ORIGINAL PAPER

Spatio-temporal change of support modeling with R

Andrew M. Raim[1] · **Scott H. Holan**[2,3] · **Jonathan R. Bradley**[4] ·
Christopher K. Wikle[2]

Abstract

Spatio-temporal change of support methods are designed for statistical analysis on spatial and temporal domains which can differ from those of the observed data. Previous work introduced a parsimonious class of Bayesian hierarchical spatio-temporal models, which we refer to as STCOS, for the case of Gaussian outcomes. Application of STCOS methodology from this literature requires a level of proficiency with spatio-temporal methods and statistical computing which may be a hurdle for potential users. The present work seeks to bridge this gap by guiding readers through STCOS computations. We focus on the R computing environment because of its popularity, free availability, and high quality contributed packages. The stcos package is introduced to facilitate computations for the STCOS model. A motivating application is the American Community Survey (ACS), an ongoing survey administered by the U.S. Census Bureau that measures key socioeconomic and demographic variables for various populations in the United States. The STCOS methodology offers a principled approach to compute model-based estimates and associated measures of uncertainty for ACS variables on customized geographies and/or time periods. We present a detailed case study with ACS data as a guide for change of support analysis in R, and as a foundation which can be customized to other applications.

Keywords American Community Survey · Areal data · Basis functions · Bayesian statistics · Model-based estimates · Official statistics

1 Introduction

In the course of an analysis where data are inherently spatio-temporal, an investigator may desire estimates on spatial and/or temporal domains not coinciding exactly with domains of the observations. This can include customized geographies and time

✉ Andrew M. Raim
andrew.raim@census.gov

Extended author information available on the last page of the article

periods conceived long after the data have been collected. Spatio-temporal change of support methods aim to provide this capability. A methodology recently proposed by Bradley et al. (2015b) captures spatio-temporal dependencies in areal data by constructing several key matrices which become the foundation of a Bayesian hierarchical model. Model fitting is done via Markov chain Monte Carlo (MCMC); in particular, the model permits a Gibbs sampler which is conveniently composed of draws from standard distributions. Estimates, predictions, and appropriate measures of uncertainty are provided by the fitted model. This methodology, hereafter referred to as the *STCOS model* or *STCOS methodology*, is the focus of the present paper. Although STCOS methodology has been fully specified by Bradley et al. (2015b), potential users—such as subject-domain scientists who may not be experts in spatio-temporal statistics—may find proceeding from the previous literature to their own applications to be a difficult hurdle. A successful implementation requires managing datasets containing estimates, geospatial data, operations on sparse matrices, Bayesian computing, plotting, as well as carrying out computations tailored to the STCOS model.

In this paper, we demonstrate an assortment of tools to perform STCOS modeling through a detailed case study, with the objective of making the methodology more accessible to potential users. The required tasks can be accomplished with a variety of modern computing platforms, but we will focus on R, the popular open source environment for statistical computing (R Core Team 2020). R is supported by an active community of academic, corporate, and individual users. A large and diverse collection of packages has been contributed by its community and published to repositories such as the Comprehensive R Archive Network (CRAN). Much of R, including the base platform and CRAN packages, is freely available on the internet. The high-level R programming language facilitates data analysis, fast prototyping of new methods, and simulation, and can be augmented with C, C++, and FORTRAN when speed or efficient use of memory are crucial. In addition to highlighting some established R packages, we introduce the stcos package to handle some of the more intricate STCOS computations in an efficient and user-friendly way. Familiarity with R will be assumed throughout the remainder of the paper.

Bradley et al. (2015b) developed STCOS with a motivating application to the American Community Survey (ACS), an ongoing survey administered by the U.S. Census Bureau for the purpose of measuring key socioeconomic and demographic variables for the U.S. population. ACS is again showcased in the present paper as it remains an important application for change of support methods (Weinberg et al. 2018). However, STCOS methodology is not limited to applications involving the ACS or the U.S. Census Bureau. The problem of spatial change of support has arisen in atmospheric science and oceanography (Wikle and Berliner 2005), water quality modeling (Rode et al. 2010), environmental health (Fuentes et al. 2006), and remote sensing (Nguyen et al. 2012), among others. See Gotway and Young (2002), Bradley et al. (2015b), and the references therein for a review of the change of support literature.

Existing software tools for change of support appear to originate from geographic information systems (GIS) literature, emphasizing methods such as pycnophylactic interpolation (Tobler 1979), areal-weighted interpolation (Lam 1983), and dasymetric mapping (e.g. Eicher and Brewer 2001). In R, such tools include the pycno package (Brunsdon 2014), the st_interpolate_aw function in the sf package (Pebesma

2018), and the `areal` package (Prener and Revord 2019). The `Tobler` package
was developed for `Python` by Cortes et al. (2019). Qiu et al. (2012) and Mileu and
Queirós (2018) describe adding change of support capabilities to the `ArcGIS` and
`QGIS` platforms, respectively. From the perspective of software tools, the present work
offers two major contributions: (1) measures of uncertainty expressed via a statistical
model, and (2) the capability to carry out change of support in both space and time.
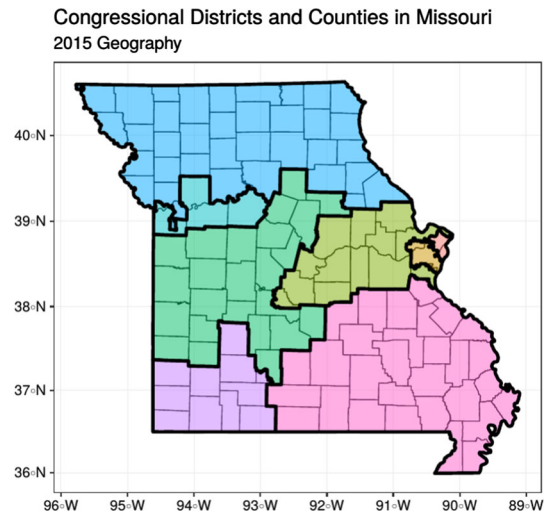
The remainder of the article proceeds as follows. Section 2 discusses STCOS con-
cepts in the context of the ACS. Section 3 reviews STCOS methodology. Here some
additional details are provided—and some small modifications are made—from the
original formulation of Bradley et al. (2015b). Section 4 discusses the set of R tools
to be demonstrated, including basic functionality of the `stcos` package. Section 5
presents our case study to demonstrate STCOS programming; we produce model-
based estimates of median household income for several neighborhoods in the City of
Columbia in Boone County, Missouri. Section 6 concludes the article. This article is
intended to be largely self-contained for a wide range of readers; those eager to begin
programming can focus primarily on Sects. 4 and 5. The `stcos` package is available
on CRAN at https://CRAN.R-project.org/package=stcos. The complete code for the
City of Columbia data analysis is provided as a supplement to this article.

2 Change of support concepts and the ACS

To facilitate our discussion of the change of support problem and STCOS methodology,
we now give a brief overview of the ACS. Public-use ACS data are available through the
Census Bureau's ACS website (https://www.census.gov/programs-surveys/acs) dat-
ing back to the year 2005. Estimates have historically been released for 1-year, 3-year,
or 5-year periods; 3-year period estimates were discontinued after 2013. The Census
Bureau releases annual ACS period estimates for a variety of geographies including
states, counties, census tracts, and school districts. At their finest geography, data are
released at the census block-group level; however, estimates for an area are suppressed
unless the area meets certain criteria. An area typically must have a population of at
least 65,000 for 1-year estimates to be released, but there is no population require-
ment for 5-year estimates (U.S. Census Bureau 2016). ACS estimates consist of point
estimates and associated measures of uncertainty such as margins of error (MOEs)
corresponding to 90% confidence intervals, or variance estimates; we will refer to them
collectively as direct estimates. Because statistical agencies like the Census Bureau
have direct access to the confidential microdata, special tabulations for new geogra-
phies or period lengths can be prepared internally as needed. However, data users out-
side of the Census Bureau may be interested in custom geographies and/or nonstandard
time periods which are not provided by the agency. Providing ACS data users tools for
change of support has recently been identified as an important problem by a National
Academy of Sciences panel (National Academy of Sciences 2015). STCOS method-
ology enables model-based estimates to be computed with public-use ACS releases.

The change of support problem can be illustrated by a concrete example, taking
median household income as the variable of interest here and for the remainder of
the article. Suppose we would like to produce 3-year model-based estimates in Mis-

Fig. 1 The state of Missouri in 2015. Thin lines mark boundaries between the 114 counties and one independent city. Shaded areas with thick lines mark the eight congressional districts

Congressional Districts and Counties in Missouri
2015 Geography

souri congressional districts for the year 2015. Congressional districts are geographic regions which receive representation by an elected official in the U.S. House of Representatives and are determined by a redistricting process which is based on data from each decennial census. The Census Bureau does release ACS estimates on congressional districts, but releases of 3-year estimates for all geographies were discontinued after 2013; therefore, model-based estimates may be of interest to data users. Geographies on which we want to produce estimates and predictions are referred to as *target supports*. Figure 1 displays the eight designated congressional districts in Missouri for the year 2015. Geographies on which direct estimates are available are used to fit the STCOS model and are referred to as *source supports*. For this illustration, we could take the source supports to be all 1-year, 3-year, and 5-year ACS releases for the counties within Missouri. Including available periods over a number of years allows the STCOS model to find trends in both time and space, and make use of estimates which represent varying levels of granularity and sparseness. Figure 2 shows direct estimates for Missouri in the year 2013. We notice that 1-year and 3-year period estimates have been suppressed for many counties. We emphasize that counties and congressional districts do not necessarily align, and the crux of the STCOS problem is to "translate" between the county-level observations and the congressional districts. The third type of support which must be discussed is the *fine-level support*. For this example, we could take the fine-level support to be the 2015 definition of counties in Missouri, shown in Fig. 1. The STCOS methodology works by translating each of the source supports to the fine-level support during the model fitting process. Once the model has been fit, estimates and predictions on target supports of interest are obtained by translating from the fine-level support. Raim et al. (2017) presents a model selection study with counties in the continental U.S. as target and source supports, congressional districts as target supports, and median household income as the ACS variable of interest. Section 5 will demonstrate a smaller-scale problem which requires less computing time.
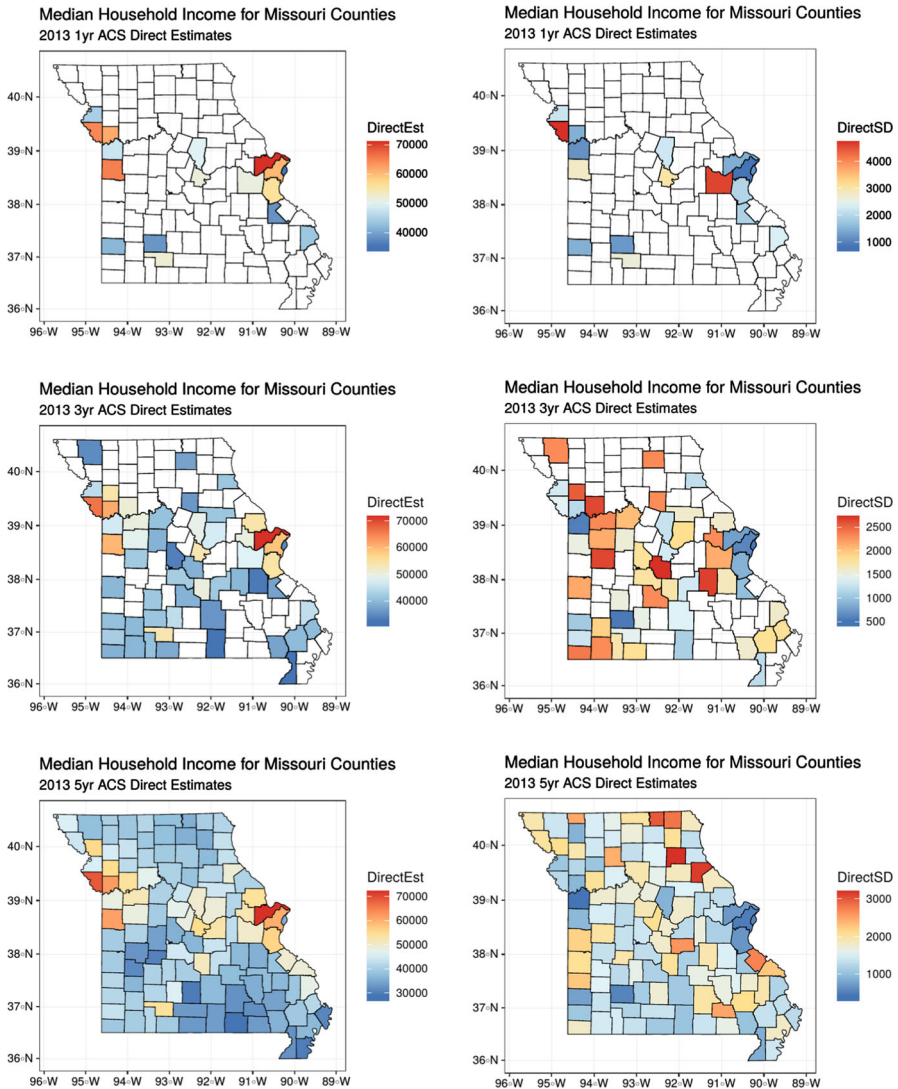
Fig. 2 County-level ACS data for median household income in Missouri for the year 2013. The left column shows direct estimates and the right column displays standard errors. The first, second, and third rows correspond to 1-year, 3-year, and 5-year period estimates, respectively. Public ACS estimates were not available for areas with white shading

3 The STCOS model

Let $\mathcal{T} = \{v_1, \ldots, v_T\}$ represent the set of times for which direct estimates are available, indexed by $t = 1, \ldots T$. Let $\mathcal{L}$ denote the set of possible lookback periods for which these estimates have been constructed. We will take $\mathcal{T}$ to consist of the years 2005 through 2017, corresponding to the ACS releases available during the prepa-

ration of this article, and $\mathcal{L} = \{1, 3, 5\}$ to denote 1-year, 3-year, and 5-year period releases. Therefore, ℓ-year direct estimates for year v_t are based on the time period $(v_{t-\ell+1}, \ldots, v_t)$. Data may not be released for all $(v_t, \ell) \in \mathcal{T} \times \mathcal{L}$; for example, ACS 3-year estimates were discontinued after 2013. Let $(\mathcal{T} \times \mathcal{L})^*$ denote the subset of $\mathcal{T} \times \mathcal{L}$ that corresponds to a data release. For each $(v_t, \ell) \in (\mathcal{T} \times \mathcal{L})^*$, the associated source support $D_{t\ell}$ is a collection of areal units whose estimates are included in the release. For each areal unit $A \in D_{t\ell}$, $Z_t^{(\ell)}(A)$ is the direct point estimate for one ACS variable of interest and $V_t^{(\ell)}(A)$ is the corresponding variance estimate. The fine level support will be denoted $D_B = \{B_1, \ldots, B_{n_B}\}$. The total surface area of a given areal unit A will be denoted $|A|$.

The STCOS model is a Bayesian hierarchical model (Cressie and Wikle 2011, Section 2.1) which will first state before describing components in detail. Let $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denote the multivariate normal distribution with density $\phi(\boldsymbol{x} \mid \boldsymbol{\mu}, \boldsymbol{\Sigma}) = (2\pi)^{-k/2}|\boldsymbol{\Sigma}|^{-1/2} \exp\{-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x} - \boldsymbol{\mu})\}$ for $\boldsymbol{x} \in \mathbb{R}^k$, where the dimension k depends on the context. Let $IG(a, b)$ denote the Inverse Gamma distribution with density $f_{IG}(x \mid a, b) = b^a x^{-a-1} e^{-b/x}/\Gamma(a) \cdot I(x > 0)$, where $I(\cdot)$ is the indicator function. First, the *data model* is

$$Z_t^{(\ell)}(A) = Y_t^{(\ell)}(A) + \varepsilon_t^{(\ell)}(A), \quad \varepsilon_t^{(\ell)}(A) \overset{ind}{\sim} N(0, V_t^{(\ell)}(A)),$$

for $A \in D_{t\ell}$ and $(v_t, \ell) \in (\mathcal{T} \times \mathcal{L})^*$. Second, the *process model* is

$$Y_t^{(\ell)}(A) = \boldsymbol{h}(A)^\top \boldsymbol{\mu}_B + \boldsymbol{s}_t^{(\ell)}(A)^\top \boldsymbol{\eta} + \xi_t^{(\ell)}(A),$$
$$[\boldsymbol{\eta} \mid \sigma_K^2] \sim N(\boldsymbol{0}, \sigma_K^2 \boldsymbol{K}),$$
$$[\xi_t^{(\ell)}(A) \mid \sigma_\xi^2] \overset{iid}{\sim} N(0, \sigma_\xi^2),$$

for $A \in D_{t\ell}$ and $(v_t, \ell) \in (\mathcal{T} \times \mathcal{L})^*$. Finally, the *parameter model* is

$$\boldsymbol{\mu}_B \sim N(\boldsymbol{0}, \sigma_\mu^2 \boldsymbol{I}), \quad \sigma_\mu^2 \sim IG(a_\mu, b_\mu), \quad \sigma_K^2 \sim IG(a_K, b_K), \quad \sigma_\xi^2 \sim IG(a_\xi, b_\xi).$$

The STCOS model assumes that direct estimates $Z_t^{(\ell)}(A)$ constitute a noisy observation of an underlying latent process $Y_t^{(\ell)}(A)$. The variance of the noise $\varepsilon_t^{(\ell)}(A)$ is assumed to be the direct variance estimate $V_t^{(\ell)}(A)$. The mean of the latent process $Y_t^{(\ell)}(A)$ consists of a coarse spatial trend $\boldsymbol{h}(A)^\top \boldsymbol{\mu}_B$ and a spatio-temporal random process $\boldsymbol{s}_t^{(\ell)}(A)^\top \boldsymbol{\eta}$. Conjugate priors are assumed for the coefficients and variance parameters from the previous two stages. The matrix $\boldsymbol{K}$, which provides the covariance structure for the random coefficient of $\boldsymbol{\eta}$, is assumed to be known and is computable from the fine-level support.

The latent process model is motivated by the following construction. Define a continuous-space discrete-time process,

$$Y(\boldsymbol{u}, v) = \delta(\boldsymbol{u}) + \sum_{j=1}^{\infty} \psi_j(\boldsymbol{u}, v) \cdot \eta_j, \quad \text{for } \boldsymbol{u} \in \bigcup_{i=1}^{n_B} B_i \text{ and } v \in \mathcal{T},$$

where $\delta(\boldsymbol{u})$ is a large-scale spatial trend process and $\{\psi_j(\boldsymbol{u}, v)\}_{j=1}^{\infty}$ is a prespecified set of spatio-temporal basis functions. Integrating $Y(\boldsymbol{u}, v)$ uniformly over $\boldsymbol{u} \in A$ and an ℓ-year period $\boldsymbol{v} = (v_{t-\ell+1}, \ldots, v_t)$,

$$
\begin{aligned}
Y_t^{(\ell)}(A) = {} & \frac{1}{|A|} \int_A \delta(\boldsymbol{u})\, d\boldsymbol{u} + \frac{1}{\ell|A|} \sum_{k=t-\ell+1}^{t} \sum_{j=1}^{r} \int_A \psi_j(\boldsymbol{u}, v_k) \cdot \eta_j\, d\boldsymbol{u} \\
& + \frac{1}{\ell|A|} \sum_{k=t-\ell+1}^{t} \sum_{j=r+1}^{\infty} \int_A \psi_j(\boldsymbol{u}, v_k) \cdot \eta_j\, d\boldsymbol{u} \\
= {} & \mu(A) + \boldsymbol{s}_t^{(\ell)}(A)^{\top} \boldsymbol{\eta} + \xi_t^{(\ell)}(A).
\end{aligned}
\tag{3.1}
$$

In (3.1), we have used the notation

$$
\mu(A) = \frac{1}{|A|} \int_A \delta(\boldsymbol{u})\, d\boldsymbol{u},
\tag{3.2}
$$

$$
\boldsymbol{s}_t^{(\ell)}(A)^{\top} \boldsymbol{\eta} = \frac{1}{\ell|A|} \sum_{k=t-\ell+1}^{t} \sum_{j=1}^{r} \int_A \psi_j(\boldsymbol{u}, v_k) \cdot \eta_j\, d\boldsymbol{u},
\tag{3.3}
$$

$$
\xi_t^{(\ell)}(A) = \frac{1}{\ell|A|} \sum_{k=t-\ell+1}^{t} \sum_{j=r+1}^{\infty} \int_A \psi_j(\boldsymbol{u}, v_k) \cdot \eta_j\, d\boldsymbol{u},
\tag{3.4}
$$

so that (3.2) represents a large-scale spatial trend, (3.3) is a spatio-temporal random process, and (3.4) is the remainder. We assume that $[\xi_t^{(\ell)}(A) \mid \sigma_\xi^2] \overset{\text{iid}}{\sim} N(0, \sigma_\xi^2)$, and make use of local bisquare basis functions for the small-scale spatio-temporal trend, which are of the form

$$
\psi_j(\boldsymbol{u}, v) = \left[2 - \frac{\|\boldsymbol{u} - \boldsymbol{c}_j\|^2}{w_s^2} - \frac{|v - g_j|^2}{w_t^2} \right]^2 \cdot I(\|\boldsymbol{u} - \boldsymbol{c}_j\| \le w_s) \cdot I(|v - g_j| \le w_t),
\tag{3.5}
$$

for $j = 1, \ldots, r$, with specified knots $\{(\boldsymbol{c}_j, g_j) : j = 1, \ldots, r\}$. Knots may be taken as the Cartesian product of a set of spatial knot points $\{\boldsymbol{c}_a : a = 1, \ldots, r_{\text{space}}\}$ and a set of temporal knot points $\{g_b : b = 1, \ldots, r_{\text{time}}\}$; however, this is not required in general. The basis functions also require specification of a spatial radius w_s and temporal radius w_t. We will select evenly spaced temporal knot points and spatial knot points according to a space-filling design (Nychka and Saltzman 1998). It can be difficult to specify w_s directly, as the influence of w_s depends on the coordinate system used in the supports. We therefore take $w_s = \tilde{w}_s \cdot Q_{0.05}$, where $Q_{0.05}$ is the 0.05 quantile of all nonzero pairwise distances between spatial cutpoints and $\tilde{w}_s$ is a parameter to be selected by the user. See Raim et al. (2017) for a model selection study varying several factors in this model such as the number of knot points and the selection of $\tilde{w}_s$ and w_t.

Basis functions at the area level may be obtained from bases (3.5) defined at the point level; for area A and an ℓ-year period specified by years $\boldsymbol{v} = (v_{t-\ell+1}, \ldots, v_t)$, let

$$\psi_j^{(\ell)}(A, \boldsymbol{v}) = \frac{1}{\ell} \sum_{k=t-\ell+1}^{t} \frac{1}{|A|} \int_A \psi_j(\boldsymbol{u}, v_k) d\boldsymbol{u}, \tag{3.6}$$

which can be computed by Monte Carlo approximation via

$$\psi_j^{(\ell)}(A, \boldsymbol{v}) \approx \frac{1}{\ell Q} \sum_{k=t-\ell+1}^{t} \sum_{q=1}^{Q} \psi_j(\boldsymbol{u}_q, v_k),$$

based on a random sample of locations $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_Q$ from a uniform distribution on the region A. Therefore, the basis expansion for an ℓ-year lookback period $\boldsymbol{v} = (v_{t-\ell+1}, \ldots, v_t)$ and area A is

$$\boldsymbol{s}_t^{(\ell)}(A)^\top = \left(\psi_1^{(\ell)}(A, \boldsymbol{v}), \ldots, \psi_r^{(\ell)}(A, \boldsymbol{v}) \right).$$

Next, for the large-scale spatial trend process, we make the simplifying assumption that

$$\delta(\boldsymbol{u}) = \sum_{i=1}^{n_B} \mu_i I(\boldsymbol{u} \in A \cap B_i),$$

for an area A. Then $\delta(\boldsymbol{u})$ takes on a constant value on each overlap $A \cap B_i$ for $B_i \in D_B$. Define

$$\boldsymbol{h}(A) = \left(\frac{|A \cap B_1|}{|A|}, \ldots, \frac{|A \cap B_{n_B}|}{|A|} \right)^\top$$

as the vector of proportions in which A overlaps with each area B_i in the fine-level support; this is based on the geography, and is therefore a known quantity in the analysis. Manipulation of geographical data in R will be discussed in Sects. 4 and 5. Integrating over $\boldsymbol{u} \in A$ yields

$$\mu(A) = \frac{1}{|A|} \sum_{i=1}^{n_B} \int_{A \cap B_i} \delta(\boldsymbol{u}) d\boldsymbol{u} = \frac{1}{|A|} \sum_{i=1}^{n_B} \mu_i \int_{A \cap B_i} d\boldsymbol{u}$$

$$= \sum_{i=1}^{n_B} \mu_i \frac{|A \cap B_i|}{|A|} = \boldsymbol{h}(A)^\top \boldsymbol{\mu}_B.$$

The coefficient $\boldsymbol{\mu}_B = (\mu_1, \ldots, \mu_{n_B})^\top$ represents the change of support coefficient between the fine-level support and all other supports, and is the primary quantity of interest in the model.

To simplify the remaining presentation, we now write the model in vector form. Suppose there are N total observations, indexed $i = 1, \ldots, N$, in all of the source supports combined. Let $\mathcal{H}$ be the mapping from each index i to a triple (A, t, ℓ) consisting of the area A, time v_t, and lookback ℓ for the ith observation. Let $\mathrm{vec}(\mathcal{S})$ denote a vector constructed from the elements of an ordered collection $\mathcal{S}$, $\mathrm{Diag}(\mathcal{S})$ represent a diagonal matrix with the elements of $\mathcal{S}$, and $\mathrm{rbind}(\mathcal{S})$ represent a matrix with the elements of $\mathcal{S}$ as rows. We may then write

$$
\begin{aligned}
\boldsymbol{Z} &= \mathrm{vec}\left(Z_t^{(\ell)}(A) : (A, t, \ell) = \mathcal{H}(i),\ i = 1, \ldots, N\right), \\
\boldsymbol{H} &= \mathrm{rbind}\left(\boldsymbol{h}_t^{(\ell)}(A)^\top : (A, t, \ell) = \mathcal{H}(i),\ i = 1, \ldots, N\right), \\
\boldsymbol{S} &= \mathrm{rbind}\left(\boldsymbol{s}_t^{(\ell)}(A)^\top : (A, t, \ell) = \mathcal{H}(i),\ i = 1, \ldots, N\right), \\
\boldsymbol{\xi} &= \mathrm{vec}\left(\xi_t^{(\ell)}(A) : (A, t, \ell) = \mathcal{H}(i),\ i = 1, \ldots, N\right), \\
\boldsymbol{\varepsilon} &= \mathrm{vec}\left(\varepsilon_t^{(\ell)}(A) : (A, t, \ell) = \mathcal{H}(i),\ i = 1, \ldots, N\right), \\
\boldsymbol{V} &= \mathrm{Diag}\left(V_t^{(\ell)}(A) : (A, t, \ell) = \mathcal{H}(i),\ i = 1, \ldots, N\right),
\end{aligned}
\tag{3.7}
$$

where $\boldsymbol{h}_t^{(\ell)}(A) = \boldsymbol{h}(A)$ does not vary with t or ℓ. The STCOS model can now be written

$$
\boldsymbol{Z} = \boldsymbol{H}\boldsymbol{\mu}_B + \boldsymbol{S}\boldsymbol{\eta} + \boldsymbol{\xi} + \boldsymbol{\varepsilon},
$$
$$
\boldsymbol{\varepsilon} \sim \mathrm{N}(\boldsymbol{0}, \boldsymbol{V}), \quad [\boldsymbol{\eta} \mid \sigma_K^2] \sim \mathrm{N}(\boldsymbol{0}, \sigma_K^2 \boldsymbol{K}), \quad [\boldsymbol{\xi} \mid \sigma_\xi^2] \sim \mathrm{N}(\boldsymbol{0}, \sigma_\xi^2 \boldsymbol{I}),
$$
$$
[\boldsymbol{\mu}_B \mid \sigma_\mu^2] \sim \mathrm{N}(\boldsymbol{0}, \sigma_\mu^2 \boldsymbol{I}), \quad \sigma_\mu^2 \sim \mathrm{IG}(a_\mu, b_\mu), \quad \sigma_K^2 \sim \mathrm{IG}(a_K, b_K), \quad \sigma_\xi^2 \sim \mathrm{IG}(a_\xi, b_\xi).
\tag{3.8}
$$

We will also define $\boldsymbol{Y} = \boldsymbol{H}\boldsymbol{\mu}_B + \boldsymbol{S}\boldsymbol{\eta} + \boldsymbol{\xi}$ as the latent process for the observations.

We now discuss specification of the matrix $\boldsymbol{K}$. Let $A \sim B$ be the predicate that area A is adjacent to area B with $A \sim A$ taken to be false by definition. Denote $\boldsymbol{W} = (w_{ij})$ as the $n_B \times n_B$ adjacency matrix with $w_{ij} = I(B_i \sim B_j)$ for $i, j \in \{1, \ldots, n_B\}$, and $\boldsymbol{D} = \mathrm{Diag}(w_{1+}, \ldots, w_{n+})$ with ith diagonal entry $w_{i+} = \sum_{\ell=1}^{n_B} w_{ij}$. The matrix $\boldsymbol{Q} = \boldsymbol{I} - \tau \boldsymbol{D}^{-1} \boldsymbol{W}$ corresponds to the precision matrix of a particular class of conditional autoregressive (CAR) process. We take $\tau \in (0, 1)$ to be known, for simplicity, to ensure that $\boldsymbol{Q}$ is nonsingular provided that areas $B_1, \ldots, B_{n_B}$ form a connected graph. Other choices of $\boldsymbol{Q}$ can be considered to obtain other classes of CAR precision matrices (see Cressie and Wikle 2011; Banerjee et al. 2014, and the references therein).

For the purpose of specifying a spatio-temporal variance, suppose the fine-level support is distributed according to the process

$$
\boldsymbol{Y}_t^* = \boldsymbol{\mu}_B + \boldsymbol{\zeta}_t, \quad \boldsymbol{\zeta}_t = \boldsymbol{M}\boldsymbol{\zeta}_{t-1} + \boldsymbol{b}_t, \quad [\boldsymbol{b}_t \mid \sigma_K^2] \stackrel{\mathrm{iid}}{\sim} \mathrm{N}(\boldsymbol{0}, \sigma_K^2 \boldsymbol{Q}^{-1}),
\tag{3.9}
$$

for $v_t \in \mathcal{T}$ and assume $\boldsymbol{b}_0 = \boldsymbol{0}$. That is, $\{\boldsymbol{Y}_t^*\}$ is a vector autoregressive (VAR) process in time and a CAR process in space. Let $\boldsymbol{\Sigma}_{Y^*}$ denote the covariance matrix of

$(Y_t^* : v_t \in \mathcal{T})$ under model (3.9). We take K to be the minimizer of

$$\|\boldsymbol{\Sigma}_{Y^*} - S^* C S^{*\top}\|_{\mathrm{F}}, \quad \text{such that } C \text{ is an } r \times r \text{ positive semidefinite matrix,} \quad (3.10)$$

under the Frobenius norm $\|\cdot\|_{\mathrm{F}}$, where $S^* = \mathrm{rbind}\left(s_t^{(\ell)}(A)^\top : A \in D_B, v_t \in \mathcal{T}, \ell \in \mathcal{L}\right)$ is the basis function expansion on the fine-level geography. In (3.10), $\boldsymbol{\Sigma}_{Y^*}$ represents the desired covariance structure under model (3.9), while $S^* C S^{*\top}$ represents the realized covariance contribution of $S\boldsymbol{\eta}$ in the model (3.8), where

$$\mathrm{Var}(Y \mid \boldsymbol{\mu}_B, \sigma_\mu^2, \sigma_\xi^2, \sigma_K^2) = \sigma_K^2 S K S^\top + \sigma_\xi^2 I,$$

conditionally on the random variables in the parameter model. The solution to (3.10),

$$C^* = (S^{*\top} S^*)^{-1} S^{*\top} \boldsymbol{\Sigma}_{Y^*} S^* (S^{*\top} S^*)^{-1},$$

provides the best positive approximant to $\boldsymbol{\Sigma}_{Y^*}$; details are given in "Appendix A". For the remainder of the article, we will take $\boldsymbol{\Sigma}_{Y^*}$ to be positive definite and S^* to be full rank so that K is positive definite. Bradley et al. (2015b) and Bradley et al. (2015a) further discuss this approach within the context spatio-temporal models, and Higham (1988) discusses the positive approximant problem in the general setting. We may write $\boldsymbol{\Sigma}_{Y^*} = \sigma_K^2 \tilde{\boldsymbol{\Sigma}}_{Y^*}$ so that

$$C^* = \sigma_K^2 K, \quad K = (S^{*\top} S^*)^{-1} S^{*\top} \tilde{\boldsymbol{\Sigma}}_{Y^*} S^* (S^{*\top} S^*)^{-1}. \quad (3.11)$$

Notice that $\tilde{\boldsymbol{\Sigma}}_{Y^*}$ and K are free of unknown parameters so that the solution of (3.10) does not need to be recomputed within MCMC iterations as parameter values are updated.

We consider several possible structures for K. First, assume that $M = I$ so that the fine-level process defined in (3.9) is a vector random walk with nonstationary autocovariance function

$$\boldsymbol{\Gamma}(s, t) = \mathrm{Cov}(Y_s^*, Y_t^*) = \min(s, t)\sigma_K^2 \boldsymbol{Q}^{-1},$$

conditioning on σ_K^2. Letting $\tilde{\boldsymbol{\Gamma}}(s, t) = \sigma_K^{-2} \boldsymbol{\Gamma}(s, t)$, which is free of σ_K^2, and choosing

$$\tilde{\boldsymbol{\Sigma}}_{Y^*} = \begin{bmatrix} \tilde{\boldsymbol{\Gamma}}(1, 1) & \cdots & \tilde{\boldsymbol{\Gamma}}(1, T) \\ \vdots & \ddots & \vdots \\ \tilde{\boldsymbol{\Gamma}}(T, 1) & \cdots & \tilde{\boldsymbol{\Gamma}}(T, T) \end{bmatrix}$$

as the covariance of $\{Y_t^*\}$, K is obtained from (3.11) to be

$$K = (S^{*\top} S^*)^{-1} \left[\sum_{s=1}^{T} \sum_{t=1}^{T} \min(s, t) S_s^{*\top} \boldsymbol{Q}^{-1} S_t^* \right] (S^{*\top} S^*)^{-1}, \quad (3.12)$$

where we define $S_t^* = \text{rbind}\left(s_t^{(\ell)}(A)^\top : A \in D_B, \ell \in \mathcal{L}\right)$ for each $v_t \in \mathcal{T}$. Another useful covariance structure arises if we assume that $M = 0$. This yields autocovariance function $\Gamma(s, t) = I(s = t)\sigma_K^2 Q^{-1}$ and $\Sigma_{Y^*} = \sigma_K^2 Q^{-1} \otimes I_T$, conditioning on σ_K^2, where $\otimes$ represents the Kronecker product. This structure supports nonzero covariance among areas at common times but independence between areas across times. The approximant (3.11) with $\tilde{\Sigma}_{Y^*} = Q^{-1} \otimes I_T$ is

$$K = (S^{*\top}S^*)^{-1}\left[\sum_{t=1}^{T} S_t^{*\top} Q^{-1} S_t^*\right](S^{*\top}S^*)^{-1}. \tag{3.13}$$

One more useful covariance structure assumes no spatial or temporal covariance;

$$K = I. \tag{3.14}$$

It is worth emphasizing that K describes the covariance structure for η, but the covariance contribution to the model occurs through Y via $SKS^\top$. For example, an independence assumption for η yields $SKS^\top = SS^\top$, which is not necessarily a diagonal matrix using the basis functions (3.6) or the dimension-reduced version discussed in Sect. 5. The covariance structures we consider in this work—namely (3.12), (3.13), and (3.14)—are a departure from Bradley et al. (2015b), who recommend computing M itself by further basis function decomposition.

We can obtain a Gibbs sampler by considering the joint distribution of the random quantities in (3.8),

$$
\begin{aligned}
&f(Z, \eta, \xi, \mu_B, \sigma_\mu^2, \sigma_K^2, \sigma_\xi^2) \\
&= \phi(Z \mid H\mu_B + S\eta + \xi, V) \cdot \phi(\xi \mid 0, \sigma_\xi^2 I) \cdot \phi(\eta \mid 0, \sigma_K^2 K) \\
&\quad \times \phi(\mu_B \mid 0, \sigma_\mu^2 I) \cdot f_{\text{IG}}(\sigma_\mu^2 \mid a_\mu, b_\mu) \cdot f_{\text{IG}}(\sigma_K^2 \mid a_K, b_K) \cdot f_{\text{IG}}(\sigma_\xi^2 \mid a_\xi, b_\xi),
\end{aligned}
$$

and deriving the full conditional distribution of each unknown parameter (e.g., Banerjee et al. 2014, Section 5.3). Here, the derivation is routine and details have been omitted for brevity. The steps of the Gibbs sampler which result from the full conditionals of $\mu_B, \eta, \xi, \sigma_\mu^2, \sigma_K^2$, and σ_ξ^2 are stated as Algorithm 3.1. The notation $[X \mid \text{—}]$ is used to denote the distribution of a given random variable X conditioned on all other random quantities.

4 Implementing STCOS in R

STCOS modeling can be roughly separated into three phases: assembling published estimates and geospatial data into a usable form, preparing matrices and vectors needed to fit the model, and finally fitting the model and producing results. To read and manipulate geospatial data, we will highlight the sf package (Pebesma 2018), which we find to be intuitive and comprehensive. For general data manipulation, such as filtering

Algorithm 3.1 Gibbs sampler steps for STCOS model.

1. Draw $[\boldsymbol{\mu}_B \mid —] \sim \mathrm{N}(\boldsymbol{\vartheta}_\mu, \boldsymbol{\Omega}_\mu^{-1})$, with $\boldsymbol{\Omega}_\mu = \boldsymbol{H}^\top \boldsymbol{V}^{-1} \boldsymbol{H} + \sigma_\mu^{-2} \boldsymbol{I}$. and $\boldsymbol{\vartheta}_\mu = \boldsymbol{\Omega}_\mu^{-1} \boldsymbol{H}^\top \boldsymbol{V}^{-1}(\boldsymbol{Z} - \boldsymbol{S}\boldsymbol{\eta} - \boldsymbol{\xi})$.
2. Draw $[\boldsymbol{\eta} \mid —] \sim \mathrm{N}(\boldsymbol{\vartheta}_\eta, \boldsymbol{\Omega}_\eta^{-1})$, with $\boldsymbol{\Omega}_\eta = \boldsymbol{S}^\top \boldsymbol{V}^{-1} \boldsymbol{S} + \sigma_K^{-2} \boldsymbol{K}^{-1}$ and $\boldsymbol{\vartheta}_\eta = \boldsymbol{\Omega}_\eta^{-1} \boldsymbol{S}^\top \boldsymbol{V}^{-1}(\boldsymbol{Z} - \boldsymbol{H}\boldsymbol{\mu}_B - \boldsymbol{\xi})$.
3. Draw $[\boldsymbol{\xi} \mid —] \sim \mathrm{N}(\boldsymbol{\vartheta}_\xi, \boldsymbol{\Omega}_\xi^{-1})$, with $\boldsymbol{\Omega}_\xi = \boldsymbol{V}^{-1} + \sigma_\xi^{-2} \boldsymbol{I}$ and $\boldsymbol{\vartheta}_\xi = \boldsymbol{\Omega}_\xi \boldsymbol{V}^{-1}(\boldsymbol{Z} - \boldsymbol{H}\boldsymbol{\mu}_B - \boldsymbol{S}\boldsymbol{\eta})$.
4. Draw $[\sigma_\mu^2 \mid —] \sim \mathrm{IG}(a_\mu^*, b_\mu^*)$, with $a_\mu^* = a_\mu + n_B/2$ and $b_\mu^* = b_\mu + \boldsymbol{\mu}_B^\top \boldsymbol{\mu}_B/2$.
5. Draw $[\sigma_K^2 \mid —] \sim \mathrm{IG}(a_K^*, b_K^*)$, with $a_K^* = a_K + r/2$ and $b_K^* = b_K + \boldsymbol{\eta}^\top \boldsymbol{K}^{-1} \boldsymbol{\eta}/2$.
6. Draw $[\sigma_\xi^2 \mid —] \sim \mathrm{IG}(a_\xi^*, b_\xi^*)$, with $a_\xi^* = a_\xi + N/2$ and $b_\xi^* = b_\xi + \boldsymbol{\xi}^\top \boldsymbol{\xi}/2$.

records and selecting columns from a table, we will make use of the `dplyr` package (Wickham et al. 2020). To produce high quality graphics, we use the `ggplot2` package (Wickham 2016). The `dplyr` and `ggplot2` packages are especially convenient because of their compatibility with `sf` objects. The `tigris` package (Walker 2018) provides a convenient way to request geographical data from the Census Bureau Tiger/Line database within R. The `fields` package (Nychka et al. 2017) can be used to select spatial knot points by a space-filling design. General purpose platforms for Bayesian computing, including `Stan` (Carpenter et al. 2017), `JAGS` (Depaoli et al. 2016), `BUGS` (Lunn et al. 2009), and `Nimble` (de Valpine et al. 2017), are accessible through an R interface. A major advantage of such platforms is that samplers can be programmed simply by specifying a model and providing the data. In contrast, the traditional Gibbs sampler approach may require derivation, programming, and testing for each new model. However, general purpose platforms may not be well-suited to certain classes of models or to very large datasets. We will illustrate the use of `Stan` via the `rstan` package (Stan Development Team 2020) in addition to the Gibbs sampler from Sect. 3.

Some aspects of implementing STCOS analysis in R can be laborious and prone to error. To reduce the burden, we introduce the `stcos` package. The `stcos` package provides several major capabilities including: functions to compute overlap matrix $\boldsymbol{H}$ and adjacency matrix $\boldsymbol{W}$, basis functions to compute $\boldsymbol{S}$, construction of covariance $\boldsymbol{K}$, maximum likelihood estimation for the STCOS model, and an STCOS Gibbs sampler. Basis functions discussed in Sect. 3 will be demonstrated shortly. Internal basis function calculations are carried out in C++, for efficiency, via the `Rcpp` and `RcppArmadillo` packages (Eddelbuettel 2013; Eddelbuettel and Sanderson 2014). Matrices such as $\boldsymbol{H}$ and $\boldsymbol{S}$ are likely to be sparse in many STCOS applications; we use the `Matrix` package (Bates and Maechler 2019) to support operations on sparse matrices.

We will now give an overview of the major STCOS computations which will be needed in R. Section 5 will provide a demonstration connecting these pieces into a complete analysis. The following packages are assumed to be loaded in all coding examples.

```
1   R> library("sf")
2   R> library("dplyr")
3   R> library("stcos")
```

A natural way to encode geographical features in source, fine-level, and target supports is via `sf` objects. Data associated with the supports can be embedded into `sf` objects to facilitate model preparation and graphical display. Therefore, our preferred workflow will be to produce `sf` objects with direct and model-based estimates. An example of a prepared source support object is as follows.

```
1   R> head(acs5_2013, 3)
2   Simple feature collection with 3 features and 8 fields
3   geometry type:  POLYGON
4   dimension:      XY
5   bbox:           xmin: -10280140 ymin: 4712766 xmax: -10277220 ymax: 4714750
6   CRS:            EPSG:3857
7           geoid state county  tract blockgroup DirectEst DirectMOE DirectVar
8   1 290190005001    29    019 000500          1      9970      3157   3683788
9   2 290190005002    29    019 000500          2     12083      7048  18360194
10  3 290190006001    29    019 000600          1    105156     16979 106553987
11                                    geometry
12  1 POLYGON ((-10278231 4713772...
13  2 POLYGON ((-10279369 4713339...
14  3 POLYGON ((-10280135 4712926...
```

Note that we have manipulated this output and some subsequent outputs to ensure that they fit on the page. The CRS descriptor specifies a geographical coordinate system for the data. A number of standard coordinate systems are used to express geographical data, each having its own benefits and drawbacks. Coordinates such as latitude and longitude used in the global positioning system (GPS) describe points on the surface of the globe. Map projections provide two-dimensional representations of a region, which are convenient in many applications but necessarily distort the geography in some way. Conformal projections are designed to preserve local shape and are considered suitable for smaller domains, but distort areas when applied to large regions. On the other hand, equal-area projections are designed to preserve areas over large regions. To apply STCOS and other spatial-temporal methods, the analyst must select an appropriate coordinate system. We have chosen the Web Mercator projection (EPSG:3857) which is utilized in a number of online mapping services and is a variation of the conformal Mercator projection (Battersby et al. 2014). Further discussion and references on coordinate systems can be found in Bivand et al. (2013, Chapter 4) and Waller and Gotway (2004, Chapter 3).

All source, target, and fine-level supports in an analysis should use a common coordinate system so that they are compatible; this is not a limitation, as the analyst may transform an `sf` object from its original coordinates using the `sf::st_transform` function. Furthermore, methods described in this article are suited toward coordinates in a map projection rather than a globe representation. For example, the Euclidean distance utilized in (3.6) does not take into account the curvature of the Earth. An analysis using spherical coordinates—which may be appropriate for a larger-scale domain—might instead consider a great circle distance. Now that the importance of the coordinate system has been emphasized, coordinates will be considered as raw numerical values for the remainder of the article.

The last four lines of the previous display show a table with nine fields, where each row corresponds to an area (county) in the file. The `geometry` field contains details about the county's geography, which we typically will not want to manipulate directly. The fields STATE and COUNTY represent Federal Information Processing Standards

(FIPS) codes for the state and county respectively, and GEO_ID is an identifier which combines the two. The fields DirectEst, DirectMOE, and DirectVar represent direct ACS estimates of median household income and an associated estimate of margin of error and variance. Preparation of such an sf object from geographical data and direct estimates will be discussed in Sect. 5.1.

A function overlap_matrix is provided to compute the H matrix.

```
1   R> H = overlap_matrix(dom1, dom2, proportion = TRUE)
```

Here, dom1 and dom2 are sf objects which describe domains of areal units. The result is an nrow(dom1) by nrow(dom2) matrix. If proportion = FALSE, the entries represent the amount of area for each overlap; otherwise rows are normalized to proportions which sum to 1.

The stcos package provides several variations of the local bisquare basis functions discussed in Sect. 3. The following functions operate on data where space is represented at the point-level.

```
1   R> S = spatial_bisquare(dom, knots, w_s)
2   R> S = spacetime_bisquare(dom, knots, w_s, w_t)
```

The function spacetime_bisquare implements (3.5) which uses information both in space and time, while spatial_bisquare implements a space-only version

$$\varphi_j(\boldsymbol{u}) = \left[1 - \|\boldsymbol{u} - \boldsymbol{c}_j\|^2 / w^2\right]^2 \cdot I(\|\boldsymbol{u} - \boldsymbol{c}_j\| \le w). \tag{4.1}$$

The object dom may either be a numerical matrix or an object of type sf or sfc containing points. In both cases, the first two columns/coordinates represent the spatial coordinates and the third represents time, if applicable. The object knots provides knot points, and may similarly be specified as either a numerical matrix or a sf or sfc object containing points. Coordinate systems for the points in knots are expected to be compatible with those in dom. Two-dimensional points are expected in spatial_bisquare, where each represents a $\boldsymbol{c}_j$. Similarly, three-dimensional points are expected in spacetime_bisquare so that each represents a $(\boldsymbol{c}_j, g_j)$. The variables w_s and w_t correspond to the spatial and temporal radius, respectively.

The following functions operate on data where space is represented at an area-level.

```
1   R> S = areal_spatial_bisquare(dom, knots, w_s, control = NULL)
2   R> S = areal_spacetime_bisquare(dom, period, knots, w_s, w_t, control = NULL)
```

The function areal_spacetime_bisquare implements (3.6), while areal_spatial_bisquare computes a space-only version

$$\bar{\varphi}_j(A) = \frac{1}{|A|} \int_A \varphi_j(\boldsymbol{u}) d\boldsymbol{u},$$

based on (4.1). Here, the object dom is of type sf or sfc and provides the geography for one or more areal units. The variable period is a numeric vector which represents time period $\boldsymbol{v} = (v_{t-\ell+1}, \ldots, v_t)$ used to evaluate (3.6). For example, if dom represents ACS 5-year estimates for 2017, we will take period = 2013:2017. The arguments knots, w_s and w_t are interpreted similarly as in the point-level

functions. The optional `control` argument is a list in which some additional factors can be adjusted, such as the number of Monte Carlo repetitions to be used in the approximation. The remainder of the demonstration focuses on the STCOS analysis; further details and examples for the basis functions can be found in the `stcos` manual.

Several options were described in Sect. 3 to compute the covariance matrix K; the `stcos` package provides functions to assist with the computations.

```
1   R> K = cov_approx_blockdiag(Qinv, S_fine)
2   R> K = cov_approx_randwalk(Qinv, S_fine)
```

Both calls produce an $r \times r$ matrix. The call to `cov_approx_randwalk` corresponds to the random walk structure in (3.12), while `cov_approx_blockdiag` corresponds to (3.13) which assumes independence across time. The structure in (3.14) which represents independent and identically distributed elements of $\boldsymbol{\eta}$ can be achieved with `K = Diagonal(n = r)`. The arguments `Qinv`, and `S_fine` correspond to the matrices $\boldsymbol{Q}^{-1}$ and $\boldsymbol{S}^*$ described in Sect. 3. Note that the function `car_precision` in `stcos` can be used to compute $\boldsymbol{Q}$ from adjacency matrix $\boldsymbol{W}$.

```
1   R> Q = car_precision(W, tau = 0.9, scale = TRUE)
```

The matrix $\boldsymbol{I} - \tau \boldsymbol{D}^{-1} \boldsymbol{W}$ is returned if `scale = TRUE`; otherwise $\boldsymbol{D} - \tau \boldsymbol{W}$ is returned.

Although we focus on Bayesian analysis, a function to compute maximum likelihood estimates (MLEs) is provided.

```
1   R> out = mle_stcos(z, v, H, S, K, init = list(sig2K = 1, sig2xi = 1))
2   R> sig2K_hat = out$sig2K_hat
3   R> sig2xi_hat = out$sig2xi_hat
4   R> mu_hat = out$mu_hat
```

Some details on MLE computation are given in "Appendix A". MLE computation is often much quicker than Bayesian computation, and may provide good starting values for an MCMC sampler. Here, `H`, `S`, and `K` are the matrices $\boldsymbol{H}$, $\boldsymbol{S}$, and $\boldsymbol{K}$ described in (3.7), while `z` represents the vector $\boldsymbol{Z}$ and `v` is the diagonal of the matrix $\boldsymbol{V}$. The Gibbs sampler described in Sect. 3 can be invoked using the `gibbs_stcos` function.

```
1   R> out = gibbs_stcos(z, v, H, S, Kinv = solve(K),
2   +    R = 10000, report_period = 1000, burn = 1000, thin = 10,
3   +    init = init)
4   R> muB_mcmc = out$muB_hist
5   R> eta_mcmc = out$eta_hist
6   R> xi_mcmc = out$xi_hist
7   R> sig2mu_mcmc = out$sig2mu_hist
8   R> sig2xi_mcmc = out$sig2xi_hist
9   R> sig2K_mcmc = out$sig2K_hist
```

Some helper functions are provided to process the output from the Gibbs sampler.

```
1   print(out)
2   logLik(out)
3   DIC(out)
4   E_mcmc = fitted(out, H_new, S_new)
5   Y_mcmc = predict(out, H_new, S_new)
```

The `print` function displays a brief summary of results from the sampler, while `logLik` computes the log-likelihood for each saved draw and `DIC` computes the Deviance information criterion (Spiegelhalter et al. 2002) using saved draws. Let $\tilde{\boldsymbol{H}}$

be an $\tilde{N} \times n$ overlap matrix and $\tilde{S}$ be an $\tilde{N} \times r$ basis matrix computed from target supports of interest, and H_new and S_new denote their representations in code. Let $\tilde{Y}$ denote the vector composed of the $\tilde{N}$ latent process variables

$$\tilde{Y}_t^{(\ell)}(A) = \tilde{h}(A)^\top \boldsymbol{\mu}_B + \tilde{s}_t^{(\ell)}(A)^\top \boldsymbol{\eta} + \tilde{\xi}_t^{(\ell)}(A)$$

associated with matrices $\tilde{H}$ and $\tilde{S}$. The fitted function produces draws from the posterior distribution of the mean

$$\mathrm{E}(\tilde{Y} \mid \boldsymbol{\mu}_B, \boldsymbol{\eta}) = \tilde{H}\boldsymbol{\mu}_B + \tilde{S}\boldsymbol{\eta},$$

so that E_mcmc is a matrix with $\tilde{N}$ columns where each row corresponds to a saved draw. Alternatively, the predict function produces draws from the posterior distribution of

$$\int \phi\left(\tilde{Y} \mid \tilde{H}\boldsymbol{\mu}_B + \tilde{S}\boldsymbol{\eta}, \sigma_\xi^2 I\right) f(\boldsymbol{\mu}_B, \boldsymbol{\eta}, \sigma_\xi^2, \mid Z, V) \, d\boldsymbol{\mu}_B \, d\boldsymbol{\eta} \, d\sigma_\xi^2.$$

5 Demonstration: City of Columbia neighborhoods

We now demonstrate an STCOS analysis on a small-scale but complete example using real data. Our target support consists of four neighborhoods in the City of Columbia in Boone County, Missouri. Geospatial data of the four neighborhoods has been provided by staff from the GIS Office for the City of Columbia. We would like to produce model-based estimates of median household income using observed ACS estimates from recent years. Specifically, we will consider 5-year ACS estimates at the block-group level for years 2013–2017 as our source supports, and will produce 5-year ACS estimates for year 2017 on the four neighborhoods as our target support.

The demonstration is split into several subsections. Section 5.1 considers raw inputs—ACS direct estimates and geographical features—and discusses how they can be assembled into a useful form for the analysis. Section 5.2 then prepares the inputs to the STCOS model: namely, Z, V, H, S, and K. Section 5.3 uses the Gibbs sampler in the stcos package to produce draws from the posterior distribution of STCOS parameters and consequently obtain the desired results from the analysis. Section 5.4 uses the Stan platform via the rstan package as an alternative method to obtain results. Finally, Sect. 5.5 compares the MLE to Bayesian results.

5.1 Assembling the data

We now briefly discuss how to gather geospatial data and ACS estimates and assemble them into sf objects for convenience. This is not intended to be an extensive guide, as numerous options to gather data (e.g. portals, APIs, and R packages) are available and continue to evolve. In Sect. 5.2, we will make use of datasets which have been constructed for the demonstration.

Geospatial data representing the target support were provided in the shapefile format (ESRI 1998). We now read the file and transform it to a projection of choice.

```
1   R> neighbs = st_read("neighborhoods.shp") %>% st_transform(crs = 3857)
```

To prepare the source supports, we must gather ACS estimates and corresponding geographical features. For this example, ACS estimates can be requested from the Census Bureau's Data API. The interface and data availability of the API are subject to change in the future, and examples shown next may need to be modified accordingly. Breakstone and Anderson (2019) provide a user guide with current specifications, including URL query format, available datasets, and codes for variable names. Note that limits are placed on the frequency and size of queries for unregistered users; higher-volume users may register for an API key to reduce restrictions. Estimates for our source supports can be requested from the API by constructing URLs with the following formats.

```
1   R> est_url = paste('https://api.census.gov/data/', year,
2   +   '/acs/acs5?get=NAME,B19013_001E&for=block%20group:*&in=state:29+county:019',
3   +   sep = '')
4   R> moe_url = paste('https://api.census.gov/data/', year,
5   +   '/acs/acs5?get=NAME,B19013_001M&for=block%20group:*&in=state:29+county:019',
6   +   sep = '')
```

Data for the direct point estimates and corresponding MOEs have been gathered using two separate calls to the API. The FIPS code for Missouri is `29` and the code for Boone County is `019`. The variable `B19013_001E` represents point estimates for "Median household income in the past 12 months", and `B19013_001M` represents corresponding MOEs. We can request the years of interest by taking `year` to be values `2013` through `2017`. We use the `jsonlite` package (Ooms 2014) to call the API and load the results into an R `data.frame`.

```
1   R> json_data = jsonlite::fromJSON(est_url)
2   R> est_dat = data.frame(json_data[-1,])
3   R> colnames(est_dat) = json_data[1,]
4
5   R> json_data = jsonlite::fromJSON(moe_url)
6   R> moe_dat = data.frame(json_data[-1,])
7   R> colnames(moe_dat) = json_data[1,]
```

We now merge `est_dat` and `moe_dat` together into a single `data.frame`.

```
1   my_dat = est_dat %>%
2       inner_join(moe_dat, by = c('state' = 'state', 'county' = 'county',
3           'tract' = 'tract', 'block group' = 'block group')) %>%
4       select(state, county, tract, blockgroup = 'block group',
5           DirectEst = B19013_001E, DirectMOE = B19013_001M) %>%
6       mutate(DirectEst = as.numeric(DirectEst)) %>%
7       mutate(DirectMOE = as.numeric(DirectMOE)) %>%
8       mutate(DirectEst = replace(DirectEst, DirectEst < 0, NA)) %>%
9       mutate(DirectMOE = replace(DirectMOE, DirectMOE < 0, NA)) %>%
10      mutate(DirectVar = (DirectMOE / qnorm(0.95))^2) %>%
11      arrange(tract, blockgroup)
```

There are a few details to mention in this data manipulation. We have taken some care because there is a space in the variable name `block group`, and because variables in the ACS data are interpreted as strings by default. We have transformed the MOE to a variance estimate, noting that published MOEs are to be interpreted as margins of error from $\alpha = 0.90$ confidence intervals (U.S. Census Bureau 2018); i.e.,

$$\mathrm{MOE} = z_{\alpha/2}\sqrt{\hat{V}} \quad \Longleftrightarrow \quad \hat{V} = \left(\frac{\mathrm{MOE}}{z_{\alpha/2}}\right)^2,$$

where $z_{\alpha/2} \approx 1.645$. We have also taken care to handle special values coded in the data; namely, large negative numbers for estimates and MOEs are returned by the API when estimates are not available,[1] which we convert to NA. We sort the entries by tract and block group for readability. The resulting data.frame appears as follows.

```
1   R> head(my_dat)
2     state county  tract blockgroup DirectEst DirectMOE DirectVar
3   1    29    019 000200          1     41063      6512  15673799
4   2    29    019 000200          2     31250      6978  17997303
5   3    29    019 000300          1     19420      7643  21591022
6   4    29    019 000300          2        NA        NA        NA
7   5    29    019 000300          3     21369     14558  78333750
8   6    29    019 000500          1     10995      5563  11438356
```

The presence of NA values in direct estimates—such as in tract 000300, blockgroup 2—can vary over area, year, and period. NA values will be addressed in Sect. 5.2, before the analysis. The tigris package (Walker 2018) provides a convenient way to request shapefiles from the Census Bureau Tiger/Line database. It is necessary that all supports are converted to a common coordinate system for the analysis, so use the function st::transform to match the projection we used earlier in the target support.

```
1   my_shp = tigris::block_groups(state = '29', county = '019', year = 2017) %>%
2       st_as_sf() %>%
3       st_transform(crs = 3857)
```

Now we augment the geospatial data with direct point estimates, MOEs, and variance estimates obtained earlier.

```
1   acs5_2017 = my_shp %>%
2       inner_join(my_dat, by = c('STATEFP' = 'state', 'COUNTYFP' = 'county',
3           'TRACTCE' = 'tract', 'BLKGRPCE' = 'blockgroup')) %>%
4       select(geoid = GEOID, state = STATEFP, county = COUNTYFP,
5           tract = TRACTCE, blockgroup = BLKGRPCE,
6           DirectEst, DirectMOE, DirectVar)
```

The resulting acs5_2017 is an object of type sf, whose first few entries are as follows (geometry column is not shown).

```
1    R> head(acs5_2017)
2    Simple feature collection with 6 features and 8 fields
3    geometry type:  POLYGON
4    dimension:      XY
5    bbox:           xmin: -10280690 ymin: 4712766 xmax: -10256290 ymax: 4752109
6    CRS:            EPSG:3857
7          geoid state county  tract blockgroup DirectEst DirectMOE DirectVar
8    1 290190005001    29    019 000500          1     10995      5563  11438356
9    2 290190005002    29    019 000500          2     13872      9503  33378510
10   3 290190006001    29    019 000600          1     45208     39073 564285643
11   4 290190006002    29    019 000600          2    107500     19868 145899495
12   5 290190020002    29    019 002000          2     62237     13529  67651414
13   6 290190020003    29    019 002000          3     51019     11166  46082999
```

---

[1] https://census.gov/data/developers/data-sets/acs-1year/notes-on-acs-estimate-and-annotation-values.html.

5.2 Preparing the analysis

The steps in Sect. 5.1 can be repeated so that all target, source, and fine-level supports are assembled as `sf` objects. The `stcos` package includes the following pre-constructed datasets to facilitate our demonstration.

```
1   R> data("acs_sf")
2   R> ls(pattern = "acs5_.*")
3   [1] "acs5_2013" "acs5_2014" "acs5_2015" "acs5_2016" "acs5_2017"
4   R> data("columbia_neighbs")
5   R> ls(pattern = "columbia")
6   [1] "columbia_neighbs"
```

Before we begin to prepare the terms in (3.7) for the STCOS model, let us create a version of the source supports with `NA` estimates removed. This will help to avoid complications in model fitting.

```
1   source_2013 = acs5_2013 %>% filter(!is.na(DirectEst) & !is.na(DirectVar))
2   source_2014 = acs5_2014 %>% filter(!is.na(DirectEst) & !is.na(DirectVar))
3   source_2015 = acs5_2015 %>% filter(!is.na(DirectEst) & !is.na(DirectVar))
4   source_2016 = acs5_2016 %>% filter(!is.na(DirectEst) & !is.na(DirectVar))
5   source_2017 = acs5_2017 %>% filter(!is.na(DirectEst) & !is.na(DirectVar))
```

We will choose our fine-level support based on the `acs5_2017` geography; i.e. the block group level geography for Boone County in 2017. However, because we have dropped some areas from the source supports, we should check for areas in `acs5_2017` which have zero or very little overlap with any areas in the source supports. If we identify such areas, we will drop them from the analysis to avoid rank-deficiency of the H matrix.

```
1   U = rbind(
2       overlap_matrix(source_2013, acs5_2017, proportion = FALSE),
3       overlap_matrix(source_2014, acs5_2017, proportion = FALSE),
4       overlap_matrix(source_2015, acs5_2017, proportion = FALSE),
5       overlap_matrix(source_2016, acs5_2017, proportion = FALSE),
6       overlap_matrix(source_2017, acs5_2017, proportion = FALSE)
7   )
8   dom_fine = acs5_2017 %>%
9       mutate(keep = (colSums(U) >= 10)) %>%
10      filter(keep == TRUE) %>%
11      select(-c("DirectEst", "DirectMOE", "DirectVar", "keep"))
12  n = nrow(dom_fine)
```

This creates `dom_fine` as a version of `acs5_2017`, excluding two block-groups having very little overlap (less than 10 square meters) with any of the source support areas, and ignoring the columns for the direct estimates, MOEs, and variance estimates.

The overlap matrix H for the analysis can now be created as follows.

```
1   H = rbind(
2       overlap_matrix(source_2013, dom_fine),
3       overlap_matrix(source_2014, dom_fine),
4       overlap_matrix(source_2015, dom_fine),
5       overlap_matrix(source_2016, dom_fine),
6       overlap_matrix(source_2017, dom_fine)
7   )
8   N = nrow(H)
```

To construct a bisquare basis, we must select spatio-temporal knot points. To select spatial knot points, we first draw a large number of points uniformly over the fine-level

domain using the `st_sample` function. We then use the `cover.design` function in the `fields` package, which finds a subset of these points to fill the space.

```
1  u = st_sample(dom_fine, size = 2000)
2  P = st_coordinates(u)
3  out = fields::cover.design(P, 200)
4  knots_sp = out$design
```

To select the spatial radius w_s, we compute the 0.05 quantile of the pairwise distances among the rows of `knots_sp`, as discussed in Sect. 3.

```
1  ws_tilde = 1
2  D = dist(knots_sp)
3  w_s = ws_tilde * quantile(D[D > 0], prob = 0.05, type = 1)
```

Alternatively, evenly spaced points can be achieved with the hexagonal sampling method in the `sf::st_sample` function. This is quicker than `fields::cover.design`.

```
1  u = st_sample(dom_fine, 200, type = "hexagonal")
2  knots_sp_alt = st_coordinates(u)
3  D = dist(knots_sp_alt)
4  w_s_alt = ws_tilde * quantile(D[D > 0], prob = 0.05, type = 1)
```

Figure 3 illustrates the selected spatial knot points and radius using both the space-filling method and hexagonal sampling. Both methods succeed in creating a grid of evenly-spaced points, although the latter follow a more strict pattern. More evenly-spaced points can also be obtained with the space-filling method by taking an initial sample size larger than our selection of 2000.

We choose the temporal knot points to be (2009, 2009.5, ..., 2016.5, 2017), covering the years relevant to the 5-year ACS estimates for years 2013–2017.

```
1  knots_t = seq(2009, 2017, by = 0.5)
2  w_t = 1
```

More sophisticated date/time functions can assist in constructing temporal knots, though a numerical representation is ultimately needed. An alternative choice for temporal knots created with `Date` objects is given next. When treated as numerical, such objects represent days elapsed since January 1, 1970. Here we may again use the quantile approach to determine a radius which is suitable for this unit of time.

```
1  dates = seq(as.Date("2009-01-01"), as.Date("2017-01-01"), by = "6 months")
2  knots_t_alt = as.numeric(dates)
3  wt_tilde_alt = 1
4  D = dist(knots_t_alt)
5  w_t_alt = wt_tilde_alt * quantile(D[D > 0], prob = 0.05, type = 1)
```

Now we use the `merge` function in the `base` package (R Core Team 2020) to perform a Cartesian join between the spatial knots `knots_sp` and temporal knots `knots_t`, which yields the set of spatio-temporal knots.

```
1  knots = merge(knots_sp, knots_t)
```

Now, we use the basis functions to compute the design matrix S.

```
1  bs_ctrl = list(mc_reps = 500)
2  S_full = rbind(
3      areal_spacetime_bisquare(source_2013, 2009:2013, knots, w_s, w_t, bs_ctrl),
4      areal_spacetime_bisquare(source_2014, 2010:2014, knots, w_s, w_t, bs_ctrl),
5      areal_spacetime_bisquare(source_2015, 2011:2015, knots, w_s, w_t, bs_ctrl),
6      areal_spacetime_bisquare(source_2016, 2012:2016, knots, w_s, w_t, bs_ctrl),
7      areal_spacetime_bisquare(source_2017, 2013:2017, knots, w_s, w_t, bs_ctrl)
8  )
```
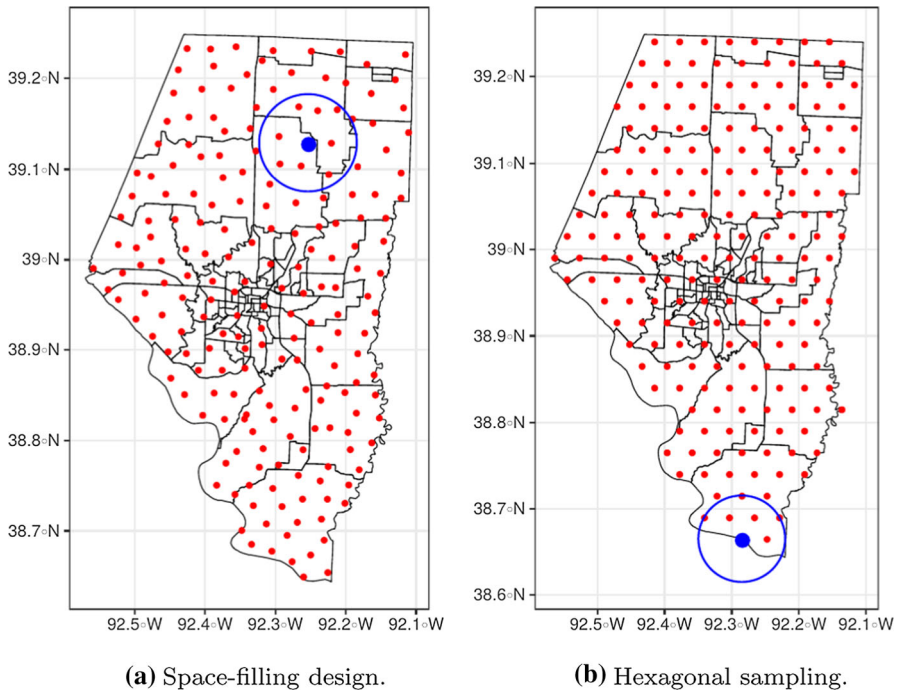
(a) Space-filling design. **(b)** Hexagonal sampling.

Fig. 3 Spatial knot points selected for spatio-temporal basis function. Red dots indicate knot points and blue circles display the spatial radius at one particular knot point. **a** The space-filling design whose radius was $w_s = 7712.70$, using the quantile calculation and taking $\tilde{w}_s = 1$. **b** Hexagonal sampling, whose radius was $w_s = 7169.13$ using the same quantile calculation and choice of $\tilde{w}_s$. Note that units of w_s are meters in the selected coordinate system (color figure online)

We can also compute the design matrix S^* on the fine-level support, which is needed to compute K under some of the possible structures.

```
1   S_fine_full = rbind(
2       areal_spacetime_bisquare(dom_fine, 2009, knots, w_s, w_t, bs_ctrl),
3       areal_spacetime_bisquare(dom_fine, 2010, knots, w_s, w_t, bs_ctrl),
4       areal_spacetime_bisquare(dom_fine, 2011, knots, w_s, w_t, bs_ctrl),
5       areal_spacetime_bisquare(dom_fine, 2012, knots, w_s, w_t, bs_ctrl),
6       areal_spacetime_bisquare(dom_fine, 2013, knots, w_s, w_t, bs_ctrl),
7       areal_spacetime_bisquare(dom_fine, 2014, knots, w_s, w_t, bs_ctrl),
8       areal_spacetime_bisquare(dom_fine, 2015, knots, w_s, w_t, bs_ctrl),
9       areal_spacetime_bisquare(dom_fine, 2016, knots, w_s, w_t, bs_ctrl),
10      areal_spacetime_bisquare(dom_fine, 2017, knots, w_s, w_t, bs_ctrl)
11  )
```

Next we need vectors z and v to represent the direct point estimates and associated variance estimates.
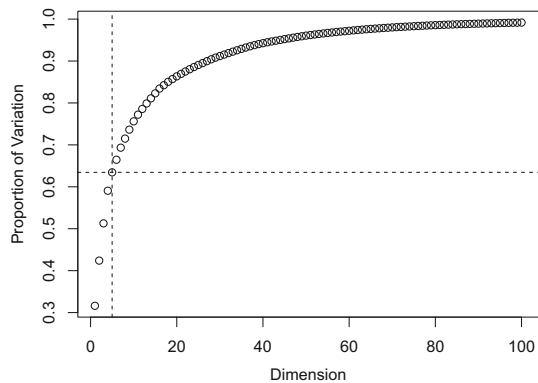
```
1   z = c(source_2013$DirectEst, source_2014$DirectEst, source_2015$DirectEst,
2       source_2016$DirectEst, source_2017$DirectEst)
3   v = c(source_2013$DirectVar, source_2014$DirectVar, source_2015$DirectVar,
4       source_2016$DirectVar, source_2017$DirectVar)
```

Because z and v contain rather large numbers, we standardize z for the analysis and make a corresponding transformation to v.

Fig. 4 Plot of the eigenvalues of $S^\top S$. The vertical line shows that 5 eigenvectors are needed to capture 65% of the variation. The y-axis has been truncated to maintain visibility for small dimensions; the total number of eigenvalues is 7500


```
1   z_scaled = scale(z)
2   v_scaled = v / var(z)
```

The expression for `v_scaled` arises from considering $\mathrm{Var}[a^{-1/2}(Z_i - b)] = a^{-1}\mathrm{Var}(Z_i)$ for constants $a > 0$ and $b \in \mathbb{R}$, which is estimated by $a^{-1}e_i^\top V e_i$ with e_i the ith column of an $N \times N$ identity matrix. The design matrix S with our choice of basis function can have a large number of columns and a high degree of multi-collinearity; if not addressed, this can lead to poor mixing in the MCMC sampler. A simple workaround is to reduce the dimension of S using principal components analysis (PCA). First we compute the reduction, using 65% of the variability, as expressed as a proportion of the eigenvalues.

```
1   eig = eigen(t(S_full) %*% S_full)
2   idx_S = which(cumsum(eig$values) / sum(eig$values) < 0.65)
```

Figure 4 shows that this can be accomplished by projecting from the original 3,400 columns to $r = 19$ columns. Now we apply the reduction to S as well as S^*.

```
1   Tx_S = eig$vectors[,idx_S]
2   S = S_full %*% Tx_S
3   S_fine = S_fine_full %*% Tx_S
4   r = ncol(S)
```

The last ingredient needed to run the analysis is the matrix K. We will use the random walk structure in (3.12) to express both spatial and temporal dependence. First, let us compute the covariance matrix Q^{-1} of a CAR process for the fine-level support.

```
1   W = adjacency_matrix(dom_fine)
2   Q = car_precision(W, tau = 0.9, scale = TRUE)
3   Qinv = solve(Q)
```

Now compute K using Q^{-1} and S^*.

```
1   K = cov_approx_randwalk(Qinv, S_fine)
```

5.3 Fitting with Gibbs sampler

We now proceed to run the Gibbs sampler. We will produce a chain of 10,000 iterations, discard the first 2000 draws, and keep one of every 10th remaining draw. We will use hyperparameters $a_\mu = 1$, $b_\mu = 1$, $a_K = 1$, $b_K = 2$, $a_\xi = 1$, and $b_\xi = 2$.
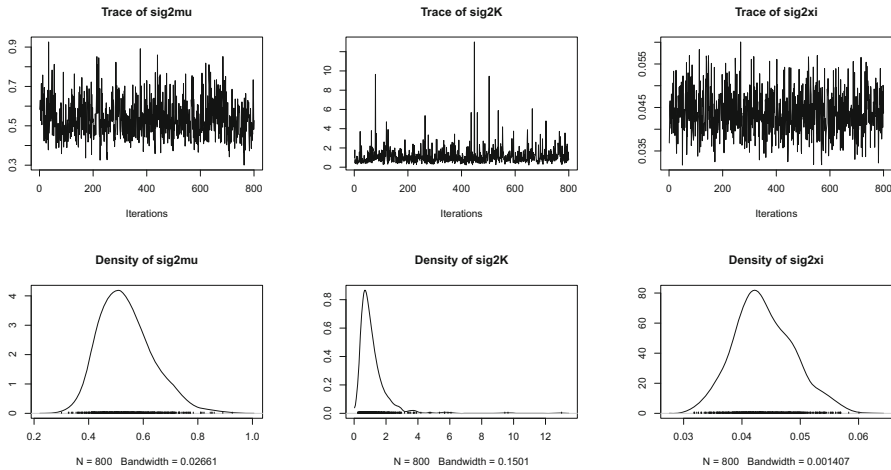
Fig. 5 Trace and density plots for draws of the variance components σ_μ^2, σ_K^2, and σ_ξ^2 from the Gibbs sampler

```
1   R> hyper = list(a_sig2K = 1, b_sig2K = 2, a_sig2xi = 1, b_sig2xi = 2,
2   +     a_sig2mu = 1, b_sig2mu = 2)
3   R> gibbs_out = gibbs_stcos(z = z_scaled, v = v_scaled, H = H, S = S,
4   +     Kinv = Kinv, R = 10000, report_period = 2000, burn = 2000,
5   +     thin = 10, hyper = hyper)
6   2020-05-17 17:11:15 - Begin Gibbs sampler
7   2020-05-17 17:11:52 - Begin iteration 2000
8   2020-05-17 17:12:29 - Begin iteration 4000
9   2020-05-17 17:12:50 - Begin iteration 6000
10  2020-05-17 17:13:09 - Begin iteration 8000
11  2020-05-17 17:13:28 - Begin iteration 10000
12  2020-05-17 17:13:28 - Finished Gibbs sampler
13  R> print(gibbs_out)
14  Fit for STCOS model
15  --
16            Mean         SD        2.5%        25%        75%       97.5%
17  sig2mu 0.52574414 0.094603198 0.36964018 0.45835116 0.57962109 0.74252701
18  sig2K  1.12632689 0.829211900 0.34238233 0.62142495 1.34628662 3.35183509
19  sig2xi 0.04368451 0.005022067 0.03485603 0.04021765 0.04669926 0.05533603
20  --
21  Saved 800 draws
22  DIC: 210.798981
23  Elapsed time: 00:02:08
```

The mcmc class in the coda package (Plummer et al. 2006) helps to manage and plot the draws.

```
1   library("coda")
2   varcomps_mcmc = mcmc(data.frame(
3       sig2mu = gibbs_out$sig2mu_hist,
4       sig2xi = gibbs_out$sig2xi_hist,
5       sig2K = gibbs_out$sig2K_hist
6   ))
7   plot(varcomps_mcmc)
```

Figure 5 displays trace and density plots of the variance components $\sigma_\mu^2, \sigma_\xi^2$, and σ_K^2.

Using the fitted model, we can produce model-based estimates on target supports of interest. In this example, we would like to produce 5-year 2017 estimates for our

four neighborhoods in Boone County: Central, East, North, and Paris. The following code computes model-based estimates for these areas and embeds them into an `sf` object for plotting.

```
1   nb_out = neighbs
2   H_new = overlap_matrix(nb_out, dom_fine)                # New overlap
3   S_new_full = areal_spacetime_bisquare(nb_out,
4       2013:2017, knots, w_s, w_t, bs_ctrl)               # New basis fn
5   S_new = S_new_full %*% Tx_S                            # Reduce dimension
6
7   EY_scaled = fitted(gibbs_out, H_new, S_new)           # Get draws of E(Y)
8   EY = sd(z) * EY_scaled + mean(z)                      # Uncenter and unscale
9
10  alpha = 0.10
11  nb_out$E_mean = colMeans(EY)                          # Point estimates
12  nb_out$E_sd = apply(EY, 2, sd)                        # SDs
13  nb_out$E_lo = apply(EY, 2, quantile, prob = alpha/2)  # Credible interval lo
14  nb_out$E_hi = apply(EY, 2, quantile, prob = 1-alpha/2) # Credible interval hi
15  nb_out$E_median = apply(EY, 2, median)                # Median
16  nb_out$E_moe = apply(EY, 2, sd) * qnorm(1-alpha/2)    # MOE
```

The objects `H_new` and `S_new` represent design matrices $\tilde{H}$ and $\tilde{S}$, respectively, based on the geography of neighborhoods. The `fitted` function was then used to produce draws from the posterior distribution of $E(\tilde{Y})$. We then transformed the resulting estimates back to the original scale, having previously centered and scaled them before model fitting. The remainder of the code display summarizes draws of the posterior mean in several ways, obtaining a model-based estimate of its median, mean, standard deviation, MOE ($z_{\alpha/2} \times$ standard deviation), and a 90% credible interval. The resulting `sf` object is displayed below (`geometry` column is not shown).

```
1   R> print(nb_out)
2   Simple feature collection with 4 features and 7 fields
3   geometry type:  POLYGON
4   dimension:      XY
5   bbox:           xmin: -10280270 ymin: 4715036 xmax: -10269750 ymax: 4723860
6   CRS:            EPSG:3857
7             Region   E_mean     E_sd     E_lo     E_hi E_median    E_moe
8   1        Central 26705.85 1921.623 23456.96 29709.53 26720.70 3160.788
9   2           East 44127.78 2450.811 40155.21 47983.46 44186.03 4031.225
10  3          North 44171.24 2863.373 39519.40 48933.11 44040.79 4709.829
11  4 Paris63Corridor 20386.72 3663.098 14448.61 26325.73 20309.78 6025.261
```

We are now ready to plot our estimates. The code to reproduce our plots is somewhat lengthy and can be found in the supplemental materials. First we compare direct and model-based estimates for 2017 source supports to assess whether the model fit is reasonable. Figure 6a, b show maps of the two sets of estimates. Figure 6c, d compare the two sets of estimates via scatter plots; year 2014 and year 2017 estimates are shown for comparison. Variation between direct and model-based estimates appears to be smaller for year 2014, with the exception of the block group with the largest direct estimate that year. Finally, Fig. 7 shows the four neighborhoods of our target support in the context of the 2017 5-year direct estimates. This provides a visual aid to assess plausibility of the target support estimates. The North and East neighborhoods appear to be in the immediate vicinity of block groups with higher median household income than the West and Paris neighborhoods.
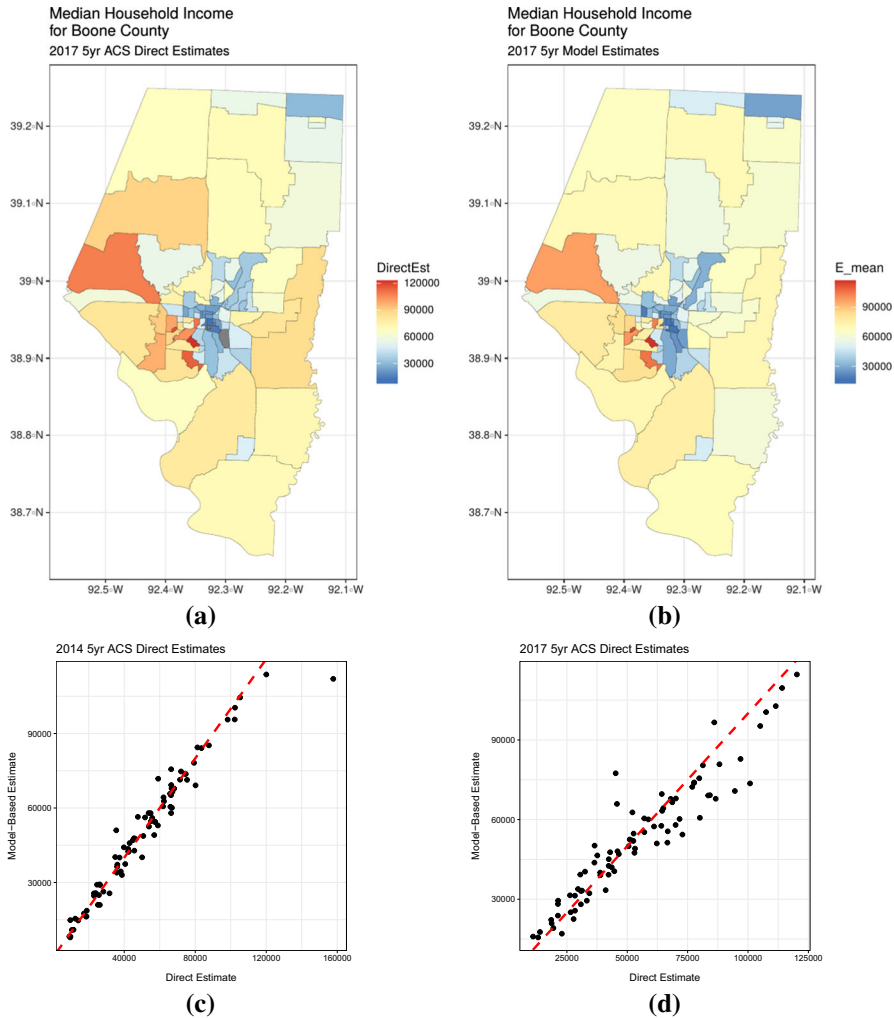
Fig. 6 Comparison of direct and model-based ACS 5-year estimates. **a**, **b** Maps based on the two estimates for year 2017. **c**, **d** Scatter plots comparing the two sets of estimates for years 2014 and 2017 respectively

5.4 Fitting with `Stan`

We will now refit the model from Sect. 5.3 using `Stan` instead of the Gibbs sampler. First, we will need a `Stan` model specification. We will create a file named `stcos.stan` with the following contents.

```
1   data {
2       int<lower=0> N;   int<lower=0> n;   int<lower=0> r;
3       vector[N] z;      vector[N] v;      matrix[N,n] H;
4       matrix[N,r] S;    matrix[r,r] K;    real alpha_K;
5       real beta_K;      real alpha_xi;    real beta_xi;
6       real alpha_mu;    real beta_mu;
7   }
```

```
8    parameters {
9        vector[n] mu;      real<lower=0> sig2K;
10       vector[r] eta;     real<lower=0> sig2xi;
11       vector[N] xi;      real<lower=0> sig2mu;
12   }
13   model {
14       sig2K ~ inv_gamma(alpha_K, beta_K);
15       sig2xi ~ inv_gamma(alpha_xi, beta_xi);
16       sig2mu ~ inv_gamma(alpha_mu, beta_mu);
17       eta ~ multi_normal(rep_vector(0,r), sig2K * K);
18       mu ~ normal(0, sqrt(sig2mu));
19       xi ~ normal(0, sqrt(sig2xi));
20       z ~ normal(H*mu + S*eta + xi, sqrt(v));
21   }
```

Now, in R, pass the data and model specification to stan to initiate fitting.

```
1    library("rstan")
2    stan_dat = list(
3      N = N, n = n, r = r, z = z_scaled, v = v_scaled, H = as.matrix(H),
4      S = as.matrix(S), K = as.matrix(K),
5      alpha_K = 1, beta_K = 2, alpha_xi = 1, beta_xi = 2, alpha_mu = 1, beta_mu = 2
6    )
```

```
1    R> stan_out = stan(file = "stcos.stan", data = stan_dat, iter = 2000, chains = 2)
2    SAMPLING FOR MODEL 'stcos' NOW (CHAIN 1).
3    ...
4    Chain 1:  Elapsed Time: 11.8561 seconds (Warm-up)
5    Chain 1:                10.5266 seconds (Sampling)
6    Chain 1:                22.3827 seconds (Total)
7    ...
8    SAMPLING FOR MODEL 'stcos' NOW (CHAIN 2).
9    ...
10   Chain 2:  Elapsed Time: 10.9951 seconds (Warm-up)
11   Chain 2:                9.87796 seconds (Sampling)
12   Chain 2:                20.873 seconds (Total)
```

Here we have requested two chains of length 2000 each. In addition to the time needed for sampling, Stan may require time to compile the model specification. Upon successful completion of sampling, the following R code can be used to extract draws and produce results.
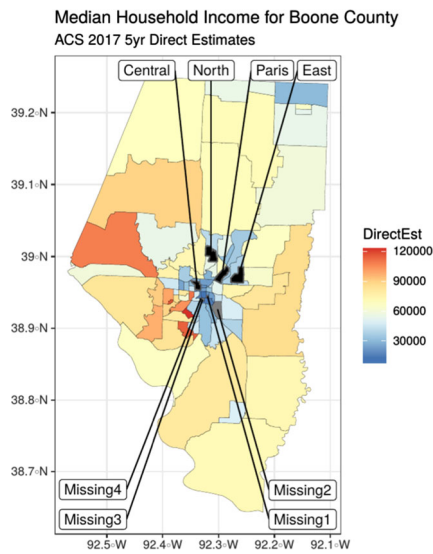
```
1    stan_draws = extract(stan_out, pars = c("mu", "eta"), permuted = TRUE)
2
3    nb_out = neighbs
4    H_new = overlap_matrix(nb_out, dom_fine)                    # New overlap
5    S_new_full = areal_spacetime_bisquare(nb_out,
6        2013:2017, knots, w_s, w_t, bs_ctrl)                    # New basis fn
7    S_new = S_new_full %*% Tx_S                                 # Reduce dimension
8
9    EY_scaled = stan_draws$mu %*% t(H_new) +
10       stan_draws$eta %*% t(S_new)                             # Draws of E(Y)
11   EY = sd(z) * EY_scaled + mean(z)                           # Uncenter and unscale
12
13   alpha = 0.10
14   nb_out$E_mean = colMeans(EY)                                # Point estimates
15   nb_out$E_sd = apply(EY, 2, sd)                              # SDs
16   nb_out$E_lo = apply(EY, 2, quantile, prob = alpha/2)        # Credible interval lo
17   nb_out$E_hi = apply(EY, 2, quantile, prob = 1-alpha/2)      # Credible interval hi
18   nb_out$E_median = apply(EY, 2, median)                      # Median
19   nb_out$E_moe = apply(EY, 2, sd) * qnorm(1-alpha/2)          # MOE
```

The result of print(nb_out) can be compared to the corresponding output from the Gibbs sampler in Sect. 5.3.

Region	Mean	SD	CI Lo	CI Hi	Median	MOE
Central	26,931.74	1,921.27	23,693.43	29,925.61	26,963.39	3,160.21
East	44,199.97	2,449.92	40,217.61	48,043.27	44,256.71	4,029.76
North	44,329.41	2,861.68	39,679.15	49,037.54	44,202.85	4,707.04
Paris	20,822.12	3,636.90	14,965.75	26,665.93	20,772.98	5,982.17

(a) Estimates based on STCOS model.

(b) Map of 2017 5-year direct estimates.

Fig. 7 Model-based ACS 5-year estimates for the Central, East, North, and Paris neighborhoods in year 2017 are shown in **a**. **b** The locations of the four neighborhoods (shaded in black), and year 2017 direct 5-year estimates in Boone County block groups for comparison. Direct estimates were not available for block groups marked as "Missing", which are shaded white

Fig. 8 Comparison of MLEs and draws from the Gibbs sampler. The first 10 components of μ_B are displayed in **a**, while **b** displays estimates for the four neighborhoods which have been transformed to the original scale of the direct estimates. Boxplots correspond to Gibbs sampler draws and red triangles represent MLEs (color figure online)

5.5 Fitting with maximum likelihood

Finally, we compute maximum likelihood estimates to compare to our Bayesian results.

```
1   R> mle_out = mle_stcos(z_scaled, v_scaled, H, S, K)
2   R> print(mle_out$sig2K_hat)
3   [1] 1.310004e-11
4   R> print(mle_out$sig2xi_hat)
5   [1] 1.225813e-11
6   R> print(mle_out$loglik)
7   [1] 67.29006
```

Estimates for both σ_ξ^2 and σ_K^2 are very small, which indicates that the direct variance estimates V are capturing much of the variability among Z. This can be contrasted with the Bayesian approach, which finds a non-zero effect of σ_ξ^2 and σ_K^2 through the addition of prior information. The following code extracts the MLE $\hat{\mu}_B$, computes estimates $\tilde{H}\hat{\mu}$ of the four neighborhoods, and transforms those estimates to the original scale of the direct estimates.

```
1   H_new = overlap_matrix(neighbs, dom_fine)
2   mu_hat = mle_out$mu_hat
3   z_hat_scaled = as.numeric(H_new %*% mu_hat)
4   z_hat = sd(z) * z_hat_scaled + mean(z)
```

Figure 8 plots MLEs with box plots of corresponding saved draws from the Gibbs sampler. Figure 8a plots the first 10 components of $\hat{\mu}_B$, while Fig. 8b plots the four components of $\tilde{H}\hat{\mu}$ corresponding to neighborhoods. Code to reproduce Fig. 8 is provided in the supplemental materials. Bayesian and maximum likelihood estimates are not seen to be vastly different, and we anticipate that they would become closer as the total number of observations N becomes large relative to n_B. In the current setting where $N = 421$ and $n_B = 85$, we would recommend the Bayesian approach.

6 Conclusions

In this article, we have demonstrated a complete implementation of STCOS methodology for R users. We worked through a small example to estimate median household income in several neighborhoods in the City of Columbia in Boone County, MO. Established R packages such as sf, dplyr, Matrix, and rstan were instrumental in the process, from initially gathering the data, to carrying out the MCMC, to placing results into a usable form. The stcos package was introduced to assist with some intricate programming steps not covered by other packages, especially computing areal spatio-temporal basis functions. Use of the highlighted tools significantly reduces the learning curve to program an analysis; however, some technical experience and effort are still required for a successful implementation. Future work may involve additional improvements to the stcos package for efficiency and usability, as well as software support for other spatial and spatio-temporal methodologies.

A Computational details and proofs

We will make use of the following well-known property in several places.

Property 1 If $A \in \mathbb{R}^{m \times k}, B \in \mathbb{R}^{k \times l}, C \in \mathbb{R}^{l \times n}$, then $\mathrm{vec}(ABC) = (C^\top \otimes A)\mathrm{vec}(B)$.

The following proposition gives the explicit solution to the minimization problem stated in (3.10). Bradley et al. (2015a) considers a similar problem featuring a more general objective function but assuming that the columns of S are orthonormal. Higham (1988) gives a general discussion of problems involving Frobenius and 2-norm distance minimization.

Proposition 1 (Frobenius Norm Minimization) *Suppose $S \in \mathbb{R}^{n \times r}$ has rank r and $\Sigma \in \mathbb{R}^{n \times n}$ is positive definite. The minimizer $X \in \mathbb{R}^{r \times r}$ of $\|\Sigma - SXS^\top\|_F$ is $X = (S^\top S)^{-1} S^\top \Sigma S (S^\top S)^{-1}$.*

Proof Using Property 1, we have

$$
\begin{aligned}
\|\Sigma - SXS^\top\|_F^2 &= \mathrm{vec}\left[\Sigma - SXS^\top\right]^\top \mathrm{vec}\left[\Sigma - SXS^\top\right] \\
&= \left[\mathrm{vec}(\Sigma) - \mathrm{vec}(SXS^\top)\right]^\top \left[\mathrm{vec}(\Sigma) - \mathrm{vec}(SXS^\top)\right] \\
&= [\mathrm{vec}(\Sigma) - (S \otimes S)\mathrm{vec}(X)]^\top [\mathrm{vec}(\Sigma) - (S \otimes S)\mathrm{vec}(X)] \\
&= \|\mathrm{vec}(\Sigma) - (S \otimes S)\mathrm{vec}(X)\|_2^2, \qquad\qquad (A.1)
\end{aligned}
$$

where the norm on the last line is the usual 2-norm on $\mathbb{R}^{n^2}$. We recognize the expression in (A.1) as a standard least squares minimization whose solution is

$$
\begin{aligned}
\mathrm{vec}(X) &= [(S \otimes S)^\top (S \otimes S)]^{-1} (S \otimes S)^\top \mathrm{vec}(\Sigma) \\
&= [(S^\top \otimes S^\top)(S \otimes S)]^{-1}(S^\top \otimes S^\top)\mathrm{vec}(\Sigma) \\
&= [S^\top S \otimes S^\top S]^{-1}\mathrm{vec}(S^\top \Sigma S) \\
&= [(S^\top S)^{-1} \otimes (S^\top S)^{-1}]\mathrm{vec}(S^\top \Sigma S) \\
&= \mathrm{vec}\left[(S^\top S)^{-1} S^\top \Sigma S (S^\top S)^{-1}\right].
\end{aligned}
$$

Therefore, the minimizer is $X = (S^\top S)^{-1} S^\top \Sigma S (S^\top S)^{-1}$, as desired. □

Remark 1 (MLE Computation) To compute the MLE for the STCOS model, we first note that the likelihood, excluding the parameter model, is

$$
f(Z \mid \mu_B, \sigma_K^2, \sigma_\xi^2) = \int \phi(Z \mid H\mu_B + S\eta, \sigma_\xi^2 I + V) \cdot \phi(\eta \mid 0, \sigma_K^2 K)d\eta
$$

$$= \phi(\boldsymbol{Z} \mid \boldsymbol{H}\boldsymbol{\mu}_B, \boldsymbol{\Delta})$$

$$= (2\pi)^{-N/2}|\boldsymbol{\Delta}|^{-1/2}\exp\left\{-\frac{1}{2}(\boldsymbol{Z} - \boldsymbol{H}\boldsymbol{\mu}_B)^\top \boldsymbol{\Delta}^{-1}(\boldsymbol{Z} - \boldsymbol{H}\boldsymbol{\mu}_B)\right\},$$

where $\boldsymbol{\Delta} = \sigma_\xi^2 \boldsymbol{I} + \boldsymbol{V} + \sigma_K^2 \boldsymbol{S}\boldsymbol{K}\boldsymbol{S}^\top$. Given σ_K^2 and σ_ξ^2, the likelihood is maximized by the weighted least squares estimator $\hat{\boldsymbol{\mu}}_B = (\boldsymbol{H}^\top \boldsymbol{\Delta}^{-1}\boldsymbol{H})^{-1}\boldsymbol{H}^\top \boldsymbol{\Delta}^{-1}\boldsymbol{Z}$. To estimate the unknown σ_K^2 and σ_ξ^2, we carry out numerical maximization on the partially maximized log-likelihood

$$\ell(\sigma_K^2, \sigma_\xi^2) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Delta}| - \frac{1}{2}(\boldsymbol{Z} - \boldsymbol{H}\hat{\boldsymbol{\mu}}_B)^\top \boldsymbol{\Delta}^{-1}(\boldsymbol{Z} - \boldsymbol{H}\hat{\boldsymbol{\mu}}_B).$$

To enforce the constraints that $\sigma_K^2 > 0$ and $\sigma_\xi^2 > 0$, we optimize over $(\vartheta_1, \vartheta_2) \in \mathbb{R}^2$ and take $\sigma_K^2 = \exp(\vartheta_1)$, $\sigma_\xi^2 = \exp(\vartheta_2)$.

References

Banerjee S, Carlin BP, Gelfand AE (2014) Hierarchical modeling and analysis for spatial data, 2nd edn. Chapman and Hall, London

Bates D, Maechler M (2019) Matrix: sparse and dense matrix classes and methods. https://CRAN.R-project.org/package=Matrix. R package version 1.2-18

Battersby SE, Finn MP, Usery EL, Yamamoto KH (2014) Implications of Web Mercator and its use in online mapping. Cartogr Int J Geogr Inf Geovis 49(2):85–101. https://doi.org/10.3138/carto.49.2.2313

Bivand RS, Pebesma E, Gómez-Rubio V (2013) Applied spatial data analysis with R, 2nd edn. Springer, Berlin

Bradley JR, Holan SH, Wikle CK (2015a) Multivariate spatio-temporal models for high-dimensional areal data with application to longitudinal employer-household dynamics. Ann Appl Stat 9(4):1761–1791. https://doi.org/10.1214/15-AOAS862

Bradley JR, Wikle CK, Holan SH (2015b) Spatio-temporal change of support with application to American Community Survey multi-year period estimates. Stat 4(1):255–270. https://doi.org/10.1002/sta4.94

Breakstone CD, Anderson TS (2019) Census data API user guide. https://www.census.gov/data/developers/guidance/api-user-guide.html. Version 1.6

Brunsdon C (2014) pycno: Pycnophylactic Interpolation. https://CRAN.R-project.org/package=pycno. R package version 1.2

Carpenter B, Gelman A, Hoffman M, Lee D, Goodrich B, Betancourt M, Brubaker M, Guo J, Li P, Riddell A (2017) Stan: a probabilistic programming language. J Stat Softw 76(1):1–32. https://doi.org/10.18637/jss.v076.i01

Cortes RX, Rey S, Knaap E (2019) pysal/tobler: Tobler initial release. https://doi.org/10.5281/zenodo.3386577

Cressie N, Wikle CK (2011) Statistics for spatio-temporal data. Wiley, New York

de Valpine P, Turek D, Paciorek CJ, Anderson-Bergman C, Lang DT, Bodik R (2017) Programming with models: writing statistical algorithms for general model structures with NIMBLE. J Comput Graph Stat 26(2):403–413. https://doi.org/10.1080/10618600.2016.1172487

Depaoli S, Clifton JP, Cobb PR (2016) Just another Gibbs sampler (JAGS): flexible software for MCMC implementation. J Educ Behav Stat 41(6):628–649. https://doi.org/10.3102/1076998616664876

Eddelbuettel D (2013) Seamless R and C++ integration with Rcpp. Springer, Berlin

Eddelbuettel D, Sanderson C (2014) RcppArmadillo: accelerating R with high-performance C++ linear algebra. Comput Stat Data Anal 71:1054–1063. https://doi.org/10.1016/j.csda.2013.02.005

Eicher CL, Brewer CA (2001) Dasymetric mapping and areal interpolation: implementation and evaluation. Cartogr Geogr Inf Sci 28(2):125–138. https://doi.org/10.1559/152304001782173727

ESRI (1998) ESRI shapefile technical description. https://www.esri.com/library/whitepapers/pdfs/shapefile.pdf. Accessed 27 May 2020

Fuentes M, Song H-R, Ghosh SK, Holland DM, Davis JM (2006) Spatial association between speciated fine particles and mortality. Biometrics 62(3):855–863. https://doi.org/10.1111/j.1541-0420.2006.00526.x

Gotway CA, Young LJ (2002) Combining incompatible spatial data. J Am Stat Assoc 97(458):632–648. https://doi.org/10.1198/016214502760047140

Higham NJ (1988) Computing a nearest symmetric positive semidefinite matrix. Linear Algebra Appl 103:103–118. https://doi.org/10.1016/0024-3795(88)90223-6

Lam NS-N (1983) Spatial interpolation methods: a review. Am Cartogr 10(2):129–150. https://doi.org/10.1559/152304083783914958

Lunn D, Spiegelhalter D, Thomas A, Best N (2009) The BUGS project: evolution, critique and future directions. Stat Med 28(25):3049–3067. https://doi.org/10.1002/sim.3680

Mileu N, Queirós M (2018) Development of a QGIS plugin to dasymetric mapping. In: Free and open source software for geospatial (FOSS4G) conference proceedings, vol 18, no 9. https://doi.org/10.7275/3628-0a51

National Academy of Sciences (2015) Realizing the potential of the American Community Survey: challenges, tradeoffs, and opportunities. National Academies Press, Washington, DC. https://doi.org/10.17226/21653

Nguyen H, Cressie N, Braverman A (2012) Spatial statistical data fusion for remote sensing applications. J Am Stat Assoc 107(499):1004–1018. https://doi.org/10.1080/01621459.2012.694717

Nychka D, Saltzman N (1998) Design of air quality monitoring networks. Lecture notes in statistics. Springer, pp 51–76. https://doi.org/10.1007/978-1-4612-2226-2_4

Nychka D, Furrer R, Paige J, Sain S (2017) Fields: tools for spatial data. University Corporation for Atmospheric Research, Boulder, CO, USA. https://github.com/NCAR/Fields. R package version 10.3. Accessed 27 May 2020

Ooms J (2014) The jsonlite package: a practical and consistent mapping between JSON data and R objects. arXiv:1403.2805

Pebesma E (2018) Simple features for R: standardized support for spatial vector data. R J 10(1):439–446. https://doi.org/10.32614/RJ-2018-009

Plummer M, Best N, Cowles K, Vines K (2006) CODA: convergence diagnosis and output analysis for MCMC. R News 6(1):7–11

Prener CG, Revord CK (2019) Areal: an R package for areal weighted interpolation. J Open Source Softw 4(37):1221. https://doi.org/10.21105/joss.01221

Qiu F, Zhang C, Zhou Y (2012) The development of an areal interpolation ArcGIS extension and a comparative study. GISci Remote Sens 49(5):644–663. https://doi.org/10.2747/1548-1603.49.5.644

R Core Team (2020) R: a language and environment for statistical computing. R Foundation for Statistical Computing, Vienna

Raim AM, Holan SH, Bradley JR, Wikle CK (2017) A model selection study for spatio-temporal change of support. In: JSM Proceedings, government statistics section. American Statistical Association, Alexandria, pp 1524–1540

Rode M, Arhonditsis G, Balin D, Kebede T, Krysanova V, Van Griensven A, Van der Zee SE (2010) New challenges in integrated water quality modelling. Hydrol Process 24(24):3447–3461. https://doi.org/10.1002/hyp.7766

Spiegelhalter DJ, Best NG, Carlin BP, Van Der Linde A (2002) Bayesian measures of model complexity and fit. J R Stat Soc Ser B (Stat Methodol) 64(4):583–639. https://doi.org/10.1111/1467-9868.00353

Stan Development Team (2020) RStan: the R interface to Stan. http://mc-stan.org/. R package version 2.19.3. Accessed 27 May 2020

Tobler WR (1979) Smooth pycnophylactic interpolation for geographical regions. J Am Stat Assoc 74(367):519–530. https://doi.org/10.1080/01621459.1979.10481647

U.S. Census Bureau (2016) American Community Survey data suppression. https://www.census.gov/programs-surveys/acs/technical-documentation/data-suppression.html. Accessed 2 Sept 2019

U.S. Census Bureau (2018) Understanding and using American Community Survey data: What all data users need to know. https://www.census.gov/programs-surveys/acs/guidance/handbooks/general.html. Accessed 2 Sept 2019

Walker K (2018) tigris: Load Census TIGER/Line Shapefiles. https://CRAN.R-project.org/package=tigris. R package version 0.7. Accessed 27 May 2020

Waller LA, Gotway CA (2004) Applied spatial statistics for public health data. Wiley-Interscience, New York

Weinberg DH, Abowd JM, Belli RF, Cressie N, Folch DC, Holan SH, Levenstein MC, Olson KM, Reiter JP, Shapiro MD, Smyth JD, Soh L-K, Spencer BD, Spielman SE, Vilhuber L, Wikle CK (2018) Effects of a government-academic partnership: Has the NSF-Census Bureau Research Network helped improve the US statistical system? J Surv Stat Methodol. https://doi.org/10.1093/jssam/smy023

Wickham H (2016) ggplot2: elegant graphics for data analysis. Springer, New York

Wickham H, François R, Henry L, Müller K (2020) dplyr: a grammar of data manipulation. https://CRAN.R-project.org/package=dplyr. R package version 0.8.5. Accessed 27 May 2020

Wikle CK, Berliner LM (2005) Combining information across spatial scales. Technometrics 47(1):80–91. https://doi.org/10.1198/004017004000000572

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Andrew M. Raim[1] ⓘ **· Scott H. Holan[2,3] · Jonathan R. Bradley[4] · Christopher K. Wikle[2]**

Scott H. Holan
holans@missouri.edu

Jonathan R. Bradley
bradley@stat.fsu.edu

Christopher K. Wikle
wiklec@missouri.edu

[1]   Center for Statistical Research and Methodology, U.S. Census Bureau, 4600 Silver Hill Road, Washington, DC 20233, USA

[2]   Department of Statistics, University of Missouri, Columbia, MO, USA

[3]   Office of the Associate Director for Research and Methodology, Washington, DC, USA

[4]   Department of Statistics, Florida State University, Tallahassee, FL, USA