Journal of Statistical Software

April 2007, Volume 19, Issue 2. http://www.jstatsoft.org/

Bayesian Smoothing with Gaussian Processes Using Fourier Basis Functions in the spectralGP Package

Christopher J. Paciorek
Harvard School of Public Health

Abstract

The spectral representation of stationary Gaussian processes via the Fourier basis provides a computationally efficient specification of spatial surfaces and nonparametric regression functions for use in various statistical models. I describe the representation in detail and introduce the **spectralGP** package in R for computations. Because of the large number of basis coefficients, some form of shrinkage is necessary; I focus on a natural Bayesian approach via a particular parameterized prior structure that approximates stationary Gaussian processes on a regular grid. I review several models from the literature for data that do not lie on a grid, suggest a simple model modification, and provide example code demonstrating MCMC sampling using the **spectralGP** package. I describe reasons that mixing can be slow in certain situations and provide some suggestions for MCMC techniques to improve mixing, also with example code, and some general recommendations grounded in experience.

Keywords: Bayesian statistics, Fourier basis, FFT, geostatistics, generalized linear mixed model, generalized additive model, Markov chain Monte Carlo, spatial statistics, spectral representation.

1. Introduction

Smoothing in the context of spatial modeling and nonparametric regression, often in an additive modelling scenario such as a generalized linear mixed model (GLMM) or generalized additive model (GAM), is a common technique in applied statistical work. A basic general model is

$$\begin{aligned} Y_i &\sim \mathcal{F}(f(\boldsymbol{x}_i, \boldsymbol{s}_i), \kappa) \\ h(f(\boldsymbol{x}_i, \boldsymbol{s}_i)) &= \boldsymbol{x}_i^\top \boldsymbol{\beta} + g(\boldsymbol{s}_i; \boldsymbol{\theta}), \end{aligned} \tag{1}$$

where Y_i, $i = 1, \ldots n$, is the ith outcome, $\mathcal{F}$ is commonly an exponential family distribution, κ is a dispersion parameter, $h(\cdot)$ is the link function, $\boldsymbol{x}_i$ is a vector of covariates for the ith observation, and $g(\boldsymbol{s}_i; \boldsymbol{\theta})$ is a smooth function, parameterized by $\boldsymbol{\theta}$, evaluated at the location or covariate value of the ith observation, $\boldsymbol{s}_i$, depending on whether the smooth function is in the spatial domain or covariate space. In this work I focus on settings in which $g(\cdot; \boldsymbol{\theta})$ is a spatial surface, but results hold generally for one dimension and potentially for higher dimensions.

There have been two basic approaches to modelling the smooth function, $g(\cdot; \boldsymbol{\theta})$, each with a variety of parameterizations. One approach considers the function as deterministic within a generalized additive model (GAM) framework (Hastie and Tibshirani 1990; Wood 2006), for example, using a thin plate spline or radial basis function representation with the function estimated via a penalized approach. The other takes a random effects, or equivalent stochastic process, approach in which the smooth function is treated stochastically, potentially via a Bayesian approach. Within this latter approach, one might consider a collection of correlated random effects, in which case (1) is a generalized linear mixed model (GLMM) (McCulloch and Searle 2001; Ruppert, Wand, and Carroll 2003). Alternatively, stochastic process representations such as kriging (Cressie 1993) or Bayesian versions of kriging (Banerjee, Carlin, and Gelfand 2004) usually take $g(\cdot; \boldsymbol{\theta})$ to be a Gaussian process. The random effects approach can also be considered as a stochastic process representation based on the implied covariance function of the process induced by the covariance structure of the random effects. Note that by considering a prior over functions or equivalently over the coefficients of basis functions, the additive model can be expressed in a Bayesian fashion, and there are connections between the thin plate spline and stochastic process approaches (Cressie 1993; Nychka 2000) and also between thin plate splines and mixed model representations (Ruppert *et al.* 2003). When interest lies in the linear coefficients and the smooth structure/spatial covariance is a nuisance, one approach to fitting such models is via estimating equations (e.g. Heagerty and Lele 1998; Heagerty and Lumley 2000; Oman, Landsman, Carmel, and Kadmon 2007). My primary interest is in situations in which the smooth function is the outcome of interest, for example, in predicting exposure to pollutants or spatial surfaces of climate variables, in which case such methods are not useful.

While models of the form (1) have a simple structure, unless the responses are Gaussian and the sample size is limited, fitting them can be difficult for computational reasons. If the response were Gaussian, there are many methods, both classical and Bayesian, for estimating $\boldsymbol{\beta}$, $g(\cdot; \boldsymbol{\theta})$, and $\boldsymbol{\theta}$. Most methods rely on integrating $g(\cdot; \boldsymbol{\theta})$ out of the model to produce a marginal likelihood or posterior, thereby moving the smooth structure out of the mean and into the variance, such that the observations have a simple, mean structure, (in (1) this is linear in a set of covariates), and a variance that is a sum of independent noise and spatially correlated structure. This leaves a small number of parameters to be estimated, often using numerical maximization or MCMC. However, for large n, computations can be burdensome as they involve matrix calculations of $O(n^3)$. In the non-Gaussian case and in hierarchical modeling in which the unknown process lies in the hierarchy of the model, this integration cannot be done analytically, which leads to substantial difficulty in fitting the model because of the high dimensional quantities that need to be estimated, as well as burdensome matrix calculations. One set of approaches to the problem focuses on the integral in the GLMM framework, using EM (McCulloch 1994, 1997; Booth and Hobert 1999) and numerical integration (Hedeker and Gibbons 1994; Gibbons and Hedeker 1997) to

maximize the likelihood or approximating the integral to produce a penalized quasi-likelihood that can be maximized by iteratively weighted least squares (IWLS) (Ruppert *et al.* 2003). Likelihood and covariance approximations can reduce the computational complexity of the matrix calculations (Stein, Chi, and Welty 2004; Furrer, Genton, and Nychka 2006), while the `gam()` function in the **mgcv** package in R uses the reduced rank thin plate spline approach of Wood (2004) fit by penalized IWLS. Rue and Tjelmeland (2002) exploit computationally efficient methods for fitting Markov random field (MRF) models by approximating stationary GPs using MRFs.

An alternative is to fit a Bayesian version of the model using a computationally efficient basis. The approach introduced by Wikle (2002) approximates a stationary GP structure for $g(\cdot; \boldsymbol{\theta})$ using a spectral representation to decompose the function in an orthogonal basis, in particular using Fourier basis functions and employing the FFT for fast computation (Wikle 2002; Royle and Wikle 2005; Paciorek 2007). While the Fourier basis approach has some adherents and is one of the few efficient alternatives within the Bayesian paradigm, the intricacies and bookkeeping involved in working with the complex-valued basis coefficients make it hard to simply apply the methodology and replicate results. My goal here is to present the representation in detail (Section 2), and provide an R (R Development Core Team 2006) package, **spectralGP**, (freely available from CRAN, `http://CRAN.R-project.org/`) for working with the representation that handles the bookkeeping and sampling of coefficients for use within Markov chain Monte Carlo (MCMC) (Section 3). I describe several parameterizations for exponential family data (Section 4), and discuss detailed MCMC implementation and mixing issues that arise in fitting models, as well as general recommendations on parameterizations and sampling techniques (Section 5). I note that my experience shows slower mixing than one would desire; advances in this area are an open area for research.

2. Fourier basis representation

To simplify the notation I use $\boldsymbol{g_s}$ to denote the vector of values calculated by evaluating $g(\cdot)$ for each of the elements of $\boldsymbol{s}$ (e.g., for each observation location), namely $\boldsymbol{g_s} = (g(\boldsymbol{s}_1), \ldots, g(\boldsymbol{s}_n))^\top$, suppressing the dependence on hyperparameters. Also, where necessary, I denote a set of unspecified parameters as $\boldsymbol{\theta}$. Proposal values are denoted with a *, e.g., $\boldsymbol{\theta}^*$, and vectors of augmented quantities with a tilde, e.g., $\tilde{\boldsymbol{Y}}$.

2.1. Basic process model

In many Bayesian models, the unknown function, be it a spatial surface or regression function, is a represented as a Gaussian process or by a basis function representation. Diggle, Tawn, and Moyeed (1998) formalized the idea of generalized geostatistical models, with a latent Gaussian spatial process, as the natural extension of kriging models to exponential family responses. They used Bayesian estimation, suggesting a Metropolis-Hastings implementation, with the spatial function sampled sequentially at each observation location at each MCMC iteration. However, as shown in their examples and discussed elsewhere (Christensen, Møller, and Waagepetersen 2000; Christensen and Waagepetersen 2002; Christensen, Roberts, and Sköld 2006), this implementation is slow to converge and mix, as well as being computationally inefficient because of the covariance matrix involved in calculating the prior for $\boldsymbol{g_s}$.

An alternative approach that avoids large matrix calculations is to express the unknown

function in a basis, $\boldsymbol{g_s} = \boldsymbol{\Psi u}$, where $\boldsymbol{\Psi}$ contains the basis function values evaluated at the locations of interest, and estimate the basis coefficients, $\boldsymbol{u}$. These coefficients are taken to have a prior distribution; constraints on the function, such as degrees of smoothness, are imposed through this prior distribution and the basis choice. When the coefficients are normally distributed, this representation can be viewed as a GP evaluated at a finite set of locations, with $\text{Cov}(\boldsymbol{g_s}) = \boldsymbol{\Psi}\text{Cov}(\boldsymbol{u})\boldsymbol{\Psi}^\top$.

Isotropic GPs can be represented in an approximate fashion using their spectral representation as a Fourier basis expansion, which allows one to use the Fast Fourier Transform (FFT) to speed calculations. Here I describe the basic model in two-dimensional space, following Wikle (2002).

The key to the spectral approach is to approximate the function $g(\cdot)$ on a grid, $\boldsymbol{s}^\#$, of size $M = M_1 \times M_2$, where M_1 and M_2 are powers of two. Evaluated at the grid points, the vector of function values is represented as

$$\boldsymbol{g_{s^\#}} = \boldsymbol{\Psi u}, \tag{2}$$

where $\boldsymbol{\Psi}$ is a matrix of orthogonal spectral basis functions, and $\boldsymbol{u}$ is a vector of complex-valued basis coefficients, $u_m = a_m + b_m i$, $m = 1, \ldots, M$. The spectral basis functions are complex exponential functions, i.e., sinusoidal functions of particular frequencies; constraints on the coefficients ensure that $\boldsymbol{g_{s^\#}}$ is real-valued and can be expressed equivalently as a sum of sine and cosine functions. The basis functions represented in the basis matrix, $\boldsymbol{\Psi}$, capture behavior at different frequencies, with the most important basis functions for function estimation being the low-frequency basis functions. To approximate mean zero stationary, isotropic GPs, the basis coefficients have the prior distribution,

$$\begin{pmatrix} \boldsymbol{a} \\ \boldsymbol{b} \end{pmatrix} \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{\Sigma_\theta}) \tag{3}$$

where the diagonal (asymptotically; see Shumway and Stoffer (2000, Section T3.12)) covariance matrix of the basis coefficients, $\boldsymbol{\Sigma_\theta}$, parameterized by $\boldsymbol{\theta}$, can be expressed in closed form (for certain covariance functions) using the spectral density of the covariance function desired to parameterize the approximated GP.

To make this more explicit, consider the Matérn covariance popular in spatial statistics,

$$C(\tau; \rho, \nu) = \sigma^2 \frac{1}{\Gamma(\nu) 2^{\nu-1}} \left(\frac{2\sqrt{\nu}\tau}{\rho} \right)^\nu \mathcal{K}_\nu \left(\frac{2\sqrt{\nu}\tau}{\rho} \right), \tag{4}$$

where τ is distance, σ^2 is the variance of the process, ρ is the range (correlation decay) parameter, and $\mathcal{K}_\nu(\cdot)$ is the modified Bessel function of the second kind, whose order is the differentiability parameter, $\nu > 0$. This covariance function has the desirable property that sample functions of GPs parameterized with the covariance are $\lfloor \nu - 1 \rfloor$ times differentiable. As $\nu \to \infty$, the Matérn approaches the squared exponential form, with infinitely many sample path derivatives, while for $\nu = 0.5$, the Matérn takes the exponential form with no sample path derivatives.

The spectral density of this covariance, which is used to calculate the elements of $\boldsymbol{\Sigma_\theta}$, evaluated at spectral frequency, $\boldsymbol{\omega}$, is

$$\phi(\boldsymbol{\omega}; \rho, \nu) = \sigma^2 \frac{\Gamma(\nu + \frac{D}{2})(4\nu)^\nu}{\pi^{\frac{D}{2}} \Gamma(\nu)(\pi\rho)^{2\nu}} \cdot \left(\frac{4\nu}{(\pi\rho)^2} + \boldsymbol{\omega}^\top \boldsymbol{\omega} \right)^{-\left(\nu + \frac{D}{2}\right)}, \tag{5}$$

where D is the dimension of the space (two in this case) and the parameters are as above. For an appropriate set of spectral frequencies, the diagonal elements of $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$ are the values of $\phi(\cdot; \rho, \nu)$ at those frequencies, and the off-diagonals are zero.

To construct real-valued processes with the approximate GP distribution based on the complex-valued coefficients given above, some detailed bookkeeping and constraints are required. These details are provided in the Appendix A.1, while details on accounting for the periodicity of the Fourier domain appear in Appendix A.2.

2.2. Computations and statistical modeling of observations

The process at the observation locations is calculated through an incidence matrix, $\boldsymbol{K}$, which maps each observation location to the nearest grid location in Euclidean space,

$$\boldsymbol{g_s} = \boldsymbol{K}\boldsymbol{g_{s\#}} = \boldsymbol{K}\boldsymbol{\Psi}\boldsymbol{u}. \tag{6}$$

For a fine grid, the error induced in associating observations with grid locations should be negligible and the piecewise constant representation of the surface tolerable. This approach amounts to binning the data, with square bins defined by the grid points. An extension would be to do local interpolation between the grid points to define the process values at the data locations, but I do not pursue that here. The computational efficiency comes in the fact that the matrix $\boldsymbol{\Psi}$, which is $M \times M$, need never be explicitly formed, and the operation $\boldsymbol{\Psi}\boldsymbol{u}$ is the inverse FFT, and so can be done very efficiently ($O\left(M \log_2(M)\right)$). In addition, evaluating the prior for $\boldsymbol{u}$ is fast because the coefficients are independent a priori. This stands in contrast to the standard MCMC setup for GP models, in which the prior on $\boldsymbol{g_s}$ involves an $n \times n$ matrix and therefore $O(n^3)$ operations. Of course the gridding could be done without the Fourier basis approach, but this would only reduce the computations to $O\left((M/2^D)^3\right)$ (the division by 2^D occurs because the padding of Appendix A.2 would not be required). Note that with the gridded approach, the number of observations affects the calculations only through the likelihood, which scales as $O(n)$, because the observations are independent conditional on $\boldsymbol{g_s}$. The complexity of the underlying surface determines the computational efficiency by defining how large M should be; simple surfaces can be estimated very efficiently even if n is large.

3. R spectralGP package

The **spectralGP** package for R provides an object-oriented representation of the Fourier basis approach to computation with GPs. The primary purpose of the package is to handle the bookkeeping details, allowing one to easily simulate and plot processes and sample the Fourier basis coefficients within an MCMC. This allows developers to write code to fit models in which a Gaussian process is a component, easily drawing MCMC samples of the coefficients of the process, as described in Section 3.3. Also note that C. Wikle has released Matlab code for the Fourier basis computations on his website (http://www.stat.missouri.edu/~wikle/).

3.1. Description of the package

The key functions in **spectralGP** are a constructor function, `gp()`, that creates a `gp` object, and a number of S3 methods: two MCMC sampling methods for the coefficients, `Gibbs.sample.coeff.gp()` and `propose.coeff.gp()`; a method for calculating the proposal

density of the current coefficients for use in Metropolis-Hastings sampling, `Hastings.coeff.gp()`; a method for simulating GPs, `simulate.gp()`; a method for changing the GP hyperparameter values, `change.param.gp()`; calculation of the logdensity of the coefficients, `logdensity.gp()`; and prediction and plotting methods. The function `updateprocess.gp()` updates the process values based on the current values of the coefficients, while `calc.variances.gp()` calculates the coefficient variances given the current covariance/spectral density parameters. These latter two functions are generally intended for internal use, but the template code in Appendix B makes use of `updateprocess.gp()` after changing the coefficient values during a particular type of MCMC step. Auxiliary functions allow for copying, `copy.gp()`; determining the basic grid used, `getgrid.gp()`; and extracting the object element names, `names.gp()`. Several functions deal with conversion between coordinate systems: a simplistic lon/lat to Euclidean x/y projection, `lonlat2xy()`; mapping a Euclidean domain to $(0,1)^D$, `xy2unit()`; and mapping locations in the domain to the closest grid points, `new.mapping()`. Several auxiliary functions are borrowed from the fields package, namely `rdist.earth()` as well as `image.plot()` and its auxiliary functions, `image.plot.info()` and `image.plot.plt()`.

I have used native R code for the entire package both for simplicity and because the essential computations within the package are already compiled code, namely the functions, `fft()`, `rnorm()`, and `dnorm()`.

spectralGP includes only the spectral density function for the Matérn covariance (and thereby the special case of the exponential covariance) but is easily extendible by writing user-defined spectral density functions that can be used in the constructor function for a creating a new process, `gp()`.

Some additional intricacies included in the package are mentioned parenthetically in Sections 4 and 5 and in Appendix A. Note that in the package, the process is scaled by $1/\sqrt{M_1 M_2}$, as described in Section 4.1, relative to the exact process values (21).

3.2. Using environments as objects to allow pass-by-reference

Since a `gp` object will be used repeatedly during MCMC sampling, I chose to use a pass-by-reference scheme in the coding, using R environments to mimic object-orientation in traditional languages, as suggested by E.A. Houseman. In this way, one can operate on a `gp` object and change internal elements without having to pass the entire object back from the method function and thereby create a new copy of the `gp` object or overwrite the old copy by assigning the output to the same name. This is possible because unlike other R objects, environments are not copied when passed to functions. Each instantiation of a `gp` object is an environment, initialized with a call to `new.env()` and assigned the class "gp". The elements of the object, e.g., `myFun` here, are local variables within the environment, accessed via list-like syntax, e.g., `myFun$process`. S3 methods are used to operate on the `gp` objects, with the difference from standard R that global changes can be made to the elements of an object within the method by virtue of those elements residing within an environment. For example, the call, `simulate.gp(myFun)`, samples new coefficients and updates the process based on those coefficients without having to pass `myFun` back to the calling environment, yet the changes to `myFun` are effective in the calling environment. Also note that care must be taken when assigning `gp` objects because environments are not copied when used in assignments; I have created an explicit `copy.gp()` function to make a new copy of a `gp` object; assignment merely creates an additional name (i.e., pointer) referencing the existing object, so any changes to

the original gp object also occur within the 'new' object, which just references the original object. I chose not to use S4 methods because my understanding is that their implementation is still somewhat slow and **spectralGP** works with large objects and substantial computation.

This use of R environments as objects that can be passed by reference is not a standard approach in R, which uses a pass-by-value system. As a result, the package can be nonintuitive to the user, producing changes in the objects as a side effect of the call to a method function, and then returning a null value. However, I believe this is a feature of the approach that is more natural for setting up MCMC sampling, rather than a drawback. There is a single copy of each parameter object that is changed during a sampling step instead of passing back a new copy of the changed object to the calling environment. Unlike most cases in R in which an object is created and then used as an input for additional work but is not modified subsequently (such as use of `lm()`, following by `summary()` and `plot()`), in the MCMC sampling setting, a parameter object needs to be modified repeatedly. It is more natural to pass by reference in this context.

3.3. Using and extending the package

A basic use of the package is to simulate and plot GPs. After setting up the gp object, one can simulate a realization and plot an image of the realization as follows:

```
myFun=gp(c(256,256),matern.specdens,c(0.3,2))
simulate(myFun)
plot(myFun)
```

This is computationally efficient because of the use of the FFT. The constructor function is

```
gp(gridsize,specdens,specdens.param,variance.param,const.fixed=FALSE)
```

The argument `gridsize` is a scalar for processes on $\Re^1$ or vector of length two for processes on $\Re^2$. If `gridsize=c(M1,M2)`, then the effective grid is $M_1/2+1$ by $M_2/2+1$, after accounting for periodicity of the Fourier basis, as discussed in Appendix A.2. Note that if $M_1 \neq M_2$, the scaling in the two dimensions is different, which produces an anisotropic process. The argument, `specdens`, is a spectral density function; currently only a Matérn function, `matern.specdens()`, with the exponential as a special case, is implemented but users can write their own. The argument, `specdens.param`, is a scalar or vector of parameters required by `specdens`; in the Matérn case the first value is the spatial range parameter, ρ, and the second the smoothness or differentiability parameter, ν. The argument, `variance.param`, is the spatial process variance, σ^2, while `const.fixed=TRUE` indicates whether the first basis coefficient, $u_{0,0}$, should be fixed to be zero, which eliminates a non-identifiability in the MCMC when a separate mean, μ, is included in the model (Appendix A.3); for simulation of realizations one should use `const.fixed=FALSE` to preserve the approximate specified covariance for the GP realizations. The `plot.gp()` method uses arguments similar to those in `image` while the only argument to `simulate.gp()` is the gp object itself. While fast simulation of GPs on a discrete grid may be of interest, the primary use of the package is to ease the sampling of the coefficients via MCMC in a Bayesian model context, as follows.

The simplest approach is to sample the coefficients in blocks grouped by basis function frequency, with the code

```
propose.coeff(object,block,proposal.sd)
```

where `block` specifies the block to be sampled (the blocks are created with `add.blocks.gp()` and using `block=0` proposes all of the coefficients at once) and `proposal.sd` is the user-tunable Metropolis-Hastings proposal standard deviation. The user then needs to write code to accept or reject the proposal by calculating the probability density of any model quantities that depend on the process values (extracted using `predict.gp()`) and the prior density of the coefficients using `logdensity.gp()`. `predict.gp()` by default makes predictions on the effective grid of size $M_1/2+1$ by $M_2/2+1$, but can take the argument `newdata` as a matrix of prediction locations or the argument `mapping` as a set of indices indicating the grid cells whose values should be extracted. The set of indices can be created using the method `new.mapping(object,locations)`, where `locations` is a two-column matrix of location coordinates. More details on this approach are given in Section 4.3, and Appendix B provides template code illustrating this usage.

In certain models, the coefficients can be sampled by a Gibbs sampling step from their exact conditional distribution. The following conditional distribution for the coefficients,

$$\begin{aligned} \boldsymbol{u}|\cdot &\sim \mathcal{N}\left(\boldsymbol{V}\frac{s}{\sigma_e^2}\boldsymbol{\Psi}^\top(\boldsymbol{z}-m\mathbf{1}),\boldsymbol{V}\right) \\ \boldsymbol{V} &= \left(\frac{s^2}{\sigma_e^2}\boldsymbol{I}+{\boldsymbol{\Sigma}_{\boldsymbol{\theta}}}^{-1}\right)^{-1}, \end{aligned} \tag{7}$$

arises when the coefficients have prior distribution, $\boldsymbol{u}\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma}_{\boldsymbol{\theta}})$ as in (3) and the gridded data or latent process values whose distribution depends on the GP have the distribution, $\boldsymbol{z}\sim\mathcal{N}(m\mathbf{1}+s\boldsymbol{\Psi}^\top(\boldsymbol{z}-m\mathbf{1}),\sigma_e^2\boldsymbol{I})$. Here m is a mean term, s is a standard deviation term, and σ_e^2 is a residual variance term. Section 4 shows how this conditional distribution arises in several model structures for non-gridded data. Sampling is done with the code

```
Gibbs.sample.coeff(object,z,sig2e,meanVal,sdVal,returnHastings=FALSE)
```

where the mapping of the method arguments to the parameters above should be clear. Note that the prior variance structure of the coefficients, $\boldsymbol{\Sigma}_{\boldsymbol{\theta}}$, is already part of the `gp` object. One can also sample coefficients from the distribution (7) even if it is not the exact conditional distribution and base acceptance on the Metropolis-Hastings algorithm. One can calculate the log proposal density, $\pi(\boldsymbol{u}|\cdot)$, using

```
Hastings.coeff.gp(object,z,sig2e,meanVal,sdVal)
```

or as the return value of `Gibbs.sample.coeff()` with the argument `returnHastings=TRUE`. To calculate the ratio of proposal densities for the Metropolis-Hastings algorithm, one would first calculate the log proposal density of the current (original) coefficients as the output, `oldDensity`, from

```
oldDensity=Hastings.coeff.gp(object,z,sig2e,meanVal,sdVal)
```

Then, possibly after changing hyperparameter values as part of a joint proposal of hyperparameters and coefficients (Section 5.2) one makes the proposal, $\boldsymbol{u}^*$, using

```
newDensity=Gibbs.sample.coeff(object,z,sig2e,meanVal,sdVal,returnHastings=TRUE)
```

which assigns the log proposal density of the proposed coefficients to `newDensity`. The difference of `oldDensity` and `newDensity` is then used as part of the Metropolis-Hastings acceptance decision.

Together these functions allow a developer to easily sample the coefficients within the context of a larger model without worrying about the bookkeeping and technical details of the Fourier basis representation. In Appendix B I provide R template code for various forms (Section 4) of the simple model (1) for data from an exponential family distribution: normal, Poisson, or binomial. The code assumes a constant mean, μ, in place of $\boldsymbol{x}_i^\top \boldsymbol{\beta}$, and Matérn covariance function with ν fixed. While this is a simple setting, users could take the template code and use it within a more complicated hiearchical model or easily extend to a regression structure in the mean. For example, Paciorek and McLachlan (2007) use a set of `gp` objects to represent compositional data in a spatial setting with a complicated multinomial-Dirichlet likelihood based on a transformation of the GPs, making use of the block sampling approach. The template code includes code for joint sampling of covariance parameters and process coefficients, which I discuss in Section 5.2 as a way to improve mixing. Developers could build on the template code to include this type of sampling in their own models.

4. Basic MCMC sampling schemes for coefficients

In this section, I describe several parameterizations for simple Bayesian exponential family models with associated MCMC sampling schemes for the Fourier basis coefficients. Bayesian estimation of unknown processes in this framework relies on shrinkage to estimate the large number of coefficients; coefficients of low-frequency basis functions are strongly informed by the data while those of high-frequency basis functions are shrunk strongly toward their prior distributions. These following basic parameterizations can also be used with relatively straightforward modifications in more complicated hierarchical models, although the added complexity may make the simple blocked sampling scheme (Section 4.3) the most feasible approach in that case. Note that for simplicity I consider a scalar mean parameter, μ, but this could be replaced by a regression term, e.g., $\boldsymbol{X_i\beta}$, or other additive components. Also, for more compact notation, I suppress the dependence of $\boldsymbol{\Sigma_\theta}$ on the covariance/spectral density parameters, $\boldsymbol{\theta}$.

4.1. Data augmentation Gibbs sampling for normal data

For Gaussian data with mean function based on the latent process, $g(\cdot)$, a missing data scheme allows for Gibbs sampling of the coefficients. This is a simplification of the Gibbs sampling scheme of Wikle (2002), which is described in Section 4.2.

Take the data model to be

$$\boldsymbol{Y} \sim \mathcal{N}_n(\mu\mathbf{1} + \gamma \boldsymbol{K\Psi u}, \eta^2 \boldsymbol{I}), \tag{8}$$

where μ is the process mean and γ is the process standard deviation with σ^2 in (4-5) set to one. Since the prior for the coefficients is normal, we have conjugacy, and the conditional distribution for $\boldsymbol{u}$ is

$$\boldsymbol{u}|\boldsymbol{y}, \cdot \;\sim\; \mathcal{N}_M\left(\boldsymbol{V}\frac{\gamma}{\eta^2}(\boldsymbol{K\Psi})^\top(\boldsymbol{y} - \mu\mathbf{1}), \boldsymbol{V}\right)$$

$$\boldsymbol{V} = \left(\frac{\gamma^2}{\eta^2}\boldsymbol{\Psi}^\top \boldsymbol{K}^\top \boldsymbol{K}\boldsymbol{\Psi} + \boldsymbol{\Sigma}^{-1}\right)^{-1}. \tag{9}$$

The sample of $\boldsymbol{u}$ represents precision-weighted shrinkage of the data-driven estimates of the coefficients towards their prior mean of zero.

However, this sampling scheme requires calculation of $\boldsymbol{\Psi}^\top \boldsymbol{K}^\top \boldsymbol{K}\boldsymbol{\Psi}$, which is not feasible for large number of grid points; note that if $\boldsymbol{K}$ were the identity of dimension M, since $\boldsymbol{\Psi}$ is an orthogonal matrix, this simplifies to

$$\boldsymbol{V} = \left(\frac{\gamma^2}{\eta^2}\boldsymbol{I} + \boldsymbol{\Sigma}^{-1}\right)^{-1}, \tag{10}$$

which because $\boldsymbol{\Sigma}$ is diagonal, is easy to calculate. Assuming no more than one observation per grid cell, $\boldsymbol{K} = \boldsymbol{I}$ can be achieved using a missing data scheme by introducing latent pseudo-observations for all grid cells without any associated data, including grid cells in which no data can possibly fall because of padding to account for periodicity (Appendix A.2). Collecting these pseudo-observations into a vector, $\tilde{\boldsymbol{Y}}$, they can be sampled within the MCMC using a Gibbs step as

$$\tilde{\boldsymbol{Y}} \sim \mathcal{N}_{M-n}(\mu\boldsymbol{1} + \gamma\tilde{\boldsymbol{K}}\boldsymbol{\Psi}\boldsymbol{u}, \eta^2\boldsymbol{I}), \tag{11}$$

where the $\tilde{\boldsymbol{K}}$ matrix picks out grid cells with no associated data. With this augmentation, $\boldsymbol{Y}$ in (8-9) is a vector of values on the full grid, $\boldsymbol{Y} = (\boldsymbol{Y}_{obs}, \tilde{\boldsymbol{Y}})$, combining actual observations with pseudo-observations, and $\boldsymbol{K} = \boldsymbol{I}$.

The coefficients can be sampled under this approach (9-10) using

```
Gibbs.sample.coeff.gp(object,z,sig2e,meanVal,sdVal)
```

where `z` is $\boldsymbol{Y}$, `sig2e` is η^2, `meanVal` is μ, and `sdVal` is γ. Note that this straightforward expression conceals some details required in working with the complex-valued coefficients. In calculating $(\gamma^2/\eta^2 + (\boldsymbol{\Sigma}^{-1})_{ii})$, one needs to multiple γ^2/η^2 by one-half for all the elements corresponding to complex-valued coefficients to ensure that the scaling is correct as described in Appendix A.1. Also, the operation $\boldsymbol{\Psi}^\top(\boldsymbol{y} - \mu\boldsymbol{1})$ is the FFT and the correct scaling needs to occur so the result is on the scale of the coefficients. In R, I specify the coefficient variances as $M_1 M_2 \phi(\boldsymbol{\omega}; \boldsymbol{\theta})$ and update the process as $\boldsymbol{\Psi}\boldsymbol{u}/\sqrt{M_1 M_2}$. If I then divide $\boldsymbol{\Psi}^\top(\boldsymbol{y} - \mu\boldsymbol{1})$ by $\sqrt{M_1 M_2}$ when sampling the coefficients (9-10), the desired approximate covariance structure for the process is preserved, namely $\boldsymbol{\Psi}\boldsymbol{u} \dot{\sim} \mathcal{N}_M(\boldsymbol{0}, \boldsymbol{C}_{\boldsymbol{\theta}})$, where the matrix, $\boldsymbol{C}_{\boldsymbol{\theta}}$ is defined by $\boldsymbol{C}_{\boldsymbol{\theta},ij} = C(\| \boldsymbol{s}_i - \boldsymbol{s}_j \|; \boldsymbol{\theta})$ and $C(\cdot; \boldsymbol{\theta})$ is the covariance function whose spectral density defines $\phi(\cdot; \boldsymbol{\theta})$, e.g., (5). The exact algorithm is given in the `Gibbs.sample.coeff.gp()` function in the **spectralGP** package. In Appendix B I provide template code for fitting this parameterization, denoted as Code A.

If there is more than one observation per grid cell, some possible solutions are to use a finer grid or to take $y(s_j) = \bar{y}_j$, namely the average of the observations in the grid cell. Ideally, one would set $\eta_j^2 = \eta^2/n_j$, but this would require calculation of $\boldsymbol{\Psi}^\top \boldsymbol{\eta}^{-1} \boldsymbol{\Psi}$, where $\boldsymbol{\eta} = \text{diag}((\eta_1^2, \ldots, \eta_M^2))$, which is computationally infeasible. Instead, I suggest using constant η^2 so long as there are relatively few locations with multiple observations per grid cell. One could also use more extensive data augmentation to supplement the existing observations such

that there are n_j pseudo plus true observations per grid cell, with n_j equal to the maximum number of true observations in a cell over all of the grid cells.

Wikle (2002) recommends the uncentered (sensu Gelfand, Sahu, and Carlin 1996) parameterization for the process variance (8), with γ allowed to vary and $\sigma^2 \equiv 1$ in defining the variance of the coefficients (5). He notes that moving the parameter closer to the data improves mixing and helps avoid dependence with ρ. Note that I follow this approach in some cases, while in others, I allow σ^2 to vary and fix $\gamma \equiv 1$. In the **spectralGP** package, a value of σ^2 not equal to one is specified with the `variance.param` argument to `gp()` and the `new.variance.param` argument to `change.param.gp()`.

For non-normal data from the exponential family, $Y_i \sim \mathcal{F}(h^{-1}(f_i))$, where $f_i = \mu + \gamma \boldsymbol{K_i \Psi u}$, one might use a Metropolis-Hastings version of this Gibbs sampling scheme, again with data augmentation, making use of the linearized observations and working variances used in fitting GLMs in place of $\boldsymbol{y}$ and η^2 in (9-10). As above, one would need to use a single homoscedastic value for η^2. I do not pursue this approach for non-normal data further, as I had little success in tuning the proposals to achieve reasonable acceptance, but further research in this area may be worthwhile.

One approach to speeding mixing in the case of normal data is to jointly sample η^2 and $\tilde{\boldsymbol{Y}}$ by first proposing η^{2*} and then, within the same proposal, sampling from $[\tilde{\boldsymbol{Y}}|\eta^{2*}, \cdot]$ via a Gibbs sample, conditional on the proposed value, η^{2*}. Because this joint sample is not from the joint full conditional $[\tilde{\boldsymbol{Y}}, \eta^2|\cdot]$, we need to use Metropolis-Hastings in determining acceptance based on the ratio of the prior for η^2 and likelihood, $\pi(\eta^{2*})L(\boldsymbol{Y}_{obs}|\eta^{2*}, \cdot)/(\pi(\eta^2)L(\boldsymbol{Y}_{obs}|\eta^2, \cdot))$, where $\boldsymbol{Y}_{obs}$ is the actual data. Acceptance does not depend on the value of the augmented observations, $\tilde{\boldsymbol{Y}}$. So one can propose η^2, decide on acceptance based on the likelihood of the true observations, and then, if accepted, do a Gibbs sample for $\tilde{\boldsymbol{Y}}$ (11). We have effectively integrated $\tilde{\boldsymbol{Y}}$ out of the joint conditional density, $\pi(\eta^2, \tilde{\boldsymbol{Y}}|\boldsymbol{y}_{obs}, \cdot)$, thereby sampling η^2 without dependence on $\tilde{\boldsymbol{Y}}$ (Rue and Held 2005, pp. 141-143). In the iterations, one may also wish to do a separate Gibbs sample for $\tilde{\boldsymbol{Y}}$ alone, apart from the joint sample with η^2. Template code A in Appendix B also includes modifications for this sampling approach.

4.2. Latent layer Gibbs sampling for exponential family data

Parameterizing with two latent layers (the Wikle parameterization)

For non-normal data, rather than losing the Gibbs sampling structure for the coefficients, Wikle (2002) and Royle and Wikle (2005) embed the spectral basis representation in a hierarchical model with additional latent processes and associated variance components. This approach allows one to do Gibbs sampling in various generalized models in which exponential family outcomes are related to a latent spatial process in the mean structure (1).

To take a concrete example, the model for Poisson data is

$$\begin{aligned} Y_i &\sim \mathcal{P}(\exp(\lambda_i)) \\ \lambda_i &\sim \mathcal{N}(\mu + \gamma \boldsymbol{K_i z}, \eta^2) \\ \boldsymbol{z} &\sim \mathcal{N}_M(\boldsymbol{\Psi u}, \sigma_z^2 \boldsymbol{I}), \end{aligned} \tag{12}$$

where $\boldsymbol{\Psi u}$ is the Fourier basis representation with the prior structure (3). One can easily modify the likelihood and link for other exponential family distributions. The model introduces two variance components, η^2 and σ_z^2, corresponding to two latent processes, one, $\boldsymbol{\lambda}$,

defined only at the observation locations, and the second, $\boldsymbol{z}$, defined for each of the grid cells, including those in which no data can fall, as discussed in Appendix A.2. Note that the variance components account for overdispersion. In Section 5.1, I discuss issues that arise when the data are not overdispersed.

Wikle (2002) suggests a Metropolis-Hastings proposal for $\boldsymbol{\lambda}$, with conjugate normal Gibbs sampling for $\boldsymbol{z}$ and $\boldsymbol{u}$:

$$\begin{aligned}
\boldsymbol{z}|\boldsymbol{\lambda},\cdot &\sim \mathcal{N}_M\left(\boldsymbol{V}_z\left(\frac{\gamma}{\eta^2}\boldsymbol{K}^\top(\boldsymbol{\lambda}-\mu\mathbf{1})+\sigma_z^{-2}\boldsymbol{\Psi}\boldsymbol{u}\right),\boldsymbol{V}_{\boldsymbol{z}}\right)\\
\boldsymbol{V}_{\boldsymbol{z}} &= \left(\frac{\gamma^2}{\eta^2}\boldsymbol{K}^\top\boldsymbol{K}+\sigma_z^{-2}\boldsymbol{I}\right)^{-1}\\
\boldsymbol{u}|\boldsymbol{z},\cdot &\sim \mathcal{N}_M\left(\boldsymbol{V}_{\boldsymbol{u}}\frac{\boldsymbol{\Psi}^\top\boldsymbol{z}}{\sigma_z^2},\boldsymbol{V}_{\boldsymbol{u}}\right)\\
\boldsymbol{V}_{\boldsymbol{u}} &= (\sigma_z^{-2}\boldsymbol{I}+\boldsymbol{\Sigma}^{-1})^{-1}
\end{aligned} \tag{13}$$

Similar calculations to those in the previous section are needed in the Gibbs sampling for $\boldsymbol{u}$ to account for the complex-valued coefficients and to scale the proposal correctly. In the **spectralGP** package, $\boldsymbol{u}$ is sampled using

```
Gibbs.sample.coeff.gp(object,z,sig2e,meanVal,sdVal)
```

where `z` is $\boldsymbol{z}$, `sig2e` is σ_z^2, `meanVal` is 0, and `sdVal` is 1. Template code is given in Appendix B as Code B.

Note that sampling can require long chain lengths; Royle and Wikle (2005) used eight chains of length 520,000, retaining every 50th iteration, which suggest slow mixing of the sort I have experienced as well.

A simplified parameterization with a single latent layer (modified Wikle parameterization)

I propose a modification of the model above to eliminate one of the latent layers, thereby moving the coefficients closer to the data in the hierarchy and eliminating $\boldsymbol{z}$ and σ_z^2, which can be difficult to interpret and may not informed by the data (see Section 5.1). The simplified model for Poisson data is

$$\begin{aligned}
Y_i &\sim \mathcal{P}(\exp(\boldsymbol{K}_{\boldsymbol{i}}\boldsymbol{\lambda}))\\
\boldsymbol{\lambda} &\sim \mathcal{N}_M(\mu\mathbf{1}+\gamma\boldsymbol{\Psi}\boldsymbol{u},\eta^2\boldsymbol{I})
\end{aligned} \tag{14}$$

where the ith row of $\boldsymbol{K}$ maps the observation to the grid cell in which it falls. One can easily modify the likelihood and link for other exponential family distributions. Here η^2 accounts for overdispersion. Inference about the unknown smooth function should be based on $\mu\mathbf{1}+\gamma\boldsymbol{\Psi}\boldsymbol{u}$ rather than $\boldsymbol{\lambda}$, as simulations indicate that inference based on $\boldsymbol{\lambda}$ has larger posterior variances and is overly conservative for the unknown mean function because $\boldsymbol{\lambda}$ includes heterogeneity from overdispersion.

One can use Gibbs sampling for the values of $\boldsymbol{\lambda}$ corresponding to the J grid cells with no observations, denoted $\tilde{\boldsymbol{\lambda}}$, and for $\boldsymbol{u}$:

$$\begin{aligned}
\tilde{\boldsymbol{\lambda}}|\boldsymbol{u},\cdot &\sim \mathcal{N}_J\left(\mu\mathbf{1}+\gamma\tilde{\boldsymbol{K}}\boldsymbol{\Psi}\boldsymbol{u},\eta^2\boldsymbol{I}\right)\\
\boldsymbol{u}|\boldsymbol{\lambda},\cdot &\sim \mathcal{N}_M\left(\boldsymbol{V}_u\frac{\gamma}{\eta^2}\boldsymbol{\Psi}^\top(\boldsymbol{\lambda}-\mu\mathbf{1}),\boldsymbol{V}_{\boldsymbol{u}}\right)\\
\boldsymbol{V}_{\boldsymbol{u}} &= \left(\frac{\gamma^2}{\eta^2}\boldsymbol{I}+\boldsymbol{\Sigma}^{-1}\right)^{-1}.
\end{aligned} \tag{15}$$

Again, $\boldsymbol{u}$ can be sampled with

```
Gibbs.sample.coeff.gp(object,z,sig2e,meanVal,sdVal)
```

where `z` is $\boldsymbol{\lambda}$, `sig2e` is η^2, `meanVal` is μ, and `sdVal` is γ. For the elements of $\boldsymbol{\lambda}$ corresponding to grid cells in which observations fall, $\boldsymbol{\lambda}_{obs}$, I suggest Metropolis proposals, done individually for each individual element, but computed in an efficient vectorized fashion in R. Some intuition for how the information from the data diffuses to the level of the basis coefficients is that the latent layer, $\boldsymbol{\lambda}$, allows for some fluidity between the process values and the data: individual sampling of $\boldsymbol{\lambda}_{obs}$ for individual grid cells allows the latent layer to accomodate the data based on adjustments to $\boldsymbol{\lambda}_{obs}$ at individual locations, while the Gibbs sample of $\boldsymbol{u}$ translates these adjustments to the coefficients. A single joint sample for the elements of $\boldsymbol{\lambda}_{obs}$ would likely have slower mixing as it would be trying to sample many grid locations at once, with a single acceptance decision, thereby slowing local adjustments. Template code C is given in Appendix B.

In similar fashion to joint sampling of $(\eta^2,\tilde{\boldsymbol{Y}})$ in Section 4.1, with the parameterization above, one can improve mixing by jointly sampling η^2 and $\tilde{\boldsymbol{\lambda}}$. First propose η^{2*} and then, within the same proposal, sample from $[\tilde{\boldsymbol{\lambda}}|\eta^{2*},\cdot]$. Because this joint sample is not from the joint conditional of $(\eta^2,\tilde{\boldsymbol{\lambda}})$, we need a Metropolis-Hastings acceptance decision based on the ratio of the prior for η^2 and likelihood, $\pi(\eta^{2*})L(\boldsymbol{Y}|\eta^{2*},\boldsymbol{\lambda}_{obs},\cdot)/(\pi(\eta^2)L(\boldsymbol{Y}|\eta^2,\boldsymbol{\lambda}_{obs},\cdot))$, with acceptance not depending on the value for the augmented locations, $\tilde{\boldsymbol{\lambda}}$, thereby effectively integrating $\tilde{\boldsymbol{\lambda}}$ out of the joint conditional density, $\pi(\eta^2,\tilde{\boldsymbol{\lambda}}|\boldsymbol{\lambda}_{obs},\boldsymbol{y},\cdot)$. So in practice one can propose η^{2*}, decide on acceptance, and then, if accepted, do a Gibbs sample for $\tilde{\boldsymbol{\lambda}}$ (15). In the iterations, one may also wish to do a separate Gibbs sample for $\tilde{\boldsymbol{\lambda}}$ alone. Template code C in Appendix B also includes modifications for joint sampling of $(\eta^2,\tilde{\boldsymbol{\lambda}})$.

4.3. Blocked Metropolis sampling for exponential family data (simple parameterization)

An alternative to Gibbs sampling that avoids the use of the additional hierarchical layers and variance components in Section 4.2 is a simple model with straightforward Metropolis sampling for the coefficients described in Paciorek (2007). This approach has the advantage of tying the coefficients to the data by involving the coefficients directly in the likelihood, without intervening layers. For data that are not overdispersed, the simple model avoids introducing the overdispersion parameter(s), η^2 (and σ_z^2).

The basic approach is to specify the obvious parameterization in which the data are directly dependent on the latent spatial surface, which for Poisson data is

$$Y_i \sim \mathcal{P}(\exp(\mu+\gamma\boldsymbol{K}_{\boldsymbol{i}}\boldsymbol{\Psi}\boldsymbol{u})). \tag{16}$$

I suggest sampling the coefficients in a blocked Metropolis scheme, with blocks of coefficients whose corresponding frequencies have similar magnitudes (Paciorek 2007). I use smaller blocks for the low-frequency coefficients, thereby allowing these critical coefficients to move more quickly. The high-frequency coefficients have little effect on the function and are proposed in large blocks. The first block is the scalar, $u_{0,0}$, corresponding to the frequency pair, $(\omega_0^1, \omega_0^2) = (0, 0)$ (but note that in Appendix A.3 I suggest not sampling this coefficient because of lack of identifiability with respect to μ). The remaining blocks are specified so that the block size increases as the frequencies increase. For example, the next block might include the coefficients whose largest magnitude frequencies are at most one, i.e., u_{m_1,m_2} s.t. $\max\{|\omega_{m_1}^1|, |\omega_{m_2}^2|\} \leq 1$, but excluding the previous block, giving the block of coefficients, $\{u_{0,1}, u_{1,0}, u_{1,1}, u_{M_1-1,1}\}$. Recall that there are additional coefficients whose largest magnitude frequencies are at most one, e.g., $u_{M_1-1,0}$ and u_{M_1-1,M_2-1}, but these are complex conjugates of the sampled coefficients. The next block might be the coefficients whose largest magnitude frequencies are at most two, i.e., u_{m_1,m_2} s.t. $\max\{|\omega_{m_1}^1|, |\omega_{m_2}^2|\} \leq 2$, but excluding the previous block elements. The real and imaginary components of the coefficients in each block are proposed jointly, Metropolis-style, from a multivariate normal distribution with independent elements whose means are the current values. Since the coefficients have widely-varying scales, I take the proposal variance for each coefficient to be the product of a tuneable multiplier (one for each block, the `proposal.sd` argument to `propose.coeff.gp()`) and the prior variance of the coefficient, which puts the proposal on the proper scale. In the `add.blocks.gp()` function in **spectralGP** package, the default blocks are set by grouping coefficients based on the frequency thresholds, $0, 1, 2, 4, \ldots, 2^Q$, where $Q = \log_2(\max(M_1, M_2)) - 1$. The coefficients can be proposed in **spectralGP** using `propose.coeff.gp(object,block,proposal.sd)`, with the numbered block to be proposed specified as the argument `block`. Template code for fitting by this approach is given in Appendix B as Code D.

4.4. Hyperparameter priors

Assuming the Matérn covariance (4-5), the hyperparameters in the various parameterizations are $\boldsymbol{\theta} = (\mu, \gamma, \rho, \nu, \eta^2, \sigma_z^2)$, with η^2 and σ_z^2 not present in some cases. Royle and Wikle (2005) use inverse gamma priors for η^2 and σ_z^2 with a diffuse normal prior for μ. For ρ they use the reference prior of Berger, De Oliveira, and Sansó (2001) for the Gaussian likelihood case to avoid the use of an improper diffuse prior that could lead to an improper posterior. Paciorek (2007) specifies independent, proper, but non-informative priors for the elements of $\boldsymbol{\theta}$, except for ν, which cannot be estimated for a grid-level process (even for the continuous case, this is difficult to estimate without some observations very close together) and which is fixed in advance ($\nu = 4$ gives smooth processes with a small number of derivatives).

Gelman (2006) suggests truncated uniform and folded non-central t distributions on the standard deviation scale for variance components and argues against $\mathcal{IG}(\epsilon, \epsilon)$ priors as these have a sharp peak at small values that can strongly affect inference when the data do not constrain the parameters away from zero. Berger *et al.* (2001) argue for reference priors and against proper but non-informative priors, including truncated distributions, in part because they are concerned about the posterior concentrating at extreme values or on the truncation limit. In the setting here, I believe that the exact form of the priors is not critical, except that it is desirable to keep the parameters in a finite interval to prevent them from wandering in extreme parts of the space in which the likelihood is flat. In cases with sufficient data,

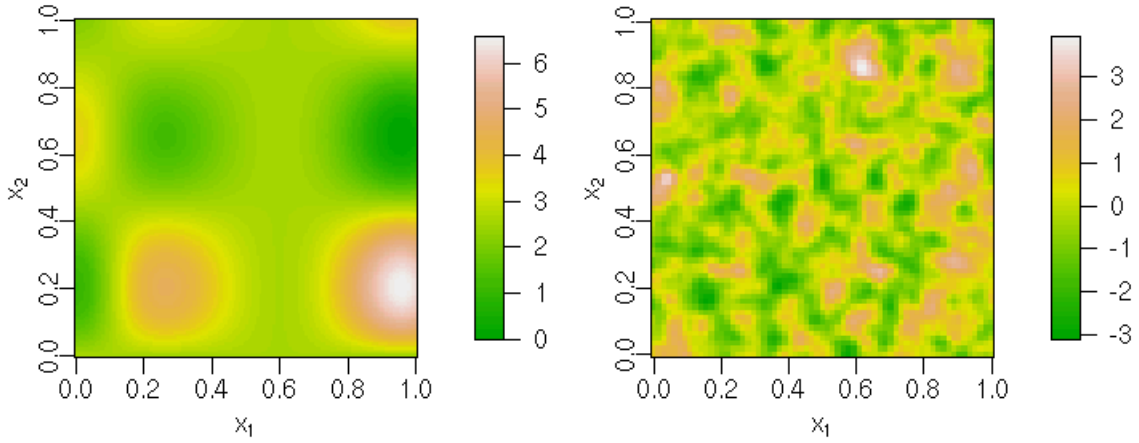

Figure 1: Mean functions used in simulated datasets: simple function (left) with $\hat{\rho} \approx 0.3$ and more complicated GP (right) with $\rho = 0.05$.

the prior should play little role in estimation and prediction. The situations that concern Berger *et al.* (2001) with regard to truncation and vague proper priors arise when the data provide little information, in which case their concern about the posterior concentrating on the truncation limit seems little different than having it constrained by the reference prior. I discuss identifiability and priors for σ_z^2 and η^2 in more detail in Section 5.1.

5. MCMC sampling considerations

In Sections 5.1-5.2, I describe some factors that impede mixing and some modifications to the basic sampling schemes discussed above that can help to improve mixing. In Section 5.3 I compare the approaches of Section 4 and make broad recommendations. Appendix A.3 discusses the lack of identifiability of $u_{0,0}$ with respect to an overall mean, μ, Appendix A.4 considers reparameterizing the covariance parameters for improved mixing, and starting values are discussed in Appendix A.5. Note that for any particular application, my recommendations may not provide the best mixing, and consideration of alternatives discussed in this paper may improve matters.

I explored sampling effectiveness using a few simulated datasets, meant to provide a range of function complexity and data intensity. All have Poisson data with the sampling locations sampled uniformly in $(0, 1)^2$: Data1 has 225 observations while Data2 has 1000 observations, both with the mean function, $f(x_1, x_2) = 1.9 \cdot (1.35 + \exp(x_1)\sin(13 \cdot (x_1 - 0.6)^2) \cdot \exp(-x_2)\sin(7x_2))$, used by Hwang, Lay, Maechler, Martin, and Schimert (1994) (Figure 1), a fairly simple function that when fit with a GP has $\hat{\rho} \approx 0.3$. Data3, Data4, and Data5 use the same mean function; a GP with $\rho = 0.05$, $\mu = 0$, $\gamma = \sigma = 1$ (Figure 1); and 400, 800, and 2500 observations, respectively (Figure 1).

I fit the data with the various parameterizations and sampling schemes using MCMC with a burn-in of 10,000 iterations and runs of 100,000 additional iterations. To assess mixing speed, I considered the trace plots, autocorrelations, and effective sample sizes (ESS) (Neal 1993, p.

105),

$$\text{ESS} = \frac{T}{1 + 2\sum_{d=1}^{\infty} \text{Cor}_d(\theta)}, \tag{17}$$

where $\text{Cor}_d(\theta)$ is the autocorrelation at lag d for a given posterior quantity, θ, truncating the summation at the lesser of $d = 10000$ or the largest d such that $\text{Cor}_d(\theta) > 0.05$. I focus on ESS for 1.) the overall log posterior density, $\pi(\cdot|\boldsymbol{y})$ (as suggested in Cowles and Carlin (1996) and calculated up to the normalizing constant), 2.) the critical smoothing parameter, ρ, and 3.) a random subset of 200 function estimates.

5.1. Variance component magnitudes and mixing speed

Here I discuss how the magnitude of a key variance component influences mixing.

The influence of the error variance in the normal model

Under the normal model (8), as $\eta^2 \to 0$, we have an interpolating surface that passes through the observations. In spatial statistics, such interpolation may arise relatively frequently when measurements are made with little measurement error or there is little fine-scale heterogeneity (Cressie 1993, p. 59). However, a key sampling consideration arising from small values of η^2 is that the size of MCMC moves for the basis coefficients is quite small. As $\eta^2 \to 0$,

$$V(u_i|\eta^2, \cdot) = \left(\frac{\gamma^2}{\eta^2} + \Sigma_{ii}^{-1}\right)^{-1} \to 0 \tag{18}$$

for the Gibbs sample proposal variance (10). For the coefficients of low-frequency basis functions, which most influence the process estimate, as η^2 get small, the proposal variance is a small fraction of the magnitude of the coefficient (Figure 2). When $\eta^2 \approx 0$, the process estimates are specified nearly exactly at the observation locations, and any proposal at those locations is constrained by the observations. This constrains the proposal for the entire spatial process. Mixing can be challenging even if uncertainty away from the observations is substantial and of real interest. While this issue seems likely to arise in other GP representations, except when the process values can be integrated out of the model, the issue is particularly clear with the spectral representation.

The influence of the dispersion parameter(s) in the exponential family model

Under the single latent layer model of Section 4.2, the magnitude of the variance component, η^2, that accounts for overdispersion affects the proposal variances for the coefficients in similar fashion. When there is not overdispersion, the posterior for η^2 should concentrate near zero. While this might be the correct inference, if the value of η^2 does approach zero in the MCMC sampling, the chain will mix very slowly, as in the case of normal data, because in (15) $V(u_i|\eta^2, \cdot) \to 0$ as $\eta^2 \to 0$ as in (18). Small values of η^2 result in small proposal variances and slow movement of the coefficients. Figure 3a shows the ESS for the log posterior density and for a sample of function values as a function of fixed η^2 for Data3, Poisson data generated without overdispersion.

In the case of overdispersion, the data can inform η^2, and inverse gamma prior distributions such as those of Wikle (2002) and Royle and Wikle (2005) may suffice. When there is little overdispersion, these priors are more problematic. Note that the inverse gamma prior has

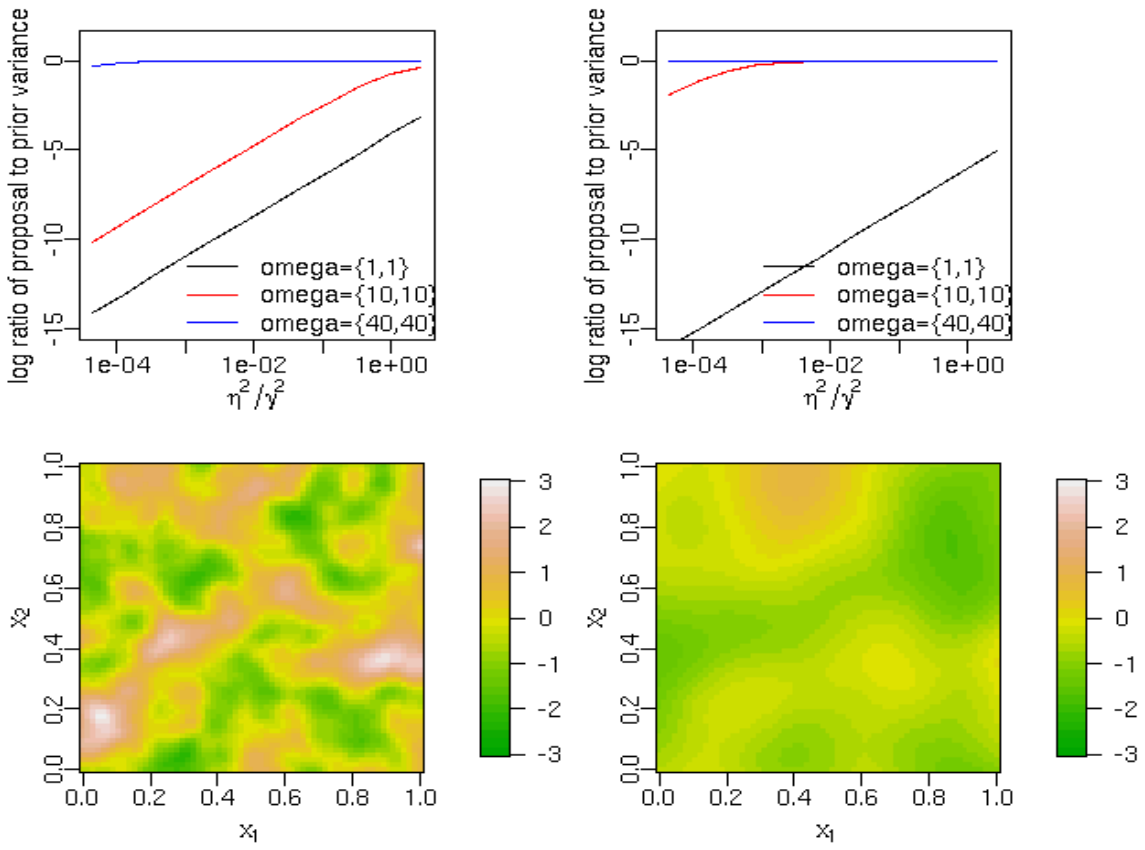

Figure 2: Decay of proposal variances (scaled relative to prior variance) for three representative coefficients as a function of the relative error variance (η^2/γ^2) for (left column) GPs with $\rho = 0.1$ and (right column) GPs with $\rho = 0.4$. Example process realizations are shown in the lower row.

a rapidly-decaying left tail, dropping off as $\exp(-1/x)$, so the inverse gamma prior assigns no mass to small values of η^2, preventing the posterior from having mass in this area. For example, the $\mathcal{IG}(0.5, 2)$ prior has very little mass at values less than 0.05 while the $\mathcal{IG}(1, 10)$ has very little mass at values less than 0.006. Fitting the model of Section 4.2 to Data2 with the $\mathcal{IG}(0.5, 2)$ prior shows that the constraints imposed by the prior cause the posterior to have its mass in the extreme lower range of the prior, while using a truncated lognormal prior results in most of the posterior mass lying at very small values of η^2 near the truncation point (Figure 3b). This suggests that when the observations exhibit little overdispersion, the prior has substantial impact on the posterior for η^2.

A prior that weights the model away from small values of η^2 has the desirable impact of improving mixing at the cost of forcing overdispersion, while use of a prior that allows for small values of η^2 carries the risk of very slow mixing. This suggests that we might choose a prior or even fix η^2 in advance to achieve optimal mixing, treating η^2 as an MCMC tuning parameter. The danger of using large values of η^2 is that while mixing will improve, the posterior variances of key quantities such as the function estimates will increase, with overly conservative inference and poor predictive ability because of oversmoothing (Table 1). In

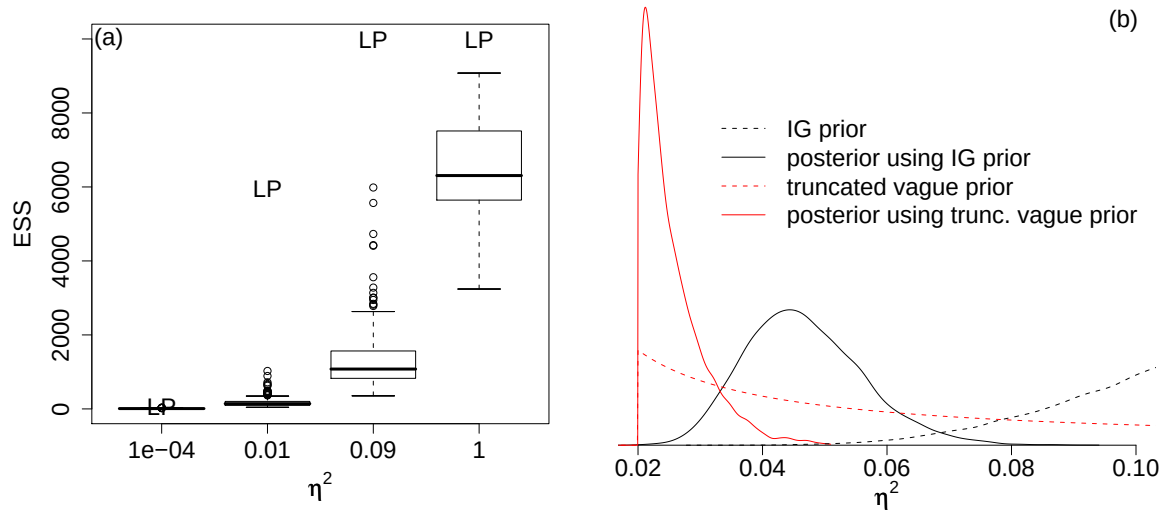

Figure 3: (a) ESS for a sample of 200 function values (boxplots) and for the log posterior density ('LP') as a function of fixed η^2 for Data2. (b) Posterior densities for η^2 under an $\mathcal{IG}(0.5, 2)$ prior and a lognormal prior truncated at 0.02 for Data3.

contrast, for the smallest value of η^2, coverage is low and predictive ability is poor because of poor mixing. A compromise value of η^2 (say 0.3^2 in this case) trades off well between mixing and statistical performance.

In general, to obtain reliable inference about overdispersion, one would want to initially allow sufficient freedom in one's prior for η^2 to allow small values of η^2. However, if the data appear to not be overdispersed and one wants to achieve reasonable mixing, one may want to run a version of the model with a fixed, larger value of η^2 and report inference for the other aspects of the model based on that MCMC. One can examine interval length as a function of η^2 in comparison with mixing properties to determine a good value of η^2. Cross-validation may be helpful for assessing coverage.

Turning to the model (12), there are two variance components, η^2 and σ_z^2. Interpretation of

η^2	coverage	interval length	test R^2
0.01^2	0.73	1.60	0.33
0.1^2	0.92	2.45	0.57
0.3^2	0.93	2.54	0.57
0.5^2	0.97	2.75	0.53
0.75^2	0.97	2.89	0.49
1^2	0.95	2.93	0.50
2^2	0.94	3.43	0.04

Table 1: For 200 function estimates, average 95% credible interval coverage and length, and test R^2 of posterior mean (indicating predictive ability), as a function of fixed η^2 for Data3.

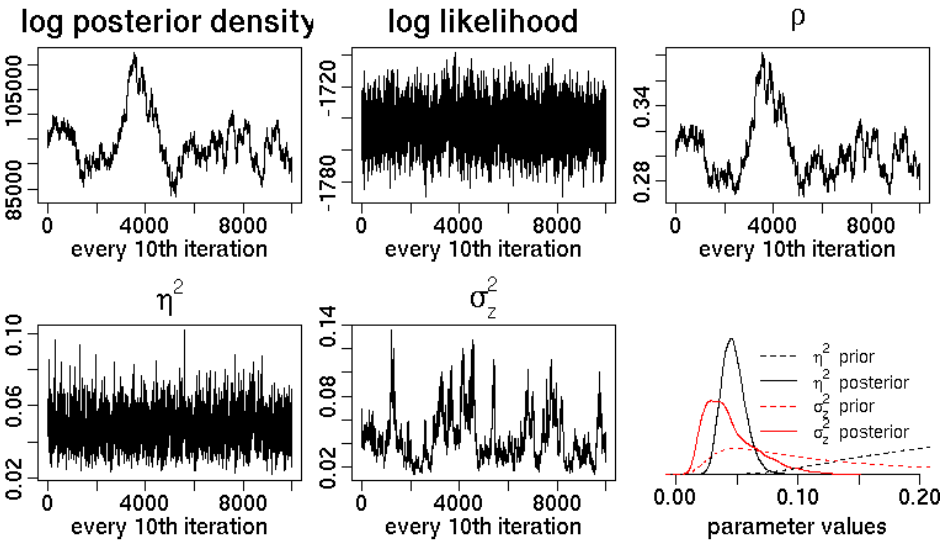

Figure 4: Trace plots for Wikle approach applied to Data2, with lower right subplot showing prior and posterior densities for η^2 and σ_z^2.

these parameters is difficult as the parameterization divides any inherent overdispersion into seemingly non-identifiable components. Royle and Wikle (2005) claim that both η^2 and σ_z^2 are identifiable with sufficient within-cell replication, with η^2 accounting for overdispersion within cell, while σ_z^2 represents uncorrelated variability across grid cells, inducing a lack of spatial smoothness beyond that induced by the discretization. Wikle (2002) and Royle and Wikle (2005) sample both components and find reasonable mixing, perhaps because their inverse gamma distributions prevent small values of the components and perhaps because their empirical count data have real overdispersion that provides information about a functional of the variance components ($\eta^2 + \gamma^2\sigma_z^2$), while replication provides information about η^2. In contrast, I have had difficulty achieving reasonable mixing for the two variance components in simulations with no overdispersion ($\sigma_z^2 \equiv \eta^2 \equiv 0$) (Figure 4), presumably because of the lack of identifiability and lack of overdispersion. Note how the posteriors for the variance components concentrate on the smallest values allowed by the priors, while at the same time the likelihood mixes well, indicating that the process values, $\boldsymbol{\lambda}$, are well-identified by the data and mix well.

5.2. Joint sampling of hyperparameters and process

The covariance hyperparameters in GP models frequently mix poorly. For illustration, consider ρ. The difficulty in sampling ρ is that a simple Metropolis-Hastings proposal for ρ results in a new set of variances for the coefficients, $\boldsymbol{u}$. Since these coefficients are not part of the proposal, proposing ρ^* can easily produce a low prior density, $\pi(\boldsymbol{u}|\rho^*, \cdot)$, because the new prior is inconsistent with the current $\boldsymbol{u}$. For example, with a process generated based on $\rho = 0.1$, the prior logdensity, $\log \pi(\boldsymbol{u}|\rho = 0.1, \cdot)$, is 50411 while the prior logdensity, $\pi(\boldsymbol{u}|\rho^* = 0.101, \cdot)$, is 50387, a change of 24 in the logdensity, despite the fact that surfaces generated based on $\rho = 0.1$ compared to $\rho = 0.101$ are indistinguishable even with massive amounts of data,

assuming non-negligible error variance. Note that the issue here is not a matter of whether we can sample from the full conditional for ρ; the primary obstacle in sampling ρ is the strong dependence of ρ and $\boldsymbol{u}$ (Rue, Steinsland, and Erland 2004). A better proposal would account for the strong dependence between ρ and $\boldsymbol{u}$ by proposing them jointly, allowing the lower levels of the model hierarchy close to the data to arbitrate between different values of ρ. In the context of Markov random field models, Rue and Held (2005, pp. 142-143) suggest a similar strategy of jointly sampling process values and their hyperparameters.

My strategy is to tie the covariance hyperparameters more closely to the coefficients and hence to the data by having the effects of proposing new hyperparameter values ripple down through the hierarchy of the model. I do this by jointly proposing a hyperparameter, generically denoted $\theta \in \{\rho, \sigma^2\}$, and, then conditional on that hyperparameter and within the same Metropolis-Hastings proposal, proposing the process coefficients from $[\boldsymbol{u}|\theta^*, \cdot]$. This provides a joint proposal for $(\theta, \boldsymbol{u})$ that adjusts $\boldsymbol{u}$ in such a way that it is more consistent with the proposed value, θ^*. Parameterizations that permit proposing from the full conditional, $[\boldsymbol{u}|\theta, \cdot]$, are likely to particularly benefit from this approach, with $\boldsymbol{u}$ effectively integrated out of the joint density. Provided the correct Hastings adjustment (ratio of proposal densities) is made, this joint proposal is a standard, valid Metropolis-Hastings sampling scheme, implemented as a marginal proposal for θ and a conditional proposal for $[\boldsymbol{u}|\theta, \cdot]$, with a single acceptance decision, as discussed in Rue and Held (2005, p. 142). I now detail these joint proposals in the three basic sampling schemes described in Section 4.

For the data augmentation sampling approach for normal data, one proposes θ^*, and then proposes from the full conditional, $[\boldsymbol{u}|\theta^*, \cdot]$ (9). Acceptance is then determined based on the ratio of the proposed and current posterior densities, $\pi(\boldsymbol{u}, \theta|\boldsymbol{y}, \cdot)$ divided by the Hastings ratio for $\boldsymbol{u}$ (and θ as well if not proposed symmetrically). Note that the proposal for $\boldsymbol{u}$ is conditional on the proposed θ^*, so a Metropolis-Hastings acceptance decision is needed because we are not doing a joint Gibbs sample for $(\boldsymbol{u}, \theta)$. The Hastings ratio is based on the proposal mean (9) and variance (10), with the variances for complex-valued coefficients scaled by one-half (22) and not including the coefficients that are complex conjugates of sampled coefficients. This is calculated in the **spectralGP** package using the `Hastings.coeff.gp()` method or by providing `returnHastings=TRUE` as an argument to `Gibbs.sample.coeff.gp()`. Template code is provided in Appendix B as Code E.

In the modified Wikle approach, the presence of the latent $\tilde{\boldsymbol{\lambda}}$ values (14) distances the coefficients from the data. However, one can mimic the proposal just described by sampling θ and then $\boldsymbol{u}$ from $[\boldsymbol{u}|\boldsymbol{\lambda}, \theta^*, \cdot]$, conditioning on $\boldsymbol{\lambda}$ rather than $\boldsymbol{y}$, and being satisfied with a proposal for $\boldsymbol{u}$ that is consistent with the new proposed θ^* and the current $\boldsymbol{\lambda}$, albeit without any direct influence of the data. Again a Hastings correction is needed, and can be calculated using the `Hastings.coeff.gp()` function, but with $\boldsymbol{\lambda}$ taking the place of $\boldsymbol{y}$. Template code is given as Code F. For the original Wikle approach, $\boldsymbol{z}$ takes the place of $\boldsymbol{\lambda}$ above.

However, in neither the modified nor original Wikle parameterizations does the sampling directly link θ to the observations, causing there to be no influence of the likelihood on the acceptance. An alternative that carries the changes through to the level of the data is to avoid sampling from the conditionals as described above and instead propose to move $\boldsymbol{u}$ and $\boldsymbol{\lambda}$ (and $\boldsymbol{z}$ in the Wikle parameterization) in such a way that their prior densities remain constant.

First propose θ^*. Then, deterministically propose,

$$u_i^* = u_i \cdot \frac{\sqrt{(\boldsymbol{\Sigma}_{\theta^*})_{i,i}}}{\sqrt{(\boldsymbol{\Sigma}_{\theta})_{i,i}}}, \ i = 1, \ldots, M. \tag{19}$$

Modifying u_i based on its prior variance, $(\boldsymbol{\Sigma}_\theta)_{i,i}$, allows the hyperparameters to mix more quickly by avoiding proposals for which the original coefficients are no longer probable based on their new prior variances. In the modified Wikle parameterization, one next proposes

$$\lambda_i^* = \lambda_i - \gamma(\boldsymbol{\Psi}\boldsymbol{u})_i + \gamma(\boldsymbol{\Psi}\boldsymbol{u}^*)_i, \tag{20}$$

while in the Wikle parameterization, one proposes $z_i^* = z_i - (\boldsymbol{\Psi}\boldsymbol{u})_i + (\boldsymbol{\Psi}\boldsymbol{u}^*)_i$ and finally $\lambda_i^* = \lambda_i - \gamma \boldsymbol{K}_i \boldsymbol{z} + \gamma \boldsymbol{K}_i \boldsymbol{z}^*$. This approach propagates the changes through the model in a way that ties θ directly to the likelihood. Such deterministic proposals are valid MCMC proposals so long as the Jacobian of the transformation is included in the acceptance ratio, based on a modification of the argument in Green (1995). The Jacobian of the transformation for $\boldsymbol{u}$ cancels with the ratio of the prior distributions for $\boldsymbol{u}$, $\pi(\boldsymbol{u}^*|\theta^*)/\pi(\boldsymbol{u}|\theta)$, to give the final Metropolis-Hastings acceptance for the entire joint proposal of $(\theta^*, \boldsymbol{u}^*, \boldsymbol{\lambda}^*)$ or $(\theta^*, \boldsymbol{u}^*, \boldsymbol{z}^*, \boldsymbol{\lambda}^*)$ based only on the ratio of the proposed and current prior densities for θ, the proposed and current likelihoods, and any required Hastings ratio to account for non-symmetric proposals for θ. Note that the transformations for $\boldsymbol{z}$ and $\boldsymbol{\lambda}$ have Jacobian of one. The validity of the deterministic proposal can be seen intuitively by considering Metropolis proposals in place of the transformation (19) with very small proposal variances, $\zeta^2 \approx 0$, e.g., $u_i^* \sim \mathcal{N}(u_i((\boldsymbol{\Sigma}_{\theta^*})_{i,i})^{1/2}((\boldsymbol{\Sigma}_\theta)_{i,i})^{-1/2}, \zeta^2)$, and calculating the acceptance ratio of such a proposal. Template code for the modified Wikle parameterization is given in Appendix B as Code G. Note that a similar joint proposal could be made for $\theta = \sigma_z^2$ to tie this hyperparameter more closely to the data.

For the coefficient block sampling scheme, no Gibbs scheme is available. Instead, one can carry out a joint sample in a similar manner to that just described, by proposing θ^* and then proposing $\boldsymbol{u}$ as in (19). Acceptance is determined by the ratio of the proposed and current prior densities for θ and proposed and current likelihoods, and any required Hastings ratio to account for non-symmetric proposals for θ. Template code is given in Appendix B as Code H.

In Table 2, I show a comparison of mixing for the modified Wikle parameterization (14) with 1.) straightforward sampling of γ and ρ, 2.) joint sampling of $(\sigma^2, \boldsymbol{u})$ and of $(\rho, \boldsymbol{u})$ based on Gibbs samples from $[\boldsymbol{u}|\sigma^2, \cdot]$ and $[\boldsymbol{u}|\rho, \cdot]$, and 3.) joint sampling via (19-20). Both joint sampling approaches appear to mix much more quickly than the simple Metropolis-Hastings proposals for the hyperparameters. The joint sampling with the full conditional sampling from $[\boldsymbol{u}|\theta, \cdot]$ does not mix as well as using (19-20), perhaps because the conditional Gibbs sample does not modify $\boldsymbol{\lambda}$ and therefore does not involve the likelihood in the determination of proposal acceptance. Note that for $\eta^2 \equiv 0.2^2$, the improved mixing of the deterministic shift proposal compared to the conditional Gibbs is even more marked (not shown).

5.3. Empirical comparison of sampling methods and recommendations

Based on the evidence provided in Section 5.2, it appears that joint sampling of θ and $\boldsymbol{u}$ in the modified Wikle parameterization greatly improves mixing, with deterministic sampling of $\boldsymbol{u}$

		Sampling Method		
Quantity	Dataset	simple: γ, ρ	joint, Gibbs: σ^2, ρ	joint, deterministic: σ^2, ρ
LP	Data1	NM	42	193
	Data2	NM	87	205
	Data3	NM	17	447
	Data5	NM	143	646
ρ	Data1	NM	37	146
	Data2	NM	70	145
	Data3	NM	15	414
	Data5	NM	125	289
f	Data1	549 (28-1898)	510 (34-1338)	597 (90-1808)
	Data2	1806 (22-3496)	1805 (132-3763)	2137 (229-4231)
	Data3	236 (5-2154)	611 (200-3062)	657 (307-3232)
	Data5	1563 (17-6503)	1964 (519-8958)	2021 (467-8971)

Table 2: ESS for log posterior density, ρ and median (range) of 200 sample function values by dataset for three sampling approaches: 1.) sampling γ and ρ using simple Metropolis-Hastings, 2.) jointly sampling each of σ^2 and ρ with $\boldsymbol{u}$ based on conditional Gibbs sample for $\boldsymbol{u}$, and 3.) jointly sampling each of σ^2 and ρ with $\boldsymbol{u}$ and $\boldsymbol{\lambda}$ as in (19-20). η^2 is fixed at 0.3^2. 'NM' indicates that the chain has not burned in or is mixing so slowly as to make calculation of ESS uninformative.

better than full conditional sampling for $\boldsymbol{u}$. In Appendix A.4 I also consider reparameterizing (σ^2, ρ) to reduce potential posterior correlation between these parameters, but find little improvement in mixing.

Here I compare mixing for the three parameterizations in Section 4: the modified Wikle approach with joint sampling of hyperparameters and coefficients, block sampling with joint sampling of hyperparameters and coefficients, and the original approach of Wikle without joint sampling. Since the latter is essentially the same as the modified Wikle approach with one extra layer, I do not devise a joint sampling scheme for it, but rather consider mixing under the sampling approach proposed by Wikle (2002) and Royle and Wikle (2005). In general, the modified Wikle approach outperforms block sampling and the original Wikle approach. Table 3 shows that for the simple function (Data1 and Data2), block sampling is worse than the modified Wikle approach but shows some degree of mixing, while for the more complicated function (Data3 and Data5), the block sampling approach does not appear to have burned in by 100,000 iterations. The original Wikle approach also has not burned in, as judged by the log posterior density and ρ although the sample function values appear to be mixing somewhat. Note that while the increase in sample size (from Data1 to Data2 and from Data3 to Data5) seems to result in somewhat improved mixing, the effect is not substantial.

A key question is how fine a resolution to use for the grid. While one does not want to oversmooth by virtue of using too coarse a resolution, finer resolution estimation takes longer to run and can exhibit slower mixing, because of the higher-dimensionality of the coefficients that are fit in the MCMC. My suggestion is to use a grid that is fine enough for reasonable prediction with the expected heterogeneity of the surface, but to make use of sensitivity

Quantity	Dataset	Sampling Method		
		modified Wikle	original Wikle	block sampling
LP	Data1	193	NM	54
	Data2	205	NM	25
	Data3	447	NM	53
	Data5	646	NM	NM
ρ	Data1	146	NM	107
	Data2	145	NM	40
	Data3	414	NM	NM
	Data5	289	22	NM
f	Data1	597 (90-1808)	125 (13-321)	551 (247-1166)
	Data2	2137 (229-4231)	56 (9-166)	473 (198-873)
	Data3	657 (307-3232)	298 (106-737)	33 (7-139)
	Data5	2021 (467-8971)	467 (127-1100)	25 (6-83)

Table 3: ESS for log posterior density, ρ and median (range) of 200 sample function values by dataset for the three parameterizations: 1.) modified Wikle with η^2 fixed at 0.3^2, 2.) original Wikle parameterization and sampling approach, and 3.) block sampling.

analyses to choose the grid resolution in light of mixing performance and computational speed. For the simple simulated data with an effective value of $\rho \approx 0.3$ (Data1 and Data2), a resolution of $k = 128$ is probably more than sufficient for good prediction (even coarser resolution might be sufficient), and runs with $k = 256$ and $k = 512$ showed slower mixing. For the simulated data with $\rho = 0.05$ (Data3, Data4, and Data5), $k = 128$ also seemed to be sufficient. Mixing with $k = 256$ was not substantially degraded relative to $k = 128$, but for $k = 512$, mixing was substantially worse.

These results suggest that mixing using the block sampling approach is substantially slower than the modified Wikle approach, particularly with a more variable underlying process. However, results may depend significantly on the form of the model and the exact data used. In Paciorek (2007), with a coarse grid, simple spatial functions, and binary observations, mixing was reasonable using the block sampling approach. In a multivariate setting within a complicated hierarchical model (Paciorek and McLachlan 2007), with a compound Dirichlet-multinomial likelihood for 10 categories and a coarse 32 by 32 grid, mixing was reasonable, albeit slow, with the block sampling approach, and the modified Wikle approach provided no improvement and was slower to compute.

6. Discussion

This paper introduces an R package for the Fourier basis representation of Gaussian processes, pioneered by Wikle (2002), and provides template code for fitting Bayesian models for exponential family data. The code can be readily adapted for more complicated hierarchical models. I discuss several possible parameterizations, including models allowing for overdispersion, and describe potential nonidentifiability in the hierarchical model of Wikle (2002) that may impact mixing. I document some of the critical issues affecting MCMC mixing in these models, in particular, the difficulty in mixing for ρ in particular and the dependence of

mixing speed on the dispersion parameter, η^2. In models with little noise (interpolating models) or non-Gaussian situations with little overdispersion, a small value of η^2 can substantially impede mixing. Based on a series of experiments with simulated Poisson data, I recommend use of a modified version of the parameterization of Wikle (2002), with an approach for joint sampling of the hyperparameters and the basis coefficients to more efficiently sample the hyperparameters by tying them more closely to the data. In contrast, while the block sampling approach of Paciorek (2007) works only somewhat less well for a relatively smooth spatial function, it mixes very poorly for a very unsmooth spatial function. However, the block sampling approach has the virtue of avoiding the overdispersion parameter that, if small, can hurt mixing and of simplicity, which may be helpful in more complicated hierarchical models. I could not achieve reasonable mixing of the parameterization and sampling approach suggested in Wikle (2002), presumably because of dataset-dependent differences in mixing, but also possibly because of the difficulty in replicating Bayesian MCMC schemes. Note that these recommendations and conclusions are based on qualitative, rather than exhaustive, testing.

The critical smoothing parameters (ρ and either σ or γ) appear to be the parameters that mix most slowly in the Fourier basis representation, as they are in many spatial models. In particular, ρ changes the amount of smoothing, by changing the prior weights on the basis functions, which vary in their frequency. Changing this parameter changes the form of the model, analogous to adding or subtracting basis functions in a free-knot spline model. Achieving reasonable mixing across model spaces is generally difficult.

Some alternative spatial models, such as thin plate splines and radial basis function models with fixed basis functions (Kammann and Wand 2003; Ruppert *et al.* 2003) have modeled spatial functions without estimating a spatial correlation parameter, relying solely on variance components (in the radial basis model) to achieve smoothing. O'Connell and Wolfinger (1997) relate the ratio of σ^2 and η^2 in a Gaussian setting to the smoothing parameter in a thin plate spline model, and Nychka (2000) speculates that this ratio may be more important than the spatial correlation parameter in smoothing noisy data. Zhang (2004) found that ρ and σ^2 cannot both be estimated consistently under infill asymptotics. I experimented with fixing ρ and forcing σ^2 to perform the smoothing role, but found that the model did not estimate the right amount of smoothing and predictive performance was poor. It may be that in this and perhaps other spatial models, models with estimated values of ρ are more efficient. This issue appears not to have been addressed thoroughly in the literature (but see Laslett (1994) and invited comments) and deserves more attention.

One might explore more sophisticated MCMC algorithms to improve mixing. For example, Christensen *et al.* (2006) develop a data-dependent reparameterization scheme for improved MCMC performance and apply the approach with Langevin updates that use gradient information; while promising, the approach is computationally intensive, again involving $n \times n$ matrix computations at each iteration, and software is not available. For the Fourier representation the high-dimensionality and complex values of the basis coefficients pose an impediment to such an approach. Based on the results here, I believe that proposals that jointly consider the key hyperparameters and the basis coefficients are critical in achieving adequate mixing.

One drawback to the GP model presented here is its restriction to stationary GPs. Future work on this model structure to allow for nonstationarity in the spatial process will consider wavelet bases in place of the Fourier basis used here, in particular the two-dimensional wavelet basis used by Matsuo, Paul, and Nychka (2006) to fit irregular Gaussian data in a non-Bayesian fashion. However, mixing may be more challenging in a more complicated model

with additional hyperparameters. An alternative relates to the work of Pintore and Holmes (2006), who have extended the Higdon/Paciorek/Stein (Higdon, Swall, and Kern 1999; Stein 2005; Paciorek and Schervish 2006) nonstationary covariance model based on kernel convolutions to the spectral domain. This allows one to build nonstationarity based on a latent process representing spatially-varying ρ or ν. Given the widespread interest in nonstationary and space-time representations, fast computation for such models is of obvious interest, but it is not clear how these covariance structures would be represented in the type of basis function approach developed here.

Acknowledgments

The author thanks Chris Wikle for introducing him to the Fourier basis representation and Andy Houseman for the idea of using R environments as a means of passing by reference. The project was supported by Grants numbered 5 T32 ES007142-23 (to the Department of Biostatistics at Harvard School of Public Health) and 5 P30 ES000002 (to Harvard School of Public Health) from the National Institute of Environmental Health Sciences (NIEHS), NIH. The contents are solely the responsibility of the author and do not necessarily represent the official views of NIEHS, NIH.

References

Banerjee S, Carlin B, Gelfand A (2004). *Hierarchical Modeling and Analysis for Spatial Data*. Chapman & Hall.

Berger JO, De Oliveira V, Sansó B (2001). "Objective Bayesian Analysis of Spatially Correlated Data." *Journal of the American Statistical Association*, **96**(456), 1361–1374.

Booth JG, Hobert JP (1999). "Maximizing Generalized Linear Mixed Model Likelihoods with an Automated Monte Carlo EM Algorithm." *Journal of the Royal Statistical Society B*, **61**, 265–285.

Borgman L, Taheri M, Hagan R (1984). "Three-dimensional, Frequency-domain Simulations of Geological Variables." In G Verly (ed.), "Geostatistics for Natural Resources Characterization, Part 1," pp. 517–541. D. Reidel Publishing Company.

Christensen O, Møller J, Waagepetersen R (2000). "Analysis of Spatial Data Using Generalized Linear Mixed Models and Langevin-type Markov Chain Monte Carlo." *Technical Report R-002009*, Department of Mathematics, Aalborg University. URL http://www.math.auc.dk/~rw/publications.html.

Christensen O, Roberts G, Sköld M (2006). "Robust Markov Chain Monte Carlo Methods for Spatial Generalized Linear Mixed Models." *Journal of Computational and Graphical Statistics*, **15**, 1–17.

Christensen OF, Waagepetersen R (2002). "Bayesian Prediction of Spatial Count Data Using Generalized Linear Mixed Models." *Biometrics*, **58**, 280–286.

Cowles MK, Carlin BP (1996). "Markov Chain Monte Carlo Convergence Diagnostics: A Comparative Review." *Journal of the American Statistical Association*, **91**, 883–904.

Cressie N (1993). *Statistics for Spatial Data*. Wiley-Interscience, New York, revised edition.

Diggle PJ, Tawn JA, Moyeed RA (1998). "Model-based Geostatistics." *Applied Statistics*, **47**, 299–326.

Dudgeon D, Mersereau R (1984). *Multidimensional Digital Signal Processing*. Prentice Hall, Englewood Cliffs, New Jersey.

Furrer R, Genton MG, Nychka D (2006). "Covariance Tapering for Interpolation of Large Spatial Datasets." *Journal of Computational and Graphical Statistics*, **15**, 502–523.

Gelfand A, Sahu S, Carlin B (1996). "Efficient Parametrizations for Generalized Linear Mixed Models." In J Bernardo, J Berger, A Dawid, A Smith (eds.), "Bayesian Statistics 5," pp. 165–180.

Gelman A (2006). "Prior Distributions for Variance Parameters in Hierarchical Models (Comment on Browne and Draper Article)." *Bayesian Analysis*, **1**(3), 515–534.

Gibbons RD, Hedeker D (1997). "Random Effects Probit and Logistic Regression Models for Three-level Data." *Biometrics*, **53**, 1527–1537.

Green P (1995). "Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination." *Biometrika*, **82**, 711–732.

Hastie TJ, Tibshirani RJ (1990). *Generalized Additive Models*. Chapman & Hall Ltd, London. ISBN 0412343908.

Heagerty PJ, Lele SR (1998). "A Composite Likelihood Approach to Binary Spatial Data." *Journal of the American Statistical Association*, **93**, 1099–1111.

Heagerty PJ, Lumley T (2000). "Window Subsampling of Estimating Functions with Application to Regression Models." *Journal of the American Statistical Association*, **95**(449), 197–211.

Hedeker D, Gibbons RD (1994). "A Random-Effects Ordinal Regression Model for Multilevel Analysis." *Biometrics*, **50**, 933–944.

Higdon D, Swall J, Kern J (1999). "Non-stationary Spatial Modeling." In J Bernardo, J Berger, A Dawid, A Smith (eds.), "Bayesian Statistics 6," pp. 761–768. Oxford University Press, Oxford, U.K.

Hwang JN, Lay SR, Maechler M, Martin D, Schimert J (1994). "Regression Modeling in Back-propagation and Projection Pursuit Learning." *IEEE Transactions on Neural Networks*, **5**, 342–353.

Kammann E, Wand M (2003). "Geoadditive Models." *Applied Statistics*, **52**, 1–18.

Laslett GM (1994). "Kriging and Splines: An Empirical Comparison of Their Predictive Performance in Some Applications (Disc: P 401-409)." *Journal of the American Statistical Association*, **89**, 391–400.

Matsuo T, Paul D, Nychka D (2006). "Nonstationary Covariance Modeling for Incomplete Data: Smoothed Monte Carlo EM Approach." Submitted.

McCulloch CE (1994). "Maximum Likelihood Variance Components Estimation for Binary Data." *Journal of the American Statistical Association*, **89**, 330–335.

McCulloch CE (1997). "Maximum Likelihood Algorithms for Generalized Linear Mixed Models." *Journal of the American Statistical Association*, **92**, 162–170.

McCulloch CE, Searle SR (2001). *Generalized, Linear, and Mixed Models*. John Wiley & Sons.

Neal R (1993). "Probabilistic Inference Using Markov Chain Monte Carlo Methods." *Technical Report CRG-TR-93-1*, Department of Computer Science, University of Toronto. URL http://www.cs.toronto.edu/~radford/papers-online.html.

Nychka DW (2000). "Spatial-Process Estimates as Smoothers." In M Schimek (ed.), "Smoothing and Regression: Approaches, Computation, and Application," pp. 393–424. John Wiley & Sons.

O'Connell M, Wolfinger R (1997). "Spatial Regression Models, Response Surfaces, and Process Optimization." *Journal of Computational and Graphical Statistics*, **6**, 224–241.

Oman S, Landsman V, Carmel Y, Kadmon R (2007). "Analyzing Spatially Distributed Binary Data Using Independent-Block Estimating Equations." *Biometrics*. In press.

Paciorek C (2007). "Computational Techniques for Spatial Logistic Regression with Large Datasets." *Computational Statistics and Data Analysis*, **51**, 3631–3653.

Paciorek C, McLachlan J (2007). "Long-term Vegetation Dynamics: Bayesian Inference for Spatio-temporal Trends in Forest Composition Using the Fossil Pollen Record." *Technical report*, Harvard University Biostatistics. Forthcoming.

Paciorek C, Schervish M (2006). "Spatial Modelling Using a New Class of Nonstationary Covariance Functions." *Environmetrics*, **17**, 483–506.

Pintore A, Holmes C (2006). "Non-stationary Covariance Functions via Spatially Adaptive Spectra." *Journal of the american Statistical Association*. In review.

R Development Core Team (2006). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Royle JA, Wikle CK (2005). "Efficient Statistical Mapping of Avian Count Data." *Environmental and Ecological Statistics*, **12**(2), 225–243.

Rue H, Held L (2005). *Gaussian Markov Random Fields: Theory and Applications*. Chapman & Hall, Boca Raton.

Rue H, Steinsland I, Erland S (2004). "Approximating Hidden Gaussian Markov Random Fields." *Journal of the Royal Statistical Society B: Statistical Methodology*, **66**(4), 877–892.

Rue H, Tjelmeland H (2002). "Fitting Gaussian Markov Random Fields to Gaussian fields." *Scandinavian Journal of Statistics*, **29**(1), 31–49.

Ruppert D, Wand M, Carroll R (2003). *Semiparametric Regression*. Cambridge University Press, Cambridge, U.K.

Shumway R, Stoffer D (2000). *Time Series Analysis and its Applications*. Springer-Verlag, New York.

Stein M (2005). "Nonstationary Spatial Covariance Functions." *Technical Report 21*, University of Chicago. URL http://www.stat.uchicago.edu/~cises/research/cises-tr21.pdf.

Stein ML, Chi Z, Welty LJ (2004). "Approximating Likelihoods for Large Spatial Data Sets." *Journal of the Royal Statistical Society B: Statistical Methodology*, **66**(2), 275–296.

Wikle C (2002). "Spatial Modeling of Count Data: A Case Study in Modelling Breeding Bird Survey Data on Large Spatial Domains." In A Lawson, D Denison (eds.), "Spatial Cluster Modelling," pp. 199–209. Chapman & Hall.

Wood S (2004). "Stable and Efficient Multiple Smoothing Parameter Estimation for Generalized Additive Models." *Journal of the American Statistical Association*, **99**, 673–686.

Wood S (2006). *Generalized Additive Models: An Introduction with R*. Chapman & Hall, Boca Raton.

Zhang H (2004). "Inconsistent Estimation and Asymptotically Equal Interpolation in Model-based Geostatistics." *Journal of the American Statistical Association*, **99**, 250–261.

A. Technical details

A.1. Constructing the gridded process

The details that follow draw on Dudgeon and Mersereau (1984), Borgman, Taheri, and Hagan (1984), and Wikle (2002). The first step in representing the function is to choose the grid size, M_d, in each dimension, $d = 1, \ldots, D$, to be a power of two. The M_d frequencies in the dth dimension are then $\omega^d \in \{0, 1, \ldots, \frac{M_d}{2}, -\frac{M_d}{2} + 1, \ldots, -1\}$, where the superscript represents the dimension. There is a complex exponential basis function for each distinct vector of frequencies, $\boldsymbol{\omega} = (\omega^1_{m_1}, \ldots, \omega^D_{m_D})$, $m_d \in \{0, \ldots, M_d - 1\}$, with corresponding complex-valued basis coefficient, $u_{m_1,\ldots,m_D}$.

First I show how to construct a random, mean zero, Gaussian process in one dimension from the M spectral coefficients, $u_m = a_m + b_m i$, $m = 0, \ldots, M-1$, and complex exponential basis functions, $\psi_m(s_j) = \exp(i\omega_m s_j)$, whose real and imaginary components have frequency ω_m. The circular domain of the process is $S^1 = (0, 2\pi)$ with the process evaluated only at the discrete grid points, $s_j \in \{0, 2\pi\frac{1}{M}, \ldots, 2\pi\frac{M-1}{M}\}$. To approximate real-valued processes, $u_0, \ldots, u_{M/2}$ are jointly independent, u_0 and $u_{M/2}$ are real-valued ($b_0 = b_{M/2} = 0$), and the remaining coefficients are determined, $u_{M/2+1} = \bar{u}_{M/2-1}, \ldots, u_{M-1} = \bar{u}_1$, where the overbar is the complex conjugate operation. This determinism causes the imaginary components of the basis functions to cancel, leaving a real-valued process,

$$\begin{aligned} g(s_j) &= \sum_{m=0}^{M-1} \psi_m(s_j) u_m = \sum_{m=0}^{\frac{M}{2}} \exp(i\omega_m s_j)(a_m + b_m i) + \sum_{m=\frac{M}{2}+1}^{M-1} \exp(i\omega_m s_j)(a_{M-m} - b_{M-m} i) \\ &= a_0 + 2\sum_{m=1}^{\frac{M}{2}-1} (a_m \cos(\omega_m s_j) - b_m \sin(\omega_m s_j)) + a_{M/2} \cos(\omega_{\frac{M}{2}} s_j). \end{aligned} \tag{21}$$

Hence for a grid of M values, the process is approximated as a linear combination of M spectral basis functions corresponding to M real-valued sinusoidal basis functions, including the constant function ($\omega_0 = 0$). To approximate mean zero Gaussian processes with a particular stationary covariance function, the coefficients have independent, mean zero Gaussian prior distributions with the spectral density for the covariance function, $\phi(\cdot; \boldsymbol{\theta})$, e.g., (5), determining the prior variances of the coefficients:

$$\begin{aligned} \mathrm{V}(u_m) &= \phi(\omega_m; \boldsymbol{\theta}) \\ &\Rightarrow \{\mathrm{V}(a_0) = \phi(\omega_0; \boldsymbol{\theta});\ \mathrm{V}(a_{M/2}) = \phi(\omega_{M/2}; \boldsymbol{\theta}); \\ &\quad \mathrm{V}(a_m) = \mathrm{V}(b_m) = \frac{1}{2}\phi(\omega_m; \boldsymbol{\theta}),\ \text{o.w.}\} \end{aligned} \tag{22}$$

.

The setup is similar in two dimensions, with a matrix of $M = M_1 M_2$ coefficients, $((u_{m_1,m_2}))$, $m_d \in \{0, \ldots, M_d - 1\}$, and corresponding frequency pairs, $(\omega^1_{m_1}, \omega^2_{m_2})$, and a toroidal domain. As seen in Table 4, many coefficients are again deterministically given by other coefficients to ensure that the process is a linear combination of real-valued sinusoidal basis functions of varying frequencies and orientations in $\Re^2$. The real and imaginary components of each coefficient, $u_{m_1,m_2} = a_{m_1,m_2} + b_{m_1,m_2} i$, are again independent. For $(m_1, m_2) \in$

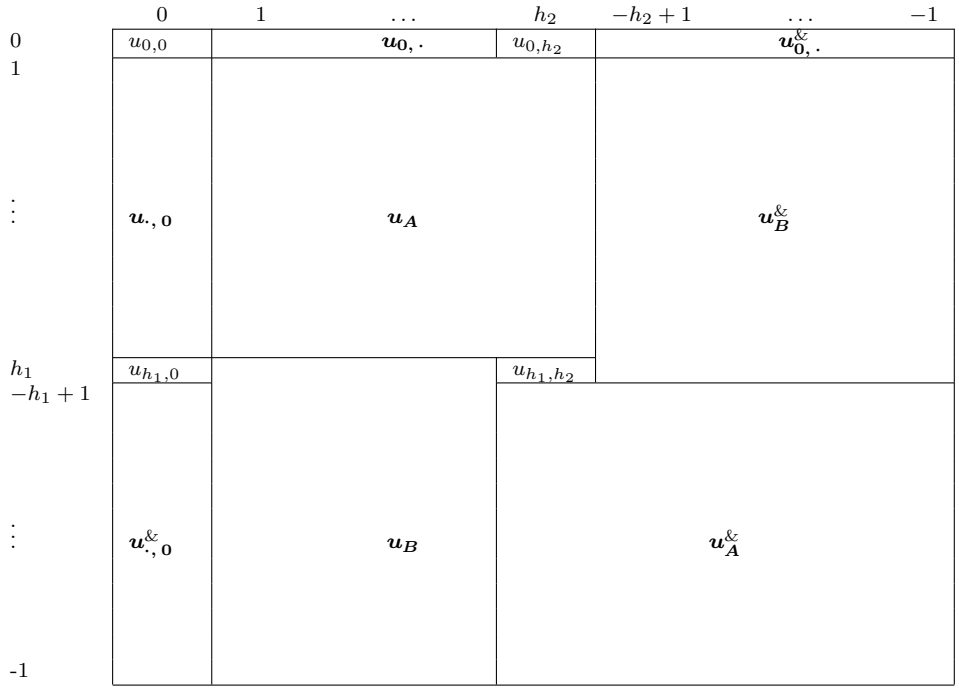

Table 4: Visual display of the spectral coefficients for a two-dimensional process. The frequencies in each dimension are indicated by the row and column labels, with $h_d = \frac{M_d}{2}$ for $d = 1, 2$. The & operation indicates that one takes the matrix or vector, flips it in both the horizontal and vertical directions (just the horizontal or vertical in the case of a vector) and then takes the complex conjugates of the elements.

$\{(0,0), (\frac{M_1}{2}, 0), (0, \frac{M_2}{2}), (\frac{M_1}{2}, \frac{M_2}{2})\}$, $b_{m_1,m_2} = 0$ and $\mathrm{V}(a_{m_1,m_2}) = \phi(\omega^1_{m_1}, \omega^2_{m_2}; \boldsymbol{\theta})$, while for the remaining complex-valued coefficients, $\mathrm{V}(a_{m_1,m_2}) = \mathrm{V}(b_{m_1,m_2}) = \frac{1}{2}\phi(\omega^1_{m_1}, \omega^2_{m_2}; \boldsymbol{\theta})$.

A.2. Periodicity and Euclidean domains

The construction produces periodic functions; in one dimension the process lives on a circular domain, while in two dimensions the process lives on a torus. To work in Euclidean space, we need to map Euclidean space onto the periodic domain. The goal is to use the representation for Euclidean domains without inducing anomalous correlations between locations that are far apart in Euclidean space, but close in the periodic domain. To do this, I suggest mapping the Euclidean domain of interest onto a portion of the periodic domain and ignoring the remainder of the periodic domain as follows.

In one dimension, $g(0) = g(2\pi)$, so the correlation function, $C(\tau) = C(g(0), g(\tau))$ of the process at distances $\tau \in (\pi, 2\pi)$ is the mirror image of the correlation function for $\tau \in (\pi, 0)$ with $\mathrm{Cor}(g(0), g(2\pi)) = 1$ (Figure 5). I avoid artifacts from this periodicity by mapping the interval $(0, 2\pi)$ to $(0, 2)$ and mapping the original domain of the observations to $(0, 1)$, thereby computing but not using the process values on $(1, 2)$. Note that the use of $\pi\rho$ rather than ρ in (5) allows us to interpret ρ on the $(0, 1)$ rather than $(0, \pi)$ scale. The modelled process on $(0, 1)$ is a piecewise constant function on an equally-spaced grid of size $M/2 + 1$. This setup ensures that the correlation structure of the approximating process is close to the correlation

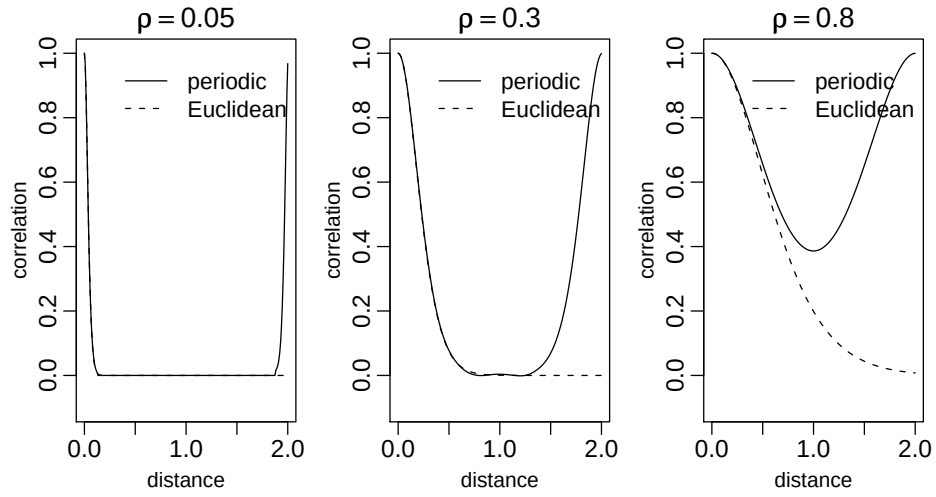

Figure 5: Comparison of correlation structure of GPs based on the standard Matérn covariance on the Euclidean domain, $(0, 2)$ (dashed lines), and approximate GPs based on the Fourier basis for the periodic domain, $(0, 2\pi)$, mapped to $(0, 2)$ (solid lines), for three values of ρ. Note that the Euclidean correlation for $\rho = 0.05$ falls off to zero as rapidly as the periodic case and remains at zero for all remaining distances.

structure of a GP with the desired stationary correlation function (Figure 5).

As the higher-dimension analogue of the one-dimensional case, I estimate the process on $(0, 1)^D$. To do so, I map the periodic domain $(0, 2\pi)^D$ to $(0, 2)^D$ and then map the observation domain onto the $(0, 1)^D$ portion (maintaining the scale ratio in the different dimensions, unless desired otherwise), thereby calculating but ignoring the process values outside this region. Note that if the original domain is far from square, I unnecessarily estimate the process in large areas of no interest, resulting in some loss of computational efficiency. Wikle (2002) and Royle and Wikle (2005) do not mention the issue of periodicity; it appears that they use a somewhat larger grid than necessary to include all observations (sometimes called padding) and rely on the correlation decaying sufficiently fast that anomalously high correlations between distant observations induced by the periodicity do not occur. For example, notice in Figure 5 that for $\rho = 0.05$ the correlation does not start to rise again until the distance is almost 2.0, so a small amount of padding would suffice).

A.3. Non-identifiability of $u_{0,0}$ and μ

The Fourier basis function corresponding to the coefficient, $u_{0,0}$, is a constant function. As such, it is not identifiable with respect to an overall mean parameter, μ, specified outside of the Fourier basis representation of the Gaussian process (12,14,16). One might choose to omit μ from the model, but this would generally be a mistake as the covariance structure (22) imposes a restrictive prior on $u_{0,0}$. A large value of γ or σ would allow for a process mean far from zero, but this would also allow the function to have high variability. An example of where the problem arises is a process with large mean, say 100, but whose variability places the function entirely in $(99, 101)$. Such a process would require a large value of $u_{0,0}$ but if γ or σ is

large enough to allow this, each would be so large as favor process estimates that vary widely around 100. Instead, a separate mean parameter is a better choice that will help to avoid slow mixing because of nonidentifiability. One can fix $u_{0,0} = 0$, without otherwise constraining the model. The **spectralGP** package can fix the coefficient and ignore it when calculating the prior density of the coefficients; this is done using `const.fixed=TRUE` as an argument to `gp()`. However, simulate GP realizations using the Fourier basis approximation, one should not fix this constant, in order to retain the desired approximate covariance structure.

A.4. Reparameterizing the covariance

In a GP model part of the difficulty in estimating the covariance parameters occurs because of limitations on identifiability. The data cannot readily distinguish the overall variability in the function, captured by γ or σ, from the decay in the spatial correlation, captured by ρ. In Bayesian models, these parameters tend to have high posterior correlation, while Zhang (2004) has shown that these two parameters cannot both be estimated consistently under infill asymptotics, but that a functional of the two can be estimated consistently. Note that in thin plate spline models and in the mixed model representation suggested by Kammann and Wand (2003) and Ruppert *et al.* (2003), there is only one parameter in place of the two covariance parameters here. However, as discussed further in Section 6, comparisons of estimates using the Fourier basis approach here suggest that ρ cannot be fixed in advance without seriously affecting the function estimates because the function heterogeneity is not adequately represented.

Given the results of Zhang (2004), in which the ratio, $\sigma^2/\rho^{2\nu}$, can be estimated consistently, consider reparameterizing on the log scale as $\psi_1 = \log\sigma + \log\rho$ and $\psi_2 = \log\sigma - \log\rho$. This approach uses the centered parameterization, fixing $\gamma \equiv 1$. The reparameterization will tend to reduce posterior correlation and allow each parameter to move more freely. Joint sampling as described in Section 5.2 can also be employed with this reparameterization. Template code for sampling based on the reparameterization and joint sampling of the parameters and process values using deterministic conditional proposals for $\boldsymbol{u}$ (19-20) is given in Appendix B as Code I under the modified Wikle parameterization and code J under the block sampling approach.

Since the joint sampling of each of σ^2 and ρ with $\boldsymbol{u}$ based on deterministic proposals for $\boldsymbol{u}$ and $\boldsymbol{\lambda}$ appeared to be the best of the options in Section 5.2, in Table 5, I compare mixing for that approach with the (σ^2, ρ) parameterization and the same joint approach using deterministic proposals with the (ψ_1, ψ_2) parameterization. There is little difference in mixing between the two parameterizations. Table 6 shows posterior correlations of (σ, ρ) and of (ψ_1, ψ_2) based on sampling under the original and the new parameterizations. For Data1, Data2, and Data3, ψ_1 and ψ_2 have little posterior correlation, suggesting that in principle, sampling using the new parameterization would mix more quickly, although this is not the case in Table 5. The minimal difference in mixing was also seen when using the joint sampling with the full conditional samples from $[\boldsymbol{u}|\theta, \cdot]$ and when fixing $\eta^2 = 0.2^2$ and $\eta^2 = 0.5^2$, as well as for an alternative reparameterization, $\psi_1 = \log\sigma$ and $\psi_2 = \log\sigma - \log\rho$. In practice, the minimal difference in mixing suggests that the posterior correlation between σ^2 and ρ is not materially hurting mixing, in sharp contrast to the importance of jointly sampling each covariance hyperparameter with $\boldsymbol{u}$.

Table 5: ESS for log posterior density, ρ, and median (range) of 200 sample function values by dataset when sampling is done using the parameterizations: 1.) (ρ, σ^2) and 2.) (ψ_1, ψ_2). η^2 is fixed at 0.3^2.

		Parameterization	
Quantity	Dataset	original: σ^2, ρ	Zhang: ψ_1, ψ_2
LP	Data1	193	105
	Data2	205	244
	Data3	447	443
	Data5	646	777
ρ	Data1	146	56
	Data2	145	157
	Data3	414	431
	Data5	289	426
f	Data1	597 (90-1808)	591 (89-1427)
	Data2	2137 (229-4231)	2154 (235-4274)
	Data3	657 (307-3232)	656 (305-3169)
	Data5	2021 (467-8971)	1972 (469-8879)

Table 6: Posterior correlations for $(\log\sigma, \log\rho)$ and (ψ_1, ψ_2) when sampling is done using the parameterizations: 1.) (σ^2, ρ) and 2.) (ψ_1, ψ_2). η^2 is fixed at 0.3^2.

		Parameterization	
Dataset	Posterior correlation	original: σ^2, ρ	Zhang: ψ_1, ψ_2
Data1	$\text{Cor}(\log\sigma, \log\rho)$	0.70	0.79
	$\text{Cor}(\psi_1, \psi_2)$	0.15	0.24
Data2	$\text{Cor}(\log\sigma, \log\rho)$	0.81	0.84
	$\text{Cor}(\psi_1, \psi_2)$	0.08	0.13
Data4	$\text{Cor}(\log\sigma, \log\rho)$	0.20	0.21
	$\text{Cor}(\psi_1, \psi_2)$	0.26	0.26
Data5	$\text{Cor}(\log\sigma, \log\rho)$	0.56	0.56
	$\text{Cor}(\psi_1, \psi_2)$	0.11	0.12

A.5. Starting values

Good starting values for the coefficients can be difficult to determine because of the high dimensionality of the coefficients and lack of a maximum likelihood-based estimate due to the need for shrinkage in estimating the coefficients. In addition, as described in Section A.2, a portion of the domain contains no observations. For the grid points not used to represent the domain of interest $\left(((0,1)^2)^C \cap (0,2)^2\right)$, it is helpful to initiate values for these buffering grid points so as to keep the variability and spatial range features of the data similar across the whole domain. This can be achieved by 'mirroring' the initial values from the portion of the domain in which the observations lie, as follows, in one dimension,

$$\hat{g}(s_M), \ldots, \hat{g}(s_{M/2+2}) \equiv \hat{g}(s_{M/2}), \ldots, \hat{g}(s_2). \tag{23}$$

In two dimensions, the mirroring occurs first across the the line $s_1 = 1$ (for $s_2 < 1$) and then across the line $s_2 = 1$, such that $\hat{g}(s_{m_1,m_2})$ is defined, for $m_1 > M_1/2+1$ and $m_2 <= M_2/2+1$ as $\hat{g}(s_{m_1,m_2}) \equiv \hat{g}(s_{M_1-m_1+2,m_2})$. For $m_2 > M_2/2+1$, take $\hat{g}(s_{m_1,m_2}) \equiv \hat{g}(s_{m_1,M_2-m_2+2})$.

In the data augmentation scheme for normal data, I suggest using a `gam()` fit to estimate the process values, predicting $\tilde{\boldsymbol{Y}}$ values at unobserved locations using the fitted model, mirroring the values, and then doing a Gibbs sample for the coefficients. In the Wikle approach, one can estimate the spatial process at the grid points based on a `gam()` fit, assign these values to $\boldsymbol{z}$ ($\boldsymbol{\lambda}$ in the modified Wikle approach) and initialize $\boldsymbol{u}$ via a Gibbs sample. For the block sampling scheme, one might use `gam()` to estimate the process on the grid, $\hat{\boldsymbol{g}}_{\boldsymbol{s}\#}$, add error and mirror the values, and then set $\boldsymbol{u} = (\frac{\gamma^2}{\eta^2}\boldsymbol{I} + \boldsymbol{\Sigma}^{-1})^{-1}\frac{\gamma}{\eta^2}\boldsymbol{\Psi}^\top(\hat{\boldsymbol{g}}_{\boldsymbol{s}\#} - \mu\boldsymbol{1})$, mimicing (15).

Some basic experiments with simulated datasets Data1, Data2, Data3, and Data5 suggest little difference between starting the coefficients based on a Gibbs sample and starting at values simulated from the prior conditional on the hyperparameter starting values. Reasonably rapid burn-in occurred when the coefficients were simulated from their prior, although mixing for Data1 was slightly better for the Gibbs sample starting values. For the coefficients corresponding to low frequencies, the long-run estimates are comparable for the different starting values. However, it may be the case that the Gibbs sample starting values are useful in some circumstances.

B. Template code

I provide R template code for MCMC sampling under various parameterizations and sampling schemes described in Sections 4 and 5. The code is designed for the simple model (1) in which the data come from an exponential family distribution: normal, Poisson, or binomial. The code assumes a constant mean, μ, in place of $\boldsymbol{x}_i^\top\boldsymbol{\beta}$, and Matérn covariance function with ν fixed. While this is a simple setting, users could take the template code and use it within a more complicated hiearchical model or easily extend to a regression structure in the mean. The contribution of this work is to provide code for easy manipulation and MCMC sampling of the Fourier basis representation of the Gaussian process component in the model.

The code makes use of the **spectralGP** package and uses easily modifiable R functions for the log-likelihood, prior distributions, and Gibbs sampling; the names of these will be obvious in the code. Also note that parameters take the form of R lists, with components that will be obvious from the code. The code does not save iterations, report acceptance rates, or adapt

the proposal variances based on acceptance rates, but these features could be readily added. The code assumes the default Matérn spectral density function but the only changes needed to use a different covariance function are to define a new spectral density function, use that function as the argument to the constructor `gp()`, and define an appropriate prior density for the parameter(s) of the covariance function. If the covariance/spectral density function has a second parameter, the code would need to be modified to allow for MCMC sampling of that parameter; the code currently takes the second parameter, ν, of the Matérn to be fixed.

Next I provide a brief overview of the model structure and sampling approach used in each template code file. The information is summarized in Table 7.

Code A (Section 4.1) In this template, data, $\boldsymbol{Y}$, are assumed to be normally distributed and are augmented with pseudo-observations, $\tilde{\boldsymbol{Y}}$,

$$\begin{aligned} \boldsymbol{Y} &\sim \mathcal{N}_n(\mu\mathbf{1} + \gamma\boldsymbol{K}\boldsymbol{\Psi}\boldsymbol{u}, \eta^2\boldsymbol{I}) \\ \tilde{\boldsymbol{Y}} &\sim \mathcal{N}_{M-n}(\mu\mathbf{1} + \gamma\tilde{\boldsymbol{K}}\boldsymbol{\Psi}\boldsymbol{u}, \eta^2\boldsymbol{I}), \end{aligned}$$

which allows Gibbs sampling of the process coefficients, $\boldsymbol{u}$. The pseudo-observations are also sampled via a Gibbs step. Covariance hyperparameters are sampled individually by Metropolis-Hastings. To speed mixing in some cases, joint sampling of η^2 and $\tilde{\boldsymbol{Y}}$ is also possible.

Code B (Section 4.2) In this template, data, $\boldsymbol{Y}$, are taken to be Poisson or binomial, and two latent processes are embedded in the model,

$$\begin{aligned} Y_i &\sim \mathcal{F}(h^{-1}(\lambda_i)) \\ \lambda_i &\sim \mathcal{N}(\mu + \gamma\boldsymbol{K}_{\boldsymbol{i}}\boldsymbol{z}, \eta^2) \\ \boldsymbol{z} &\sim \mathcal{N}_M(\boldsymbol{\Psi}\boldsymbol{u}, \sigma_z^2\boldsymbol{I}). \end{aligned}$$

Here $\boldsymbol{\lambda}$ is sampled via Metropolis-Hastings, and $\boldsymbol{z}$ and $\boldsymbol{u}$ are sampled via Gibbs steps. Covariance hyperparameters are sampled individually by Metropolis-Hastings. There are two overdispersion parameters, η^2 and σ_z^2, in this model.

Code C (Section 4.2) In this template, data, $\boldsymbol{Y}$, are taken to be Poisson or binomial, and one latent processes is embedded in the model,

$$\begin{aligned} Y_i &\sim \mathcal{F}(h^{-1}(\boldsymbol{K}_{\boldsymbol{i}}\boldsymbol{\lambda})) \\ \boldsymbol{\lambda} &\sim \mathcal{N}_M(\mu\mathbf{1} + \gamma\boldsymbol{\Psi}\boldsymbol{u}, \eta^2\boldsymbol{I}). \end{aligned}$$

I divide the latent process values into a group whose grid cells contain data, $\boldsymbol{\lambda}_{obs}$, which are sampled via Metropolis-Hastings, and the remaining values, $\tilde{\boldsymbol{\lambda}}$, which are sampled via a Gibbs step. To speed mixing in some cases, joint sampling of η^2 and $\tilde{\boldsymbol{\lambda}}$ is also possible. Covariance hyperparameters are sampled individually by Metropolis-Hastings. There is one overdispersion parameter, η^2, in this model.

Code D (Section 4.3) In this template, data, $\boldsymbol{Y}$, are taken to be Poisson or binomial, and the likelihood depends directly on the process coefficients,

$$Y_i \sim \mathcal{F}(h^{-1}(\mu + \gamma \boldsymbol{K_i \Psi u})).$$

The coefficients, $\boldsymbol{u}$, are sampled by Metropolis-Hastings in blocks. Covariance hyperparameters are sampled individually by Metropolis-Hastings. The model contains no overdispersion parameter.

Code E (Section 5.2) As in Code A, data, $\boldsymbol{Y}$, are assumed to be normally distributed and are augmented with pseudo-observations, $\tilde{\boldsymbol{Y}}$,

$$\begin{aligned} \boldsymbol{Y} &\sim \mathcal{N}_n(\mu \mathbf{1} + \boldsymbol{K \Psi u}, \eta^2 \boldsymbol{I}) \\ \tilde{\boldsymbol{Y}} &\sim \mathcal{N}_{M-n}(\mu \mathbf{1} + \tilde{\boldsymbol{K}} \boldsymbol{\Psi u}, \eta^2 \boldsymbol{I}), \end{aligned}$$

with the modification that $\gamma \equiv 1$ and σ^2 in (5) is allowed to vary. The process coefficients, $\boldsymbol{u}$, and pseudo-observations are both sampled via Gibbs steps. In contrast to Code A in which covariance hyperparameters are sampled by Metropolis-Hastings on their own, in this code, each covariance hyperparameter, σ^2 and ρ, is sampled jointly with $\boldsymbol{u}$ in a Metropolis-Hastings step.

Code F (Section 5.2) As in Code C, data, $\boldsymbol{Y}$, are taken to be Poisson or binomial, and one latent process is embedded in the model,

$$\begin{aligned} Y_i &\sim \mathcal{F}(h^{-1}(\boldsymbol{K_i \lambda})) \\ \boldsymbol{\lambda} &\sim \mathcal{N}_M(\mu \mathbf{1} + \gamma \boldsymbol{\Psi u}, \eta^2 \boldsymbol{I}). \end{aligned}$$

As in Code C, I divide the latent process values into a group whose grid cells contain data, $\boldsymbol{\lambda}_{obs}$, which are sampled via Metropolis-Hastings, and the remaining values, $\tilde{\boldsymbol{\lambda}}$, which are sampled via a Gibbs step. In contrast to Code C in which covariance hyperparameters are sampled by Metropolis-Hastings on their own, in this code, each covariance hyperparameter, σ^2 and ρ, is sampled jointly with $\boldsymbol{u}$ in a Metropolis-Hastings step. There is one overdispersion parameter, η^2, in this model.

Code G (Section 5.2) As in Code F, data, $\boldsymbol{Y}$, are taken to be Poisson or binomial, and one latent process is embedded in the model,

$$\begin{aligned} Y_i &\sim \mathcal{F}(h^{-1}(\boldsymbol{K_i \lambda})) \\ \boldsymbol{\lambda} &\sim \mathcal{N}_M(\mu \mathbf{1} + \gamma \boldsymbol{\Psi u}, \eta^2 \boldsymbol{I}). \end{aligned}$$

The one difference from Code F is that here, each covariance hyperparameter, σ^2 and ρ, is sampled jointly with both $\boldsymbol{u}$ and $\boldsymbol{\lambda}$ in a Metropolis-Hastings step.

Code H (Section 5.2) As in Code D, data, $\boldsymbol{Y}$, are taken to be Poisson or binomial, and the likelihood depends directly on the process coefficients,

$$Y_i \sim \mathcal{F}(h^{-1}(\mu + \gamma \boldsymbol{K_i \Psi u})).$$

The coefficients, $\boldsymbol{u}$, are sampled by Metropolis-Hastings in blocks. I divide the latent process values into a group whose grid cells contain data, $\boldsymbol{\lambda}_{obs}$, which are sampled via Metropolis-Hastings, and the remaining values, $\tilde{\boldsymbol{\lambda}}$, which are sampled via a Gibbs step. In contrast to Code D in which covariance hyperparameters are sampled by Metropolis-Hastings on their own, in this code, each covariance hyperparameter, σ^2 and ρ, is sampled jointly with $\boldsymbol{u}$ in a Metropolis-Hastings step. The model contains no overdispersion parameter.

Code I (Appendix A.4) As in Code G, data, $\boldsymbol{Y}$, are taken to be Poisson or binomial, and one latent process is embedded in the model,

$$\begin{aligned} Y_i &\sim \mathcal{F}(h^{-1}(\boldsymbol{K_i \lambda})) \\ \boldsymbol{\lambda} &\sim \mathcal{N}_M(\mu \mathbf{1} + \gamma \boldsymbol{\Psi u}, \eta^2 \boldsymbol{I}). \end{aligned}$$

The one difference from Code G is that here, the covariance parameters are reparameterized as $\psi_1 = \log \sigma + \log \rho$ and $\psi_2 = \log \sigma - \log \rho$. Each of ψ_1 and ψ_2 is sampled jointly with $\boldsymbol{u}$ and $\boldsymbol{\lambda}$ in a Metropolis-Hastings step.

Code J (Appendix A.4) As in Code H, data, $\boldsymbol{Y}$, are taken to be Poisson or binomial, and the likelihood depends directly on the process coefficients,

$$Y_i \sim \mathcal{F}(h^{-1}(\mu + \gamma \boldsymbol{K_i \Psi u})).$$

The one difference from Code H is that here, the covariance parameters are reparameterized as $\psi_1 = \log \sigma + \log \rho$ and $\psi_2 = \log \sigma - \log \rho$. Each of ψ_1 and ψ_2 is sampled jointly with $\boldsymbol{u}$ in a Metropolis-Hastings step.

Data distribution	Latent structure	Sampling of cov. parameters	Template code
Normal	data augmentation	individually: $\theta \in \{\sigma^2, \rho\}$	A
		joint: $\{\theta, \boldsymbol{u}\}$, $\theta \in \{\sigma^2, \rho\}$	E
Poisson/binomial	two processes	individually: $\theta \in \{\sigma^2, \rho\}$	B
Poisson/binomial	one process	individually: $\theta \in \{\sigma^2, \rho\}$	C
		joint: $\{\theta, \boldsymbol{u}\}$, $\theta \in \{\sigma^2, \rho\}$	F
		joint: $\{\theta, \boldsymbol{u}, \boldsymbol{\lambda}\}$, $\theta \in \{\sigma^2, \rho\}$	G
		joint: $\{\theta, \boldsymbol{u}, \boldsymbol{\lambda}\}$, $\theta \in \{\psi_1, \psi_2\}$	I
Poisson/binomial	no process	individually: $\theta \in \{\sigma^2, \rho\}$	D
		joint: $\{\theta, \boldsymbol{u}\}$, $\theta \in \{\sigma^2, \rho\}$	H
		joint: $\{\theta, \boldsymbol{u}\}$, $\theta \in \{\psi_1, \psi_2\}$	J

Table 7: Overview of model structure and covariance parameter sampling algorithms for template code.

Affiliation:

Christopher J. Paciorek
Department of Biostatistics
Harvard School of Public Health
655 Huntington Avenue
Boston, MA 02115, United States of America
E-mail: paciorek@alumni.cmu.edu
URL: http://www.biostat.harvard.edu/~paciorek/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 19, Issue 2 *Submitted:* 2006-08-09
April 2007 *Accepted:* 2007-04-10