldots,c_2}$ be the column vector of coefficients, where vec is the operation that stacks each column of a matrix into a vector. An estimate of $\boldsymbol{\theta}$ is obtained by minimizing

$$||\mathbf{y} - (\mathbf{B}_2 \otimes \mathbf{B}_1)\boldsymbol{\theta}||^2 + \boldsymbol{\theta}^T \mathbf{P}_\lambda \boldsymbol{\theta} = ||\mathbf{y} - \mathbf{B}\boldsymbol{\theta}||^2 + \boldsymbol{\theta}^T \mathbf{P}_\lambda \boldsymbol{\theta} \tag{2}$$

with respect to $\boldsymbol{\theta}$, where $\mathbf{B}_1 := [b_k^1(u_i)]_{1 \leq i \leq n, 1 \leq k \leq c_1}$, $\mathbf{B}_2 := [b_l^2(v_i)]_{1 \leq i \leq n, 1 \leq l \leq c_2}$, $\mathbf{B} := \mathbf{B}_2 \otimes \mathbf{B}_1$, and

$$\mathbf{P}_\lambda := \lambda_1 \mathbf{I}_{c_2} \otimes \mathbf{P}_1 + \lambda_2 \mathbf{P}_2 \otimes \mathbf{I}_{c_1} \tag{3}$$

is a penalty matrix. More specifically, $\mathbf{P}_1$ and $\mathbf{P}_2$ are penalty matrices associated with the basis functions for the u and v dimensions, respectively, and $\lambda := (\lambda_1, \lambda_2)$ is the vector of smoothing parameters. Penalties are commonly related to derivative properties of the splines (Reinsch, 1967; O'Sullivan, 1986) or differences in the basis coefficients (Eilers and Marx, 1996). In this context, the smoothed data are obtained via the equation

$$\hat{\mathbf{y}} = \mathbf{S}_\lambda \mathbf{y}, \tag{4}$$

where

$$\mathbf{S}_\lambda = \mathbf{B}(\mathbf{B}^T\mathbf{B} + \mathbf{P}_\lambda)^{-1}\mathbf{B}^T.$$

The most difficult challenge for applying tensor product smoothing splines is typically the choice of smoothing parameters. Frequently, these parameters are chosen to minimize the Generalized Crossvalidation (GCV) statistic of Craven and Wahba (1978), defined as

$$GCV(\lambda) = \frac{n\text{trace}\{\mathbf{y}^T(\mathbf{I} - \mathbf{S}_\lambda)^2\mathbf{y}\}}{\{\text{trace}(\mathbf{I} - \mathbf{S}_\lambda)\}^2}. \tag{5}$$

The smoothing matrix $\mathbf{S}_\lambda$ greatly simplifies in the one-dimensional context, and computation of the GCV can be efficiently done for many different values of (scalar) λ by solving a generalized eigenvalue problem (Ramsey and Silverman, 2005, Ch. 5). In the two-dimensional setting, a similar approach can be used if isotropic smoothing is done, i.e., when $\lambda_1 = \lambda_2$. In the general case of $\lambda_1 \neq \lambda_2$, the computation of the GCV becomes much more expensive since there are no helpful features to exploit (Eilers et al., 2015). In the special case of gridded data, efficient algorithms can be used to smooth the data when $\lambda_1 \neq \lambda_2$ (Eilers et al., 2006; Currie et al., 2006).

2.2. The sandwich smoother

Similar to Eilers et al. (2006) and Currie et al. (2006), Xiao et al. (2013) consider data on an $n_1 \times n_2$ grid. The sampled data can be represented as an $n_1 \times n_2$ matrix $\mathbf{Y} := [y_{i,j}]_{i=1,\ldots,n_1,j=1,\ldots,n_2}$, where $y_{i,j} := y(u_i, v_j)$; similarly, $\hat{\mathbf{Y}} := [\hat{y}_{i,j}]_{i=1,\ldots,n_1,j=1,\ldots,n_2}$ with $\hat{y}_{i,j} = \hat{\mu}(u_i, v_j)$. As in the previous subsection, $\mathbf{y} := \text{vec}(\mathbf{Y})$ and $\hat{\mathbf{y}} := \text{vec}(\hat{\mathbf{Y}})$. Xiao et al. (2013) utilize the P-splines, i.e., B-spline bases and coefficient difference penalties, which were first proposed

by Eilers and Marx (1996); the penalty matrix for P-splines is

$$\mathbf{P}_i := \mathbf{D}_i^T \mathbf{D}_i, \tag{6}$$

where $\mathbf{D}_i$ is a differencing matrix of order m_i.

Xiao et al. (2013) proposed smoothing $\mathbf{Y}$ as $\hat{\mathbf{Y}} = \mathbf{S}_1 \mathbf{Y} \mathbf{S}_2$ (from which the nickname "sandwich smoother" is derived), where $\mathbf{S}_1$ and $\mathbf{S}_2$ are smoother matrices associated with dimensions u and v, respectively. Properties of the tensor product can be used to show that $\hat{\mathbf{y}} = (\mathbf{S}_2 \otimes \mathbf{S}_1)\mathbf{y}$.

The important difference between the traditional tensor product smooth and the sandwich smoother is the way the penalty is computed. For simplicity, assume that penalties are based on the P-spline difference penalty in Equation (6). The traditional penalty and sandwich smoother penalty are constrasted in Table 1. While the differences may appear to be subtle, the form of the sandwich smoother penalty allows for gains in computational efficiency that can span orders of magnitude (Xiao et al., 2013, Sections 5, 7). This is mainly because the GCV statistic in Equation (5) for the OSS only requires two eigenvalue decompositions of relatively small matrices of size $c_i \times c_i$ for each penalty, $i = 1, 2$, along with some very elementary matrix operations, without requiring matrix inversion. In contrast, computing the GCV statistic for the traditional tensor product smooth (even when the data are on a grid and the the generalized linear array model (GLAM) algorithm of Currie et al. (2006) may be utilized) requires inverting a $c_1 c_2 \times c_1 c_2$ matrix for every combination of λ_1 and λ_2. The details of efficiently computing the GCV using the sandwich smoother are described by Xiao et al. (2013, Section 2).

	Penalty
Traditional	$\lambda_1 \mathbf{I}_{c_2} \otimes \mathbf{D}_1^T \mathbf{D}_1 + \lambda_2 \mathbf{D}_2^T \mathbf{D}_2 \otimes \mathbf{I}_{c_1}$
Sandwich	$\lambda_1 \mathbf{B}_2^T \mathbf{B}_2 \otimes \mathbf{D}_1^T \mathbf{D}_1 + \lambda_2 \mathbf{D}_2^T \mathbf{D}_2 \otimes \mathbf{B}_1^T \mathbf{B}_1 + \lambda_1 \lambda_2 \mathbf{D}_2^T \mathbf{D}_2 \otimes \mathbf{D}_1^T \mathbf{D}_1$

Table 1: The penalty terms for the traditional two-dimensional tensor product smoothing splines and the sandwich smoother.

The OSS can be generalized to $d \geq 3$ dimensions, where

$$y_{i_1 i_2, \ldots, i_d} = \mu(t_{i_1}, \ldots, t_{i_d}) + \epsilon(t_{i_1}, \ldots, t_{i_d}), \quad 1 \leq i_k \leq n_k, 1 \leq k \leq d.$$

In this setting, the data can be arranged in an array $\mathbf{Y} := [y_{i_1 i_2 \ldots, i_d}]_{1 \leq i_k \leq n_k, 1 \leq k \leq d}$ and the generalized linear array model (GLAM) algorithm Currie et al. (2006) can be used to efficiently fit the model to the data (both in terms of speed and storage requirements). Additionally, the properties allowing for fast computation of the GCV statistic in the bivariate setting directly generalize to higher dimensions, enabling quick selection of the smoothing parameters.

Xiao et al. (2013) demonstrate that the OSS is much faster to implement and apply than the traditional tensor product P-spline model, even when the GLAM algorithm is utilized in fitting the traditional model (see Table 2 of Xiao et al. (2013)). Similar patterns are observed in higher dimensions.

2.3. Difficulties applying the sandwich smoother to spatio-temporal data

The OSS can be extended to d-dimensions, so it may be possible to directly apply it to three-dimensional spatio-temporal data on a grid. However, there are problems with naively applying the OSS to spatio-temporal arrays.

First, the spatial dimension is fundamentally different than the time dimension, so it may be appropriate to use different types of basis functions in these two dimensions. A more serious issue and restriction is that the OSS requires the data to be on a rectilinear grid, which is atypical for geographically-referenced data, even when produced by a designed experiment. A rectilinear grid is defined by all combinations of points constructed from the increasing sequences $u_1, \ldots, u_{n_1}$ and $v_1, \ldots, v_{n_2}$, i.e., by the set $\{(u_i, v_j), i = 1, \ldots, n_1, j = 1, \ldots, n_2\}$ (Reed et al., 1996). Many large spatio-temporal data sets are observed on irregular grids (French, 2017) or may not defined on a grid of any type; they are formed by collections of spatially-referenced time series available at more or less irregular locations and over domains with irregular shapes. Examples include pollution data, historical weather data, or ionosphere data studied by Gromenko et al. (2017b). Examples related to computer climate models are presented in Section 5. Figure 1 provides a constrast between a rectlinear and irregular grid.

A final issue with applying the OSS to the spatio-temporal data we consider is that Euclidean distance is an inappropriate distance metric. The data we consider typically are referenced using longitude/latitude coordinates on a large domain. In that context, Euclidean distance is no longer appropriate for measuring distance between points and great circle distance should be used instead. This is a critical detail for the use of tensor-product splines, which implicitly assume that the basis functions will not change if the data locations are translated in the spatial domain. Thus, the OSS is typically not appropriate for geographically-referenced data covering large spatial domains.

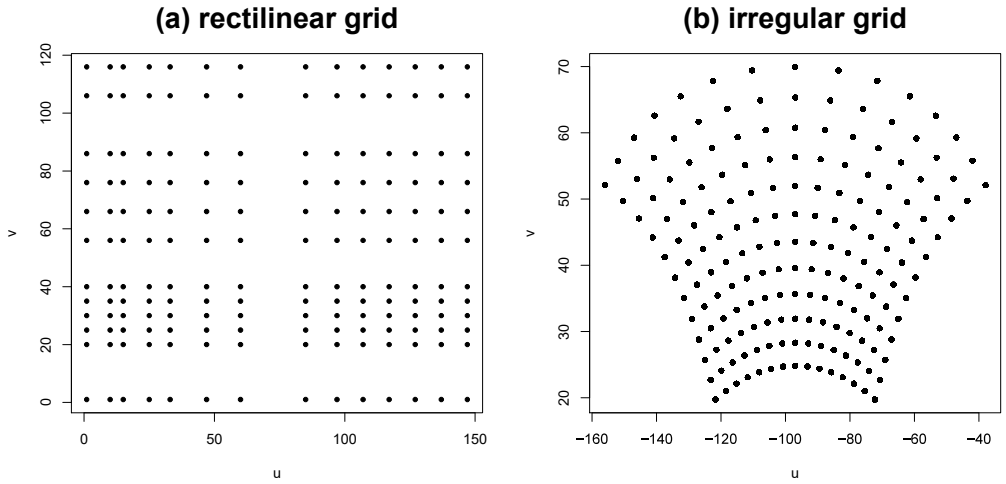

Figure 1: A contrast between a rectilinear and irregular grid.

3. Spatio-temporal sandwich smoother

We propose a new spatio-temporal sandwich smoother (STSS) appropriate for spatio-temporal data that addresses the issues raised in Section 2.3. It inherits many of the computational benefits of the OSS while also applying to more general settings. The two main building blocks of the new smoother are 1) suitable spatial basis functions with a suitable penalty, and 2) a more flexible spatial structure. The following subsections explain the details.

3.1. Radial basis functions

Radial basis functions provide an alternative to tensor product basis functions for smoothing bivariate data (Ruppert et al., 2003). A radial basis function is a function whose value depends on the distance from a knot location $\mathbf{k} \in D$. For a specific knot $\mathbf{k}_i$, the ith radial basis function, $r_i(\mathbf{s})$ is a function depending only on $|\mathbf{k}_i - \mathbf{s}|$, where $|\cdot|$ is a distance metric. For example, the

bi-square basis function (Cressie and Johannesson, 2008) is defined by

$$r_i(\mathbf{s}) = (1 - |\mathbf{k}_i - \mathbf{s}|^2\phi^{-2})^2 I(|\mathbf{k}_i - \mathbf{s}| < \phi), \tag{7}$$

where ϕ is a bandwidth parameter that controls the rate of decay of the basis function. The Gaussian radial basis function is another radial basis function, defined by $r_i(\mathbf{s}) = \exp(-|\mathbf{k}_i - \mathbf{s}|^{-2}\phi^{-2})$. The density estimation literature related to kernel smoothing provides many additional radial basis function by generalizing one-dimensional distance to two dimensions. Several of these are summarized in Waller and Gotway (2004). Many spatial covariance functions are examples of radial basis functions (Schabenberger and Gotway, 2005).

We must give special consideration to the type of radial basis functions we want to use in our smoothing. The B-splines used by Xiao et al. (2013) have many beneficial properties that we would like to maintain in our spatial smoothing. B-splines have: 1. compact support, 2. easy-to-compute derivatives, 3. specifiable parameters related to smoothness. We desire to retain these properties for our radial basis functions. While many radial basis functions satisfy at least one of these properties, the Wendland covariance function (Wendland, 1995) satisfies all three. The basic form of the Wendland covariance function is

$$r(h) = \begin{cases} \sum_{j=1}^{N} a_j h^j & 0 \leq h \leq \phi \\ 0 & \phi < h \end{cases},$$

where h is distance between two points in d-dimensional space, N is the desired degree of the polynomial (and is related to smoothness), ϕ defines the support of the function, and $\{a_j, j = 1, \ldots, N\}$ are a set of non-zero coefficients. In addition to the dependence of the covariance function on

the range parameter ϕ, the specific form of the covariance function equation depends on N and d; Table 1 of Wendland (1995) provides several examples. Additional details about the Wendland covariance function may be found in Wendland (1995) and Gneiting (2002).

3.2. Data structure

The OSS assumes an array structure $\mathbf{Y} := [y_{i,j}]_{1\leq i\leq n_1, 1\leq j\leq n_2}$ for $d = 2$, but can be extended to higher dimensions. In order to exploit the computational benefits of the OSS, we need to retain this structure while dealing with the issues highlighted in Section 2.3.

We retain an array structure in our spatio-temporal data by treating the spatial dimension as the first dimension and the temporal dimension as the second dimension. Suppose that responses are observed at n_s spatial locations at n_t times. Note that the spatial locations do not need to be on a grid, but the responses must be observed for all locations at each of the n_t times, i.e., there are no missing data. In this setting, our data can be represented as

$$\mathbf{Y} = [y(\mathbf{s}_i, t_j)]_{i=1,\ldots,n_s, j=1,\ldots,n_t}.$$

As will be shown, using this form of data representation along with radial basis functions in the spatial dimension solves the issues with treating the spatial and temporal dimensions the same, having non-gridded spatial data, and dealing with non-Euclidean distance.

3.3. Penalty function

Xiao et al. (2013) utilize the P-splines of Eilers and Marx (1996) in the OSS. This means that in addition to utilizing B-spline basis functions in

Table 2: Differencing matrices for several orders of m for coefficients $c_1, c_2, \ldots, c_5$.

Order	c_1	c_2	c_3	c_4	c_5
	1	-1			
		1	-1		
$m = 1$			1	-1	
				1	-1
	1	-2	1		
$m = 2$		1	-2	1	
			1	-2	1
	1	-3	3	-1	
$m = 3$		1	-3	3	-1
$m = 4$	1	-4	6	-4	1

each direction, the penalty for each set of basis functions is given by the crossproduct of $(\mathbf{D}^m)^T\mathbf{D}^m$, where $\mathbf{D}^m$ is a differencing matrix of order m (cf., Eilers and Marx (1996) and Wood (2017)). Examples of $\mathbf{D}^m$ are shown for $m = 1, 2, 3, 4$ for 5 coefficients $c_1, \ldots, c_5$ in Table 2 (up to a factor -1). Essentially, differences between adjacent coefficients are iteratively repeated.

While one could technically utilize the differencing matrices to penalize the coefficients of radial basis functions, this does not seem sensible since there is no coherent ordering of the associated spatial knot locations. Specifically, the coefficients of a univariate B-spline are associated with knot locations $u_1, u_2, \ldots, u_c$. The coefficients of a radial basis functions are associated with knot locations $\mathbf{k}_1, \mathbf{k}_2, \ldots, \mathbf{k}_c$. There is a natural ordering of the u-locations, but not for the $\mathbf{k}$-locations, so "adjacent" locations for the radial

basis must be defined judiciously.

We propose penalizing the radial basis coefficients using a *spatial* differencing matrix of order m, $\mathbf{S}^m$, based on the concept of nearest neighbors. Consider spatial locations $\mathbf{k}_1, \ldots, \mathbf{k}_c$. There are many ways to definine the neighbors of each knot location (Rue and Held, 2005). Let $\mathcal{N}_i$ denote the neighbors of knot location $\mathbf{k}_i$. Note that for the differencing matrices in Table 2, every row is a contrast (i.e., the coefficients sum to 0). We will use this property to recursively define a spatial differencing matrix. Let $\mathbf{S}_i^m$ denote the ith row of the spatial difference matrix of order m and $\mathbf{S}_{i,j}^m$ denote position j of the ith row. For $i = 1, 2, \ldots, c$, we define

$$\mathbf{S}_{i,j}^1 := \begin{cases} \#\{\mathcal{N}_i\} & \text{if } j = i \\ -1 & \text{if } \mathbf{k}_j \in \mathcal{N}_i, j \neq i \\ 0 & \text{otherwise.} \end{cases} \tag{8}$$

Thus, in row i, coefficient j receives a weight of -1 if $\mathbf{k}_j$ is a neighbor of $\mathbf{k}_i$, while the weight for coefficient i is the number of nearest neighbors for $\mathbf{k}_i$. We define the mth order neighbors of knot $\mathbf{k}_i$ to be the union of the neighbors of the $(m-1)$th order neighbors of knot $\mathbf{k}_i$ (excluding $\mathbf{k}_i$ itself and neighbors from any previous order):

$$\mathcal{N}_i^m := \left\{ \bigcup_{j:\mathbf{k}_j \in \mathcal{N}_i^{m-1}} \mathcal{N}_j^{m-1} \right\} \setminus \left\{ \mathbf{k}_i \cup \left\{ \bigcup_{j=1}^{m-1} \mathcal{N}_i^j \right\} \right\}, \tag{9}$$

where $\mathcal{N}_i^1 = \mathcal{N}_i$, $i = 1, \ldots, c$. The ith row of the mth order spatial difference is then defined to be the difference between the ith row of the $(m-1)$th order spatial difference matrix and the sum of the rows of the $(m-1)$th order spatial difference matrix corresponding to the mth order neighbors of

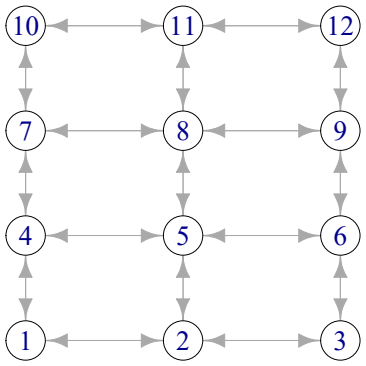

Figure 2: A graph displaying relationships between coordinates. Arrows indicate locations that are neighbors.

knot $\mathbf{k}_i$:

$$\mathbf{S}_i^m = \mathbf{S}_i^{m-1} - \sum_{j:\mathbf{k}_j \in \mathcal{N}_i^m} \mathbf{S}_j^{m-1}. \tag{10}$$

Lastly, just as $\mathbf{P}_i = \mathbf{D}_i^T \mathbf{D}_i$ in Equation (6) in the traditional P-splines context, the penalty matrix $\mathbf{P}_s$ for the spatial dimension is

$$\mathbf{P}_s := (\mathbf{S}^m)^T \mathbf{S}^m. \tag{11}$$

The goal of the spatial differencing penalty is to control the magnitude of the difference of coefficients associated with knot locations near one another in space. As the difference order m increases, the penalty is stricter in the sense that more neighboring coefficients are included in each contrast, which favors coefficients that vary more slowly across the spatial domain.

We illustrate with a simple example of the spatial differencing penalty. Consider 12 knot locations on a regular grid defined by u-locations 1, 2, and 3 and v-locations 1, 2, 3, 4. The indices of the knots are provided in Table 3, and Figure 2 displays the relationships between the knots. Arrows connect

Table 3: Indices of coordinates for spatial difference penalty example.

Index	u-coordinate	v-coordinate
1	1	1
2	2	1
3	3	1
4	1	2
5	2	2
6	3	2
7	1	3
8	2	3
9	3	3
10	1	4
11	2	4
12	3	4

knots that are neighbors. The spatial penalty matrices $\mathbf{S}^1$ and $\mathbf{S}^2$ for these knots are provided in Table 4. In a spatial context, the spatial difference penalty makes much more sense than that standard P-splines differencing penalty.

The reader may note that if one constructs the traditional tensor-product penalty from Table 1 based on differencing the u and v coordinates of points on a rectlinear grid that one arrives at a penalty identical to the spatial difference penalty shown for $m = 1$ in Table 4. However, one should note that this only occurs when the knots are on a rectilinear grid and does not occur for $m \geq 2$. More importantly, the traditional tensor-product penalty based on differences of linearly-adjacent coefficients does not naturally extend to two-dimensional space. However, the spatial difference penalty applies to knots located arbitrarily in space without requiring a rectilinear grid.

3.4. Additional implementation details

We now briefly discuss some additional details of practical application of the STSS to real data.

Several scales of radial basis functions should be used to smooth the spatial dimension in order capture differing scales of data variability (Cressie and Johannesson, 2008; Nychka et al., 2015). For each scale, a (unique) set of knot locations is determined, and a separate set of basis function parameters (e.g., range parameter ϕ, polynomial order N) can be used for each scale. Higher resolutions (finer scales) require more knots. It is common to approximately quadruple the number of knots used for each successive scale (Nychka et al., 2015), though this is not required. When multiple scales are used, if the spatial penalty is utilized, then the penalties are computed for

Table 4: Spatial difference penalty for locations in Figure 2. Zero entries are left blank.

Difference order	Index	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9	c_{10}	c_{11}	c_{12}
$m = 1$	1	2	-1		-1								
	2	-1	3	-1		-1							
	3		-1	2			-1						
	4	-1			3	-1		-1					
	5		-1		-1	4	-1		-1				
	6			-1		-1	3			-1			
	7				-1			3	-1		-1		
	8					-1		-1	4	-1		-1	
	9						-1		-1	3			-1
	10							-1			2	-1	
	11								-1		-1	3	-1
	12									-1		-1	2
$m = 2$	1	4	-4	1	-4	2		1					
	2	-3	6	-3	2	-5	2		1				
	3	1	-4	4		2	-4			1			
	4	-3	2		6	-5	1	-4	2		1		
	5	2	-4	2	-4	8	-4	2	-5	2		1	
	6		2	-3	1	-5	6		2	-4			1
	7	1			-4	2		6	-5	1	-3	2	
	8		1		2	-5	2	-4	8	-4	2	-4	2
	9			1		2	-4	1	-5	6		2	-3
	10				1			-4	2		4	-4	1
	11					1		2	-5	2	-3	6	-3
	12						1		2	-4	1	-4	4

each scale independently, and concatenated block diagonally. More formally, if radial basis functions are computed at g scales and $\mathbf{P}_s^i$ denotes the penalty for radial basis scale i, then $\mathbf{P}_s = \text{diag}(\mathbf{P}_s^1, \ldots, \mathbf{P}_s^g)$.

The support parameter ϕ defining the radial basis function (cf. Equation (7)) should be chosen to overlap at least some of the observed data coordinates. Though it is not required, the ϕ parameter is typically constant for the basis functions of a specific resolution. In order to avoid numerical instability, we recommend choosing ϕ to be at least twice the largest distance between each knot and its closest observed data coordinate. If the support of a radial basis function associated with a knot does not overlap any observed data coordinates, the knot should be removed before performing analysis. Conversely, a range parameter that is too large can also result in computational instability because columns of the resulting basis matrix $\mathbf{B}$ will have highly correlated columns.

The smoothness parameter N of the Wendland covariance function should generally increase when the smoothness of the spatial process increases. It is common to consider values of N in integer and half-integer steps. However, the condition number associated with Wendland covariance functions can quickly increase with N, so any gains potential gains from modeling a smoother process may be offset by numerical instability (Chernih et al., 2014). Additionally, there is perhaps no advantage to using different smoothing parameters for different spatial scales as the overall smoothness of the model will be driven by the finest scale.

We emphasize that the Wendland covariance function does not need to be used to define the radial basis functions used for this method, though it

has some attractive properties. However, it is highly recommended to choose compactly-supported radial basis functions in order to preserve sparsity in matrix operations.

3.5. Complete steps of spatio-temporal sandwich smoother

We now fully outline the algorithm for implementing the STSS. The algorithm is written using general notation so that it extends naturally to the OSS, with the exception that the "Initial smoothing preparation" would only be related to constructing the B-splines for each dimension of $\mathbf{Y}$. The STSS and the OSS have both been fully implemented in the `hero` R package.

Initial smoothing preparation

1. Determine the boundary polygon for the spatial study area.

2. Specify the spatial knot locations for each spatial scale. This can be done with minimal user input in the `hero` package.

3. Specify the Wendland covariance function parameters for each set of radial basis functions and evalulate the spatial basis functions at each of the observed spatial locations.

4. Specify the temporal knot locations/determine the number of B-spline basis functions for the temporal dimension.

5. Specify the B-spline basis parameters and evaluate the B-spline basis function at each of the observed time points.

Assembling spline information

Let $\mathbf{B}_i$ and $\mathbf{P}_i$ be the evaluated basis functions and associated penalty matrix for dimension i, $i = 1, 2, \ldots, d$. Let $\text{diag}(\cdot)$ denote the diagonal matrix based on the elements provided in $(\cdot)$ and c_i denote the number of columns of $\mathbf{B}_i$.

1. Let $\mathbf{V}_i \text{diag}(\mathbf{e}_i) \mathbf{V}_i^T$ be the eigen decomposition of $\mathbf{B}_i^T \mathbf{B}_i$, where $\mathbf{V}_i$ denotes the eigenvectors of the decomposition and $\mathbf{e}_i := (e_{1i}, \ldots, e_{c_i i})$ is the vector of eigenvalues.

2. Compute $\mathbf{Q}_i := \mathbf{V}_i \text{diag}(\mathbf{e}_i^{-1/2}) \mathbf{V}_i^T$ for $i = 1, 2, \ldots, d$, where $\mathbf{e}_i^{-1/2}$ denotes the vector $(e_1^{-1/2}, \ldots, e_{c_i}^{-1/2})$.

3. Let $\mathbf{U}_i \text{diag}(\mathbf{u}_i) \mathbf{U}_i^T$ be the eigen decomposition of $\mathbf{Q}_i^T \mathbf{P}_i \mathbf{Q}_i$, where $\mathbf{u}_i := (u_{1i}, \ldots, u_{c_i i})$ are the eigen values of the decomposition and $\mathbf{U}_i$ are the eigen vectors.

4. Compute $\mathbf{A}_i := \mathbf{B}_i \mathbf{Q}_i \mathbf{U}_i$ for $i = 1, 2, \ldots, d$.

Prepare the data

1. Compute $\tilde{\mathbf{Y}} := \text{RH}(\mathbf{A}_d, \text{RH}(\mathbf{A}_{d-1}, \cdots, \text{RH}(\mathbf{A}_1, \mathbf{Y}) \cdots))$, where $\text{RH}(\mathbf{A}, \mathbf{Y})$ is a rotation of the H-transform of the array $\mathbf{Y}$ by a matrix $\mathbf{A}$ (Currie et al., 2006).

2. Compute $\mathbf{y}^T \mathbf{y} := \sum y_{i_1 i_2 \cdots i_d}^2$ for $1 \leq i_j \leq n_j$, $1 \leq j \leq d$, where $\mathbf{y} := \text{vec}(\mathbf{Y})$. This is simply the sum of the squared response values.

Enhance the fit

Choose a suitable optimization algorithm to find the optimal penalty parameters $\lambda_1, \ldots, \lambda_d$. For each combination of the penalty parameters:

1. Compute $\tilde{\mathbf{u}}_i := ((1+\lambda_i u_{1i})^{-1}, \ldots, (1+\lambda_i u_{c_i i})^{-1})$ and $\tilde{\mathbf{u}}_i^{1/2} := ((1+\lambda_i u_{1i})^{-1/2} \ldots, (1+\lambda_i u_{c_i i})^{-1/2})$ for $i = 1, \ldots, d$.

2. Compute $\tilde{\mathbf{u}} := \tilde{\mathbf{u}}_d \otimes \cdots \otimes \tilde{\mathbf{u}}_1$ and $\tilde{\mathbf{u}}^{1/2} := \tilde{\mathbf{u}}_d^{1/2} \otimes \cdots \otimes \tilde{\mathbf{u}}_1^{1/2}$.

3. Compute $\tau := \prod_{i=1}^{d} \sum_{j=1}^{c_i} (1+\lambda_i u_{ji})^{-1}$.

4. Construct $\tilde{\mathbf{y}} := \text{vec}(\tilde{\mathbf{Y}})$.

5. The GCV statistic as a function of $\lambda_1, \ldots, \lambda_d$ is then $GCV(\lambda) = [(\tilde{\mathbf{y}}^T \tilde{\mathbf{u}})^2 - 2(\tilde{\mathbf{y}}^T \tilde{\mathbf{u}}^{1/2})^2 + \mathbf{y}^T \mathbf{y}]/(1 - \tau/\prod_{j=1}^{d} n_j)^2$.

Choose the values of $\lambda_1, \ldots, \lambda_d$ that minimize the GCV.

Estimate and smooth

1. Compute $\mathbf{G}_i := \mathbf{Q}_i \mathbf{U}_i \text{diag}(\tilde{\mathbf{u}}_i)$ for $i = 1, 2, \ldots, d$.

2. Compute $\hat{\boldsymbol{\theta}} = \text{RH}(\mathbf{G}_d, \text{RH}(\mathbf{G}_{d-1}, \cdots, \text{RH}(\mathbf{G}_1, \tilde{\mathbf{Y}}) \cdots))$.

3. Compute $\hat{\mathbf{Y}} = \text{RH}(\mathbf{B}_d, \text{RH}(\mathbf{B}_{d-1}, \cdots, \text{RH}(\mathbf{B}_1, \hat{\boldsymbol{\theta}}) \cdots))$.

4. Simulation Study

In order to compare the new methodology to the OSS, we construct a simulated example in such a way that both our new STSS and the OSS can be applied. The synthetic data are designed to mimic realistic data from the NARCCAP. Real data were produced for many time scales (sub-daily, daily, monthly, seasonally, and annually) at a 50 km spatial resolution (approximately 18,000 spatial locations for each time, depending on various factors) over much of the United States, Canada, and northern Mexico. The

output was produced by taking boundary conditions from atmosphere-ocean general circulation models (GCMs) and using the information to downscale climate behavior to a finer resolution using a regional climate model (RCM). Data were produced for the time periods 1971-2000 and 2041-2070, with the future data utilizing the A2 emissions scenario from the Special Report on Emission Scenarios (Nakicenovic et al., 2000). The data are publicly available through the Earth System Grid (Mearns et al., 2007, updated 2014).

For the purpose of our simulation study, we create data similar to maximum daily surface air temperature (C) from 1971-2000 for the ECP2 regional climate model (Juang et al., 1997) forced by the GFDL GCM (GFDL Global Atmospheric Model Development Team, 2004)). This particular data set is produced on a 147×116 grid (17,052 locations) covering much of North America. The observed data are noisy, so they were preprocessed to produce a smooth $\mu(\cdot)$ surface (cf., Equation (1)). The OSS was applied to the 30-year time series observed at each location using 1,050 B-spline basis functions and a P-spline difference penalty with $m = 2$. The smoothing parameter λ was chosen via GCV. The nearest neighbors within 414 km (5% of the maximum distance between spatial locations) of each spatial location were determined, and then the smoothed time series for these neighbors was averaged for each calendar day. This resulted in a smooth spatio-temporal data set on a 147×116 grid observed at 365 days, and corresponds to $\mu(\cdot)$ in Equation (1). The resulting surfaces for the first day of Spring, Summer, Fall, and Winter (March 20th, June 21st, September 22nd, and December 21st) are shown in Figure 3 .

We considered four different data-generating distributions for the error

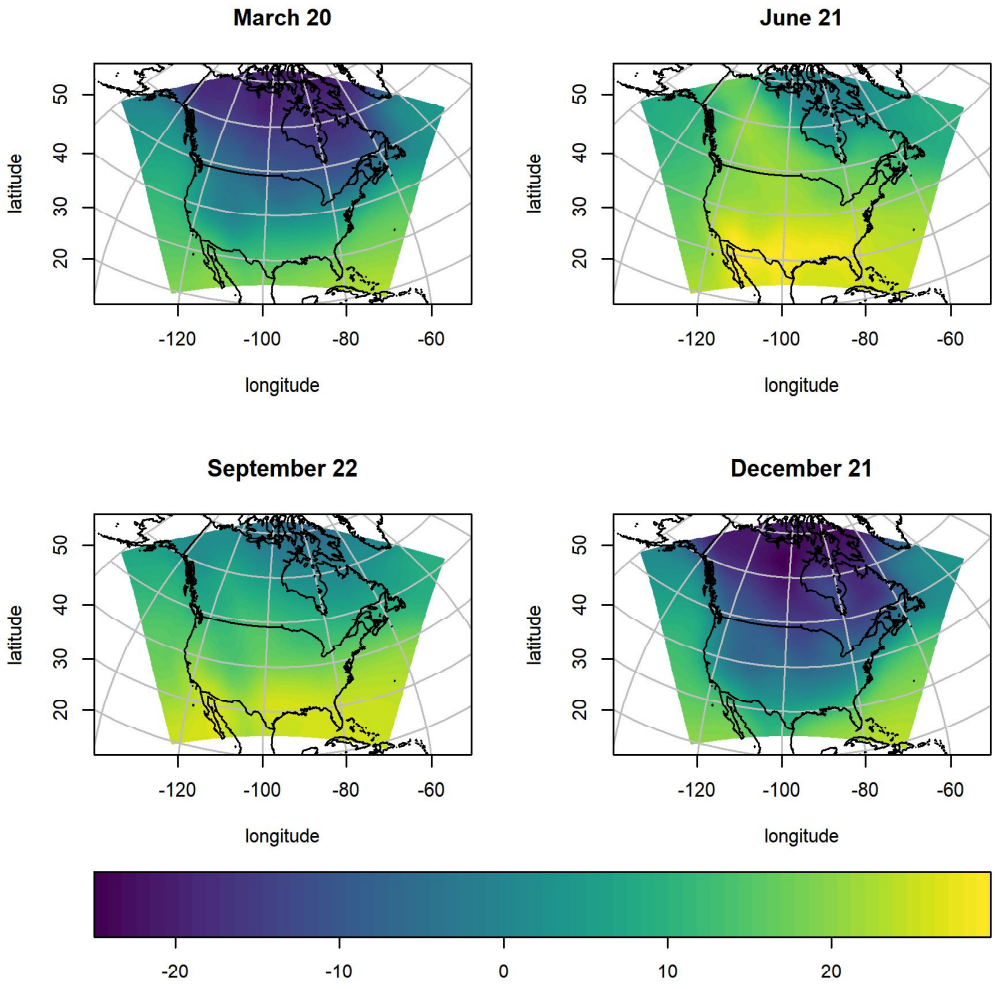

Figure 3: Smoothed maximum daily surface air temperature (C) for several days of the calendar year for the NARCCAP data.

process $\epsilon(\cdot)$ in Equation (1):

A. $\{\epsilon(\mathbf{s}) \overset{\text{i.i.d.}}{\sim} N(0, 3^2), \mathbf{s} \in D\}$.

B. $\{\epsilon(\mathbf{s}) \overset{\text{indep.}}{\sim} N(0, \sigma^2(\mathbf{s})), \mathbf{s} \in D\}$, where $\sigma^2(\mathbf{s})$ is a function that gently varies over space. $\sigma(\mathbf{s})$ was chosen as the standard deviation of the original temperature values at each spatial location. Figure 4 displays a heat map of the $\sigma(\mathbf{s})$ surface over the domain.

C. $\{\epsilon(\mathbf{s}) \overset{\text{i.i.d.}}{\sim} t_8, \mathbf{s} \in D\}$.

D. $\{\epsilon(\mathbf{s}) \overset{\text{i.i.d.}}{\sim} t_4, \mathbf{s} \in D\}$.

Scenario A, in which model/measurement errors are iid normal is fairly standard. Scenario B may be more realistic, as it reflects the fact that temperature variability is larger inland and decreases as we move closer to the coasts. Scenarios C and D are intended to explore how robust the smoothers are to departures from normality. Scenario D is a a bit extreme, and may be unrealistic, as t_4 errors have infinite fourth moment (sample kurtosis would not converge with increasing sample size).

For each data-generating scenario, ten smoothers were applied to each simulated data set; the smoothers are summarized in Table 5. The first two were the OSS with difference penalties of $m = 1$ and 2, respectively. The data are observed on an irregular grid, but the responses were treated as if they were observed on a 147×116 grid over $[0, 1]^2$. For both the u and v dimension, 60 B-spline basis functions spanning $[0, 1]$ were used. For the temporal dimension, 35 B-spline bases were created that spanned the interval $[1, 365]$. The number of basis functions was chosen based on the

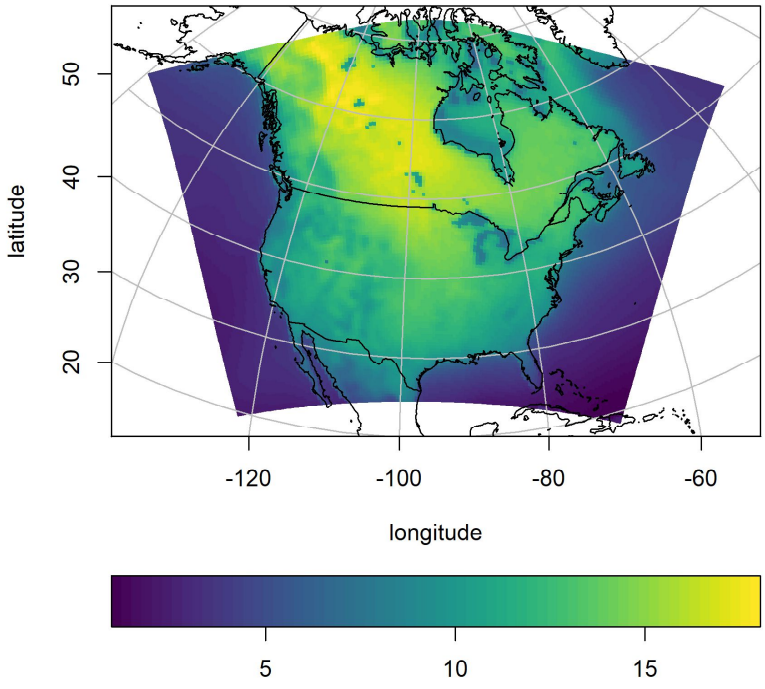

Figure 4: A heat map of the standard deviation surface (C) used for data-generating scenario B.

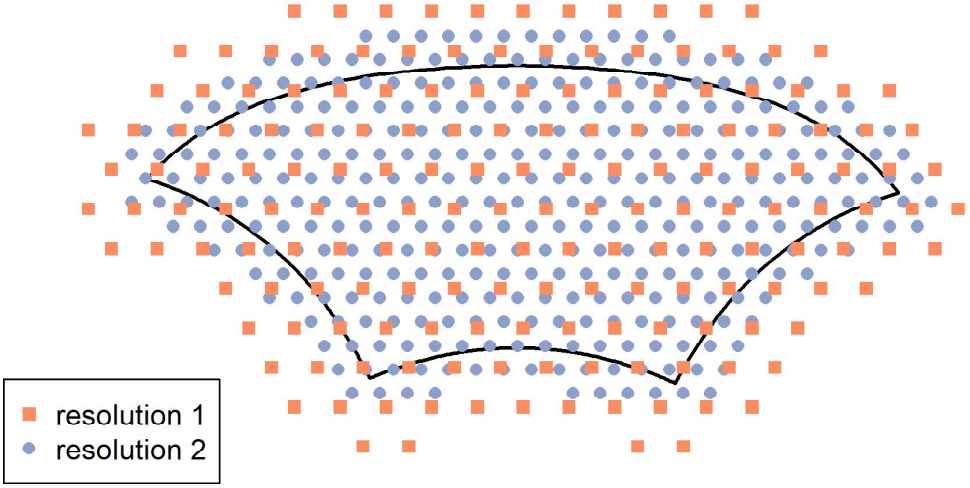

Figure 5: Knots locations for radial basis functions for the first two resolutions.

recommendation of Ruppert et al. (2003). The remaining eight smoothers were the STSS, but with different numbers of basis functions, smoothness parameters, and spatial difference penalties. We considered a larger number of implementations to get an idea how much the new method depends on the possible values of the tuning parameters. The Wendland radial basis functions were constructed using three, four, and five spatial resolutions, respectively. The support, ϕ, for each resolution was chosen to be four times the maximum distance between all knots and their nearest neighbor. The number of knots used in the radial basis functions for the five resolutions were 175, 344, 694, 1,031, and 1,388 knots, respectively. The temporal basis functions used for STSSs were the same as for the OSS. The knot locations for the first two spatial resolutions are shown in Figure 5.

For each simulated data set, the mean squared error (MSE) was com-

Smoother	Spatial basis functions	Penalty
a	2 B-splines with 60 knots each	Standard P-spline with $m = 1$
b	2 B-splines with 60 knots each	Standard P-spline with $m = 2$
c	Wendland basis ($k = 1$) with 1,213 knots at three resolutions	Spatial penalty with $m = 1$
d	Wendland basis ($k = 1$) with 1,213 knots at three resolutions	Spatial penalty with $m = 2$
e	Wendland basis ($k = 1$) with 2,244 knots at four resolutions	Spatial penalty with $m = 1$
f	Wendland basis ($k = 1$) with 2,244 knots at four resolutions	Spatial penalty with $m = 2$
g	Wendland basis ($k = 1$) with 3,632 knots at five resolutions	Spatial penalty with $m = 1$
h	Wendland basis ($k = 1$) with 3,632 knots at five resolutions	Spatial penalty with $m = 2$
i	Wendland basis ($k = 2$) with 1,213 knots at three resolutions	Spatial penalty with $m = 1$
j	Wendland basis ($k = 2$) with 1,213 knots at three resolutions	Spatial penalty with $m = 2$

Table 5: Summary of smoothing methods applied to simulated data. Temporal smoothing is the same for all methods, 35 B-splines with standard P-spline penalty with $m = 2$. Smoothers a and b are implementations of the OSS. Smoothers c-j are implementations of the STSS.

puted, where

$$\text{MSE} := \frac{1}{17,052 \times 365} \sum_{i=1}^{17,052} \sum_{j=1}^{365} [\mu(\mathbf{s}_i, j) - \hat{y}(\mathbf{s}_i, j)]^2. \tag{12}$$

The results of the simulation study are summarized in Figure 6.

The first general observation is that practically all implenations of our STSS have significantly smaller MSEs than implementation a of the OSS, and many have smaller MSEs than implementation b. The latter statement is particularly in scenarios A and B, which correspond to the most commonly encountered normal errors. For data scenarios A, B, and C, the best STSSs have significantly smaller MSE than the best OSSs; their mean is below the lower bound of the 95 percent t-confidence interval for the mean MSE of the original smoothers. In data-generating scenario D, sandwich smoother b had significantly smaller MSE than the best STSS, but errors without finite variance may often be unrealistic. While many more potential smoother variants (in terms of penalty order, number of basis functions, smoothness parameters, etc.) could be considered for both the original and spatio-temporal sandwich smoothers, these results suggest that STSS is at least competitive with the OSS when both can be utilized, and may even perform better in some scenarios. However, we emphasize that a primary goal of the STSS is to build an approach for describing spatio-temporal data when the OSS is inappropriate, not necessarily to improve performance when both approaches are appropriate.

One area where the STSS cannot challenge the OSS is computation time. On the test computer (a Windows 10 laptop with 16 GB of RAM and Intel Core i7-7500U CPU running at 2.70GHz), typical execution time for smooth-

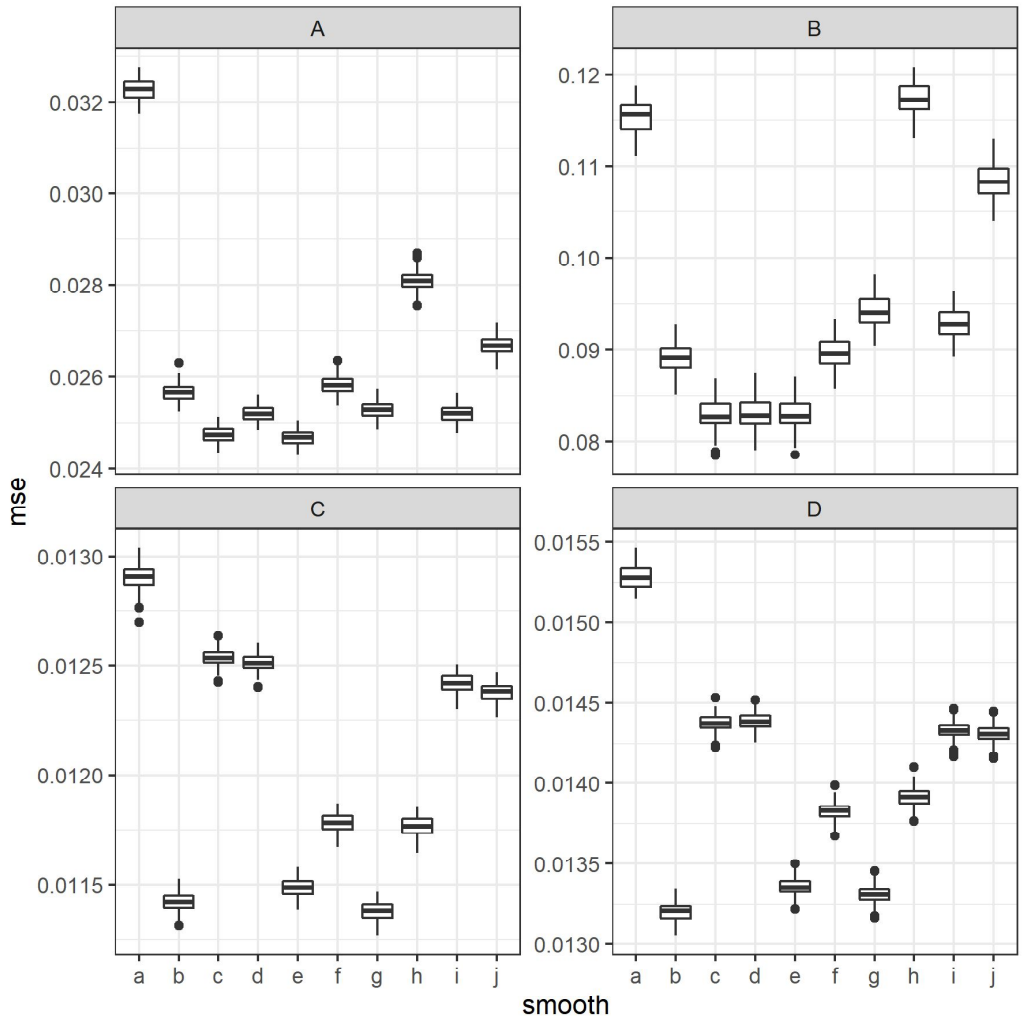

Figure 6: Boxplots of the MSEs (12) for the four error distributions, A, B, C, D, and all implementations listed in Table 5. Methods a and b correspond to the two most natural implementations of the OSS of Xiao et al. (2013).

ing the simulated data sets was about 2 seconds for the OSS and about 30, 76, and 205 seconds for the STSSs with 3, 4, and 5 spatial basis resolutions, respectively. Because eigen decompositions must be performed for the spatial basis functions at all resolutions simultaneously, this results in an eigen decomposition of a 3,632×3,632 matrix for smoothers with the most basis functions, whereas the original sandwich smoother only needed to compute eigen decompositions for a 60×60 matrix. Since the computational complexity of the eigen decomposition typically scales cubically, this increases the execution time of the STSS. However, the STSS is still very efficient computationally, and can be applied to massive spatial time series that are impossible for a standard tensor-product spline (because of computational complexity) or the OSS (because of data structure).

5. Spatial time series examples

We now consider two examples for which it would be impossible to apply the OSS because the spatial locations do not form an array. The spatial region over which the data are defined is the continental United States, shown in Figure 7. We can refer to such data as spatially-indexed time series. We consider two examples: one involving data from the NARCCAP and another from the NA-CORDEX program. The second example emphasizes computational aspects related to a very large data set.

5.1. NARCCAP example

We consider temperature data produced by the NARCCAP. We utilize a combination of models different from those used in Section 4. Specifically, we consider maximum daily surface air temperature (C) simulated

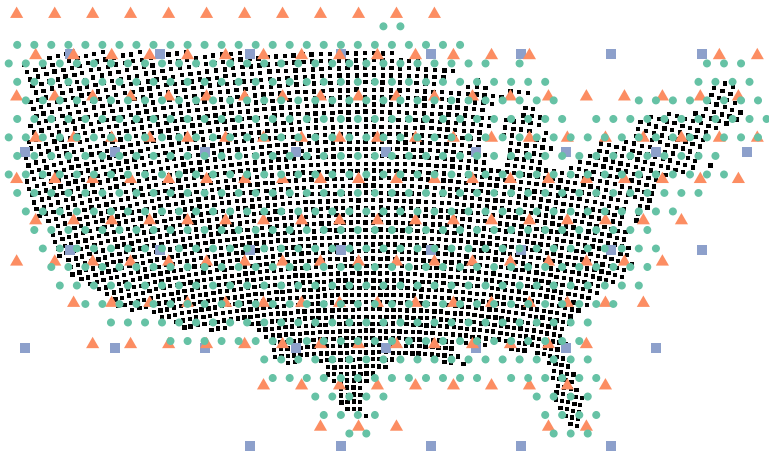

Figure 7: WRFG-CGCM3 locations in the continental United States are indicated by the small black squares. The knot locations for the lowest resolution basis functions are shown by larger blue squares, by orange triangles for the middle resolution, and green circles for the highest resolution.

for the time period January 1, 2041 to December 31, 2041 produced by the WRFG regional climate model (Skamarock et al., 2005) forced by the CGCM3 atmosphere-ocean general circulation model (Flato et al., 2000) within the continental United States. This results in time series observed at 2,948 spatial locations over 365 days, for a total of 1,079,305 total response values. The black dots in in Figure 7 indicate the observed data locations.

We now consider application of the STSS to these data. We constructed radial basis functions using the Wendland covariance function at three different resolutions using 45, 170, and 684 knots, respectively (899 total knots). The support at each resolution was designed to be at least 3 times as far as

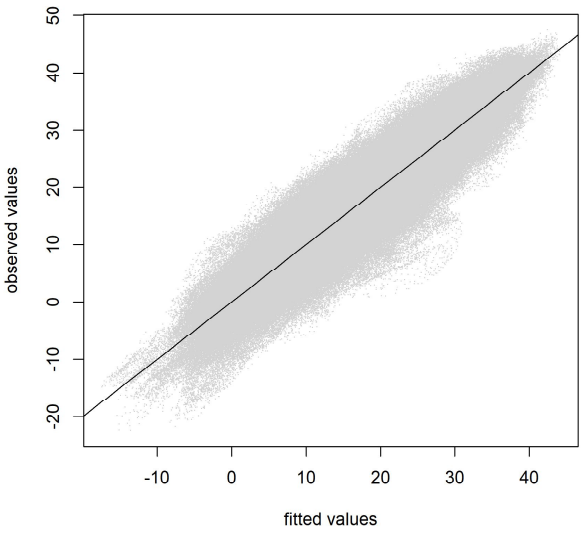

Figure 8: A scatterplot of the observed maximum daily surface air temperature (C) versus the fitted values after applying the STSS to the WRFG-CGCM3 at locations shown in Figure 7.

the nearest neighbor of each knot location, and the Wendland polynomial order was set to $N = 2$. The knot locations for each resolution are shown by differently colored and shaped symbols in Figure 7. The spatial difference penalty of order $m = 2$ was used. As in Section 4, 35 B-spline basis functions were used for the temporal dimension. On the test computer, the smoothing (preparing the splines, optimizing the penalty parameters, estimating the coefficients, etc.) took a little less than 5 seconds. Figure 8 provides a scatterplot of the observed versus fitted values. The plot shows the desired linear relationship between the variables.

5.2. NA-CORDEX example

The NA-CORDEX program is the North American component of the CORDEX program sponsored by the World Climate Research Program. It is a successor to the NARCCAP. The NA-CORDEX program has a similar spatial domain to the NARCCAP, but produces data at a higher resolution (25 km versus 50 km) and for a much longer time period (NA-CORDEX: 1950-2100, NARCCAP: 1971-2000 and 2041-2070). The NA-CORDEX program plans to produce climate data for a combination of 6 GCMs from the CMIP5 archive (Taylor et al., 2012) and 7 RCMs, though only a subset is currently available. Future climate scenarios are run under Representation Concentration Pathways (RCP) 4.5 and 8.5 adopted by the IPCC for its fifth Assessment Report (AR5, Pachauri et al., 2014) in 2014. We consider smoothing the daily maximum surface air temperature (C) of data downscaled by the CanRCM4 RCM (Scinocca et al., 2016) forced by the CanESM2 GCM Chylek et al. (2011) available as of September 12, 2018 through the Earth System Grid (Mearns et al., 2017). We consider only the RCP 8.5 Scenario for the future climate data. The domain of the observed data locations is shown in Figure 9.

The smoothing process was complicated by the massive size of the original data. The original data were available as two separate files in NetCDF format with sizes of 11.9 and 19.9 GB, respectively. The data could not be read into memory directly. However, the NetCDF format is a serialized data storage format, allowing a researcher to read a subset of the data into memory one section at a time.

The spatial data locations were determined and a polygon border for these

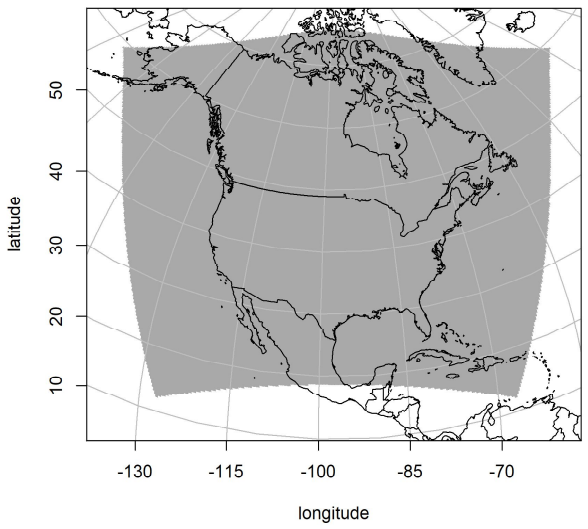

Figure 9: The domain of the observed NA-CORDEX locations is shown in dark grey.

locations was manually constructed. Using this polygon, Wendland order $N = 1$ radial basis functions were constructed at four different resolutions and a total of 2,243 knot locations. The overlap of the basis functions at each resolution was roughly 4 times the distance between each knot and its nearest neighbor. The temporal basis functions were composed of 35 B-spline basis functions spread over the interval $[1, 365]$. Necessary spline-related information was then assembled for the STSS, taking roughly 3 minutes, 45 seconds on the test computer.

Having specified the basis functions, the data were prepared for the STSS. Specifically, the data for each year were read from file and then transformed using spline-related matrices via the process outlined in Section 3.5. The transformed data could be stored using only 86.5 Mb in the gzip compres-

sion format. The process of reading and preparing the data took under 11 minutes. The optimal penalty parameters were then chosen based on the minimal GCV statistic. This step utilized a sequence of grid searches and optimization routines, taking less than 1 minute to complete. Lastly, estimating the regression coefficients and computing the fitted values took about 5 minutes, 15 seconds.

The total amount of time it took to process the nearly 32 GB of data on the test machine was around 20 minutes. Additionally, the smoothed data can be easily reconstructed from the set of evaluated basis functions and the estimated coefficients, which can be stored in gzip compressed files of size 86.2 MB. Figure 10 displays a scatterplot of the observed temperatures versus the smoothed data, confirming that the STSS has captured the overall pattern of the original temperature data.

6. Discussion

We have developed a spatio-temporal sandwich smoother (STSS) that builds on the original sandwich smoother (OSS) proposed by Xiao et al. (2013). In contrast to the OSS, the STSS treats the spatial and temporal dimensions distinctly, modeling the spatial dimension using compactly-supported Wendland covariance-based radial basis functions and the temporal dimension using B-splines. Additionally, we have proposed a spatial difference penalty more natural for controlling the variation of the spatial coefficients over the study domain and that generalizes to basis functions not observed on a grid. While the OSS can only be applied to data on a rectilinear grid, the STSS can be applied to irregularly gridded or non-gridded

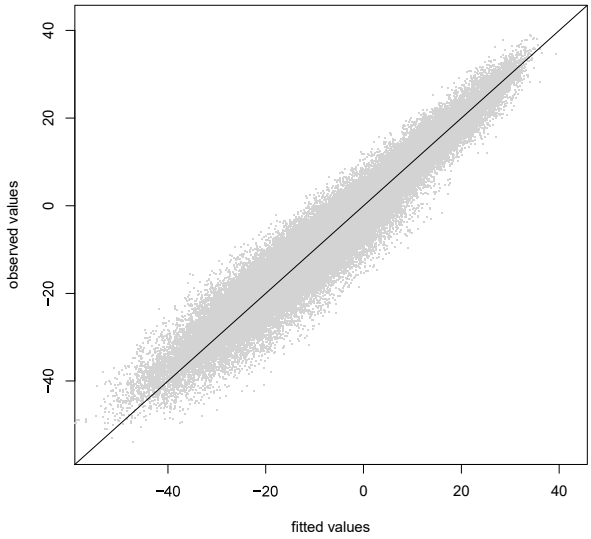

Figure 10: A scatterplot comparing the observed maximum daily surface air temperature (C) versus the smoothed fitted values in the example of Section 5.2. A random sample of 1,000 values from each year were selected for plotting.

spatial time series, which are more common for the spatio-temporal data sets motivating this research. The STSS was used to process and represent nearly 32 GB of daily temperature data observed over 150 years in around 20 minutes on a 16 GB Windows laptop, demonstrating its ability to smooth massive spatio-temporal data sets.

Our main takeaway from simulations in Section 4 is that the STSS is competitive with the OSS when data are on a rectilinear grid. However, the OSS is much more computationally efficient because the required eigen decompositions are for substantially smaller matrices (e.g., 60 versus 3,600 in the simulation studies). Thus, if the data are on a rectilinear grid, it is difficult to claim that the STSS provides a large improvement over the OSS, the difference may depend more on tuning parameters than the method used. However, when data are spatial time series observed on an irregular grid or on a set of non-gridded spatial locations, then the OSS is inappropriate while the STSS provides an effective, computationally efficient approach for smoothing the data.

A weakness of both the OSS and STSS is that they require complete data with no missing values. This is necessitated by the representation of the data as a complete array $\mathbf{Y}$. If the data are nearly complete, then imputing needed values to create a complete data will negligibly impact the fitted results. When the observed data are not on a rectilinear grid, Xiao et al. (2013) recommended creating a rectilinear "gridded" data set by averaging nearby points or imputing missing values.

Estimators for the variance of $\hat{\boldsymbol{\theta}}$ and $\hat{\mathbf{y}}$ can be derived, but it is not clear that they are computable. The solution for $\boldsymbol{\theta}$ from the penalized residual sum

of squares in Equation (2) is $\hat{\boldsymbol{\theta}} = (\mathbf{B}^T\mathbf{B} + \mathbf{P}_\lambda)^{-1}\mathbf{B}^T\mathbf{y}$, so $\text{var}(\hat{\boldsymbol{\theta}}) = (\mathbf{B}^T\mathbf{B} + \mathbf{P}_\lambda)^{-1}\mathbf{B}^T\text{var}(\mathbf{y})\mathbf{B}(\mathbf{B}^T\mathbf{B}+\mathbf{P}_\lambda)^{-1}$, noting that $\mathbf{P}_\lambda$ is symmetric. The size of the inverse matrices would likely preclude this computation. Similarly, starting in Equation (4), $\text{var}(\hat{\mathbf{y}}) = \mathbf{S}_\lambda\text{var}(\mathbf{y})\mathbf{S}_\lambda$, noting that $\mathbf{S}_\lambda$ is symmetric. Once again, the size of these matrices likely precludes the variance computation.

The computational complexity of the STSS is dominated by two pairs of eigen decompositions, discussed in more detail in Section 3.5, while the other relevant computations are simple matrix products. Specifically, the STSS requires the eigen decomposition of $\mathbf{B}_i^T\mathbf{B}_i$ (the cross-product of the basis functions of dimension i) and $\mathbf{Q}_i^T\mathbf{P}_i\mathbf{Q}_\mathbf{i}$ (where $\mathbf{Q}_i$ is computed from a matrix product and $\mathbf{P}_i$ is the penalty associated with dimension i). Both sets of matrix products are of size $c_i \times c_i$, where c_i is the number of knots needed in dimension i. Let c_{STSS} be the maximum number of knots associated with either the spatial or temporal dimension. Since eigen decomposition algorithms typically have cubic order complexity based on the size of the matrix being decomposed, the overall complexity of our approach should be $O(c_{\text{STSS}}^3)$. In contrast, the main bottleneck of a standard tensor-product smoother is the inversion of a matrix of size $c_1c_2c_3 \times c_1c_2c_3$, where each c_i is the number number of knot location in each dimension. Let $c_{\text{STD}} = c_1c_2c_3$. Since matrix inversion (or more accurately, a linear solve involving that matrix) typically has cubic order computational complexity, the overall computational complexity of the standard smoother should be $O(c_{\text{STD}}^3)$, with $c_{\text{STD}} >> c_{\text{STSS}}$.

Several aspects of the STSS warrant further investigation. A limitation of the STSS is that matrix decompositions for the radial basis matrices take

substantially longer than the corresponding decompositions on the multivariate P-splines used by the OSS. One possibility for improving the speed of the STSS would be constructing the radial basis functions at each resolution such that they are orthogonal to one another. In that case, the necessary decompositions could be done for each resolution separately, which would increase the speed of the algorithm. A single parameter controls the penalty for the spatial bases of the STSS across all resolutions. Orthogonal bases across resolutions would allow for each resolution to be penalized separately. Assuming non-orthogonal bases between resolutions, the penalty parameter is essentially controlled by the finest spatial resolution. The presentation of the STSS in this manuscript assumed isotropic, stationary Wendland covariance basis functions, which may not be appropriate. No mathematical principle is violated by using nonstationary or anisotropic covariance functions, but it is not clear how one would implement such a change in practice.

The examples presented in this manuscript all involved data with relatively dense data locations, and basis function knots were distributed in a fairly regular pattern. For spatial locations distributed more heterogeneously it would be appropriate to place fewer knots in regions with fewer observed data and more knots in regions with more observed data. Additionally, it may be appropriate to place more knots in regions or at times with greater response variability. The bivariate splines over triangulations proposed by Lai and Wang (2013) may provide an alternative approach to the spatial smoothing aspect of this problem. This may reduce the required number of knots (improving computational efficiency). Additionally, Lai and Wang state that their approach, "... is a good choice when data are located in

domains with complex boundaries and/or possible holes", which aligns well with the context we consider. Another important consideration is how to elicit parameter values that may not be easy to optimize over, such as the smoothness parameter N, the difference penalty m, or even the number of spatial resolutions for which to construct bases. The STSS allows for cross-validation assuming one selects folds with respect to spatial locations and/or time steps that allows for the data to be represented in array form. This facilitates a standard approach for choosing between models.

All analyses have been executed in Microsoft R Open 3.5.3 (Core Team, 2019; Microsoft, 2019). Code for all analyses in this paper are available as Supplementary Material. Due to their size, interested researchers must directly download the NARCCAP and NA-CORDEX data NetCDF files from the NCAR Climate Gateway (`https://www.earthsystemgrid.org/`), with additional information in the provided code. Both the OSS and STSS are implemented in the `hero` R package, which is provided in the supplementary material.

Declarations of interest

None

Acknowledgments

The authors gratefully acknowledge partial support by NSF. J. French was partially supported by NSF awards 1463642 and 1915277 and P. Kokoszka was partially supported by NSF awards 1462067 and 1914882. J. French also thanks Subrata Paul for some initial exploration and discussion of functional data smoothing using standard tensor-product splines.

References

Aston, J., Pigoli, D., Tavakoli, S., 2016. Tests for separability in nonparametric covariance operators of random surfaces. The Annals of Statistics 6, 1906–1948.

Chernih, A., Sloan, I.H., Womersley, R.S., 2014. Wendland functions with increasing smoothness converge to a Gaussian. Advances in Computational Mathematics 40, 185–200. doi:10.1007/s10444-013-9304-5.

Chylek, P., Li, J., Dubey, M.K., Wang, M., Lesins, G., 2011. Observed and model simulated 20th century Arctic temperature variability: Canadian earth system model CanESM2. Atmospheric Chemistry and Physics Discussions 8, 22893–22907.

Constantinou, P., Kokoszka, P., Reimherr, M., 2017. Testing separability of space-time functional processes. Biometrika 104, 425–437.

Craven, P., Wahba, G., 1978. Smoothing noisy data with spline functions. Numerische Mathematik 31, 377–403. doi:10.1007/BF01404567.

Cressie, N., Johannesson, G., 2008. Fixed rank kriging for very large spatial data sets. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70, 209–226. doi:10.1111/j.1467-9868.2007.00633.x.

Currie, I.D., Durban, M., Eilers, P.H.C., 2006. Generalized linear array models with applications to multidimensional smoothing. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 68, 259–280. doi:10.1111/j.1467-9868.2006.00543.x.

Eilers, P.H., Currie, I.D., Durbán, M., 2006. Fast and compact smoothing on large multidimensional grids. Computational Statistics & Data Analysis 50, 61–76.

Eilers, P.H., Marx, B.D., Durbán, M., 2015. Twenty years of P-splines. SORT: statistics and operations research transactions 39, 0149–186. URL: https://www.raco.cat/index.php/SORT/article/view/302258/391947.

Eilers, P.H.C., Marx, B.D., 1996. Flexible smoothing with B-splines and penalties. Statistical Science 11, 89–121. doi:10.1214/ss/1038425655.

Flato, G.M., et al., 2000. The Canadian centre for climate modeling and analysis global coupled model and its climate. Climate Dynamics 16, 451–467. doi:10.1007/s003820050339.

French, J., Kokoszka, P., Stoev, S., Hall, L., 2019. Quantifying the risk of heat waves using extreme value theory and spatio-temporal functional data. Computational Statistics and Data Analysis 131, 176–193.

French, J.P., 2017. autoimage: Multiple Heat Maps for Projected Coordinates. The R Journal 9, 284–297. URL: https://journal.r-project.org/archive/2017/RJ-2017-025/index.html.

GFDL Global Atmospheric Model Development Team, 2004. The new GFDL global atmospheric and land model AM2-LM2: Evaluation with prescribed SST simulations. Journal of Climate 17, 4641–4673.

Giorgi, F., Jones, C., Asrar, G.R., et al., 2009. Addressing climate information needs at the regional level: the CORDEX framework. World Meteorological Organization (WMO) Bulletin 58, 175.

Gneiting, T., 2002. Compactly supported correlation functions. Journal of Multivariate Analysis 83, 493–508. doi:10.1006/jmva.2001.2056.

Gromenko, O., Kokoszka, P., Reimherr, M., 2017a. Detection of change in the spatiotemporal mean function. Journal of the Royal Statistical Society (B) 79, 29–50.

Gromenko, O., Kokoszka, P., Sojka, J., 2017b. Evaluation of the cooling trend in the ionosphere using functional regression with incomplete curves. The Annals of Applied Statistics 11, 898–918.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer.

Juang, H.M.H., Hong, S.Y., Kanamitsu, M., 1997. The NCEP regional spectral model: an update. Bulletin of the American Meteorological Society 78, 2125–2143.

Jurek, M., Katzfuss, M., 2018. Multi-resolution filters for massive spatio-temporal data. arXiv preprint:1810.04200 .

Lai, M.J., Wang, L., 2013. Bivariate penalized splines for regression. Statistica Sinica 23, 1399–1417. doi:10.5705/ss.2010.278.

Lila, E., Aston, J.A.D., Sangalli, L.M., 2016. Smooth Principal Component

Analysis over two-dimensional manifolds with an application to neuroimaging. Ann. Appl. Stat. 10, 1854–1879. doi:10.1214/16-AOAS975.

Liu, C., Ray, S., Hooker, G., 2017. Functional principal components analysis of spatially correlated data. Statistics and Computing 27, 1639–1654.

Ma, P., Kang, E.L., 2019. Spatio-Temporal data fusion for massive sea surface temperature data from MODIS and AMSR-E instruments. Environmetrics 0, e2594. doi:10.1002/env.2594.

Mearns, L., et al., 2007, updated 2014. The North American regional climate change assessment program dataset, National Center for Atmospheric Research Earth System Grid data portal, Boulder, CO. doi:10.5065/D6RN35ST.

Mearns, L., et al., 2017. The NA-CORDEX dataset, version 1.0. doi:10.5065/D6SJ1JCH. NCAR Climate Data Gateway, Boulder CO. Accessed October 12, 2019.

Mearns, L.O., Arritt, R., Biner, S., Bukovsky, M.S., McGinnis, S., Sain, S., Caya, D., Correia Jr, J., Flory, D., Gutowski, W., et al., 2012. The North American regional climate change assessment program: overview of phase I results. Bulletin of the American Meteorological Society 93, 1337–1362.

Mearns, L.O., Gutowski, W., Jones, R., Leung, R., McGinnis, S., Nunes, A., Qian, Y., 2009. A regional climate change assessment program for North America. EOS, Transactions American Geophysical Union 90, 311–311.

Microsoft, 2019. Microsoft R Open. Microsoft. Redmond, Washington. URL: https://mran.microsoft.com/.

Nakicenovic, N., Alcamo, J., Grubler, A., Riahi, K., Roehrl, R., Rogner, H.H., Victor, N., 2000. Special report on emissions scenarios (SRES), a special report of Working Group III of the intergovernmental panel on climate change. Cambridge University Press.

Nychka, D., Bandyopadhyay, S., Hammerling, D., Lindgren, F., Sain, S., 2015. A multiresolution Gaussian process model for the analysis of large spatial datasets. Journal of Computational and Graphical Statistics 24, 579–599. doi:10.1080/10618600.2014.914946.

O'Sullivan, F., 1986. A statistical perspective on ill-posed inverse problems. Statist. Sci. 1, 502–518. doi:10.1214/ss/1177013525.

Pachauri, R.K., Allen, M.R., Barros, V.R., Broome, J., Cramer, W., Christ, R., Church, J.A., Clarke, L., Dahe, Q., Dasgupta, P., et al., 2014. Climate change 2014: synthesis report. Contribution of Working Groups I, II and III to the fifth assessment report of the Intergovernmental Panel on Climate Change. IPCC.

Ramsay, J., Hooker, G., Graves, S., 2009. Functional Data Analysis with R and MATLAB. Springer.

Ramsey, J.R.J., Silverman, B.W., 2005. Functional Data Analysis, 2nd edition. Springer-Verlag New York. doi:10.1007/b98888.

Reed, D.M., Yagel, R., Law, A., Shin, P.W., Shareef, N., 1996. Hardware assisted volume rendering of unstructured grids by incremental slicing, in: Proceedings of the 1996 Symposium on Volume Visualization, IEEE Press, Piscataway, NJ, USA. pp. 55–ff.

Reinsch, C.H., 1967. Smoothing by spline functions. Numerische Mathematik 10, 177–183. doi:10.1007/BF02162161.

Rue, H., Held, L., 2005. Gaussian Markov Random Fields: Theory and Applications. volume 104 of *Monographs on Statistics and Applied Probability*. Chapman & Hall, London.

Ruppert, D., Wand, M.P., Carroll, R.J., 2003. Semiparametric Regression. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press. doi:10.1017/CBO9780511755453.

Schabenberger, O., Gotway, C., 2005. Statistical Methods for Spatial Data Analysis. Chapman & Hall/CRC Texts in Statistical Science, Taylor & Francis, Boca Raton FL.

Scinocca, J.F., Kharin, V.V., Jiao, Y., Qian, M.W., Lazare, M., Solheim, L., Flato, G.M., Biner, S., Desgagne, M., Dugas, B., 2016. Coordinated global and regional climate modeling. Journal of Climate 29, 17–35. doi:10.1175/JCLI-D-15-0161.1.

Skamarock, W.C., Klemp, J.B., Dudhia, J., Gill, D.O., Barker, D.M., Wang, W., Powers, J.G., 2005. A description of the advanced research WRF version 2. Technical Report. National Center for Atmospheric Research. NCAR Technical Note NCAR/TN-468+STR.

Taylor, K.E., Stouffer, R.J., Meehl, G.A., 2012. An overview of CMIP5 and the experiment design. Bulletin of the American Meteorological Society 93, 485–498. doi:10.1175/BAMS-D-11-00094.1.

Core Team, R., 2019. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing. Vienna, Austria. URL: https://www.R-project.org/.

Waller, L., Gotway, C., 2004. Applied Spatial Statistics for Public Health Data. Wiley Series in Probability and Statistics, Wiley, Hoboken NJ.

Wendland, H., 1995. Piecewise polynomial, positive definite and compactly supported radial functions of minimal degree. Advances in Computational Mathematics 4, 389–396. doi:10.1007/BF02123482.

Wood, S., 2017. Generalized Additive Models: An Introduction with R, Second Edition. Chapman & Hall/CRC Texts in Statistical Science, CRC Press, Boca Raton FL.

Xiao, L., Li, Y., Ruppert, D., 2013. Fast bivariate P-splines: the sandwich smoother. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 75, 577–599. doi:10.1111/rssb.12007.

Zhu, H., Versace, F., Cinciripini, P.M., Rausch, P., Morris, J.S., 2018. Robust and Gaussian spatial functional regression models for analysis of event-related potentials. NeuroImage 181, 501–512. doi:10.1016/j.neuroimage.2018.07.006.