Take Me Out to (Analyze) the Ballgame

Visualization and Analysis Techniques for Big Spatial Data

Chris Comiskey

May 4, 2017

Contents

1 Spatial Generalized Linear Mixed Models

1.1 Introduction

No matter the performace of the previously fit model, we expect there to be unexplained spatial variation in the mean. The covariates are limited in scope and depth, and Tobler's First Law of Geography tells us that things close together in space tend to behave more similarly than things further apart [Tobler, 1970]. Accordingly, we enhance the model to capture the unexplained spatial variation in the mean, and compensate for unobserved covariates, by adding a spatially correlated random effect [Banerjee et al., 2008].

1.1.1 Gaussian Random Field

A Gaussian random field is a popular and practical distribution for spatial random effects [Gelfand et al., 2010]. Let random variable vector $\boldsymbol{w}(\boldsymbol{s})$, for vector of locations $\boldsymbol{s} \in \boldsymbol{D} \subseteq \boldsymbol{R}^2$, be distributed multivariate Normal with mean $\mathbf{0}$; with symmetric, positive definite covariance matrix $\Sigma(\boldsymbol{\theta})$, and covariance parameter vector $\boldsymbol{\theta}$ [Haran, 2011].

$$\boldsymbol{w}|\boldsymbol{\theta} \sim MVN(\mathbf{0}, \Sigma(\boldsymbol{\theta})) \tag{1}$$

Including a random effect defined in this way, with a valid covariance matrix, would retool (2) as a *spatial* generlatized linear *mixed* model (SGLMM). Next we define the covariance structure we use in this study.

1.1.2 Exponential Covariance

To add a spatial random effect, distributed as a Gaussian random field, to the linear predictor in (2), it remains to define a spatial correlation structure to $\boldsymbol{w}$. Let w_{ij} be defined as in **5.1**, with an exponential covariance structure. That is, the i,jth element of $\Sigma(\phi, \sigma^2)$ is:

$$\Sigma(\phi, \sigma^2)_{i,j} = \sigma^2 exp(-||\boldsymbol{s}_i - \boldsymbol{s}_j||/\phi), \tag{2}$$

where $||s_i - s_j||$ is the Euclidean distance between $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, σ^2 is the scale parameter, and ϕ is the range parameter.

1.1.3 Spatial Generalized Linear Mixed Model

Inserting $\boldsymbol{w}(\boldsymbol{s})$ to the linear predictor in (1) gives the following spatial generalized linear mixed model (SGLMM):

$$\text{logit}(p_{ij}|\boldsymbol{s}_{ij}) = \boldsymbol{X}_{ij}(\boldsymbol{s}_{ij})\boldsymbol{\beta}_j + w(\boldsymbol{s}_{ij}). \tag{3}$$

This spatial hierarchical model, with its latent Gaussian random field, gives p_{ij} a complicated correlation structure. Bayesian statistical methodologies, primarily Markov chain Monte Carlo (MCMC) methods, are very popular for fitting spatial models of this kind [Banerjee et al., 2014]. In fact, MCMC is one of the few practical approaches available to fit a 'big n' model with complex spatial correlation. This is because of the 'big n problem' [Lindgren et al., 2011]. Namely, the computational costs for SGLMMs increase at a rate of $\mathcal{O}(n^3)$ (REFERENCE). This rate of increase leads to prohibitively slow model fitting. To attempt to fit SGLMMs in practically useful time spans, we try:

To estimate model parameters for (4) we tried:

1. Computational optimization, C++, an efficient algorithm, with Hamiltonian Monte Carlo in Stan

2. Dimension reduction with Predictive process models in `spBayes`

3. INLA SPDEs and `INLA-R`

Note that the ultimate goal of this research is practical, real-time applications for baseball fans, broadcasts, players, scouts, and teams. Therefore, model fitting speed matters on a finer time scale than academic research demands.

1.1.4 Markov Chains

Hierarchical Modeling and Analysis for Spatial Data [Banerjee et al., 2014]

- "Without doubt, the most popular computing tools in Bayesian practice today are Markov chain Monte Carlo (MCMC) methods."

- inference from posteriors of "...arbitrarily large dimension, essentially by reducing the problem to one of recursively solving a series of lower-dimensional (often unidimensional) problems."

- "...work by producing not a closed form for posterior, but a sample of values $\{\theta^{(g)}, g = 1, \ldots, G\}$ from this distribution." (G = number of draws from posterior)

- Two issues: MCMC algorithms produce *correlated* draws from poster (hence thinning, acf(), pacf(), and *convergence* diagnosis
- Two most popular MCMC algorithms: (1) Gibbs sampler (2) Metropolis Hastings algorithm

1.1.5 "Big N Problem"

1.2 Numerical Optimization; Hamiltonian Monte Carlo in Stan

1.2.1 Hamiltonian Dynamics and MCMC

[1] Trying to understand molecular states, Metropolis et al. [1953] created MCMC for "fast machines." Later, modeling molecular motion as a deterministic process, Alder and Wainwright [1959] introduced *Hamiltonian dynamics* as an alternate representation of Newtonian mechanics. Almost 30 years later, Duane et al. [1987] combined the two to create "hybrid Monte Carlo" to simulate certain quantum mechanical processes. Over time, this named morphed into *Hamilton* Monte Carlo (HMC), as it is known today. Eventually, Neal [1996] used HMC methods for explicitly statistical applications, studying neural networks.

HMC works by reframing the variables and distribution of interest as part of a physical system. From a physics standpoint, an object in a well defined three dimensional physical space can be completely characterized by its position and momemtum. For HMC, the variables of interest function as position variables, and auxiliary Gaussian variables are introduced serve as momentum variables. Simple updates for the auxiliary momentum variables generate, via a system of differential equations, proposals for Metropolis updates to the more important position variables (which represent the variabls of interest). The differential equation solutions estimate trajectories of the hypothetical physical object, which will then occupy a new position after some chosen time step. This crafty formulation enables distant, yet high probability, proposals for the variables of interest. Note that this contrasts favorably, in terms of mixing, to the random walk proposal generation process commonly used for Metropolis updates.

1.2.2 Hamilton Equations for MCMC

Let $q(t)$ be a d-dimensional (d parameters of interest) position vector that is a function of time t; and $U(q(t))$ represent the potential energy at time t. Let $p(t)$ give the d-dimensional momentum at time t, and $K(p(t))$

[1] The history and physics presented in this section owe heavily to, and are primarily informed by, [Neal et al., 2011]

represent kinetic energy at time t. Then the Hamilton equation,

$$H(q(t), p(t)) = U(q(t)) + K(p(t)), \tag{4}$$

measures the total energy of a system.

For HMC applications, we let the potential energy, $U(q)$, be minus the log of the probability density function of interest, plus any convenient constant[2]. Typically, HMC procedures define p as a d-dimensional zero mean Gaussian with covariance matrix M, and $K(p)$ as minus the log of the multivariate Gaussian probability density function. This gives:

$$H(q, p) = -\log f_q(q) + p^T \boldsymbol{M}^{-1} p/2. \tag{5}$$

This clever formulation provides useful partial derivatives for calculating the change in position and momentum over time. For $i = 1, \ldots, d$:

$$\frac{dq_i(t)}{dt} = \frac{\partial H}{\partial p_i}, \tag{6}$$

$$\frac{dp_i(t)}{dt} = -\frac{\partial H}{\partial q_i}. \tag{7}$$

Substituting in Hamilton's equation (5) and simplifying gives

$$\frac{dq_i(t)}{dt} = [\boldsymbol{M}^{-1} p]_i \tag{8}$$

$$\frac{dp_i(t)}{dt} = \frac{\partial \left[\log f_q(q)\right]}{\partial q_i} \tag{9}$$

The solutions to these two differential equations, that is $q(t)$ and p(t) such that (8) and (9) hold, give the instantaneous rate of change of position and momentum at time t.

subsubsectionMCMC Using Hamiltonian Dynamics [Neal et al., 2011] Steps.

- **Leapfrog method** = for calculating new position (q) and momentum (p) through tiny time steps
  - for **discretizing Hamilton equations**
  - akin to Taylor Series appoximations

[2] We omit t for clarity of presentation, here and elsewhere, but position and momentum remain functions of time t.

  - Postion (q) (or momentum (p)) at t_0 plus time step times rate of change of position (q) (momentum (p)) variable at t_0
  - Leapfrom Method does half step for momentum (p), full step for postion (q), other half step for momentum (p). Damn good.

- Short version: randomly sample from K(p) (kinetic, momentum), calculate U(k) (potential, position*) — that's your Metropolis proposal.

1.2.3 Optimizing in Stan

'Big n' computational burdens can also be mitigated somewhat, by certain program specific coding techniques. These techniques, as well as other techniques aimed to encourage model convergence, bolstered our modelling efforts. We highlight some such techniques for Stan here, and include the complete .stan script in Appendix A for reference.

Bayesian Motivated Techniques

Stan allows a user to omit prior distributions for parameters, but interprets non-inclusion as a non-informative, uniform prior. However, Gelman [2016], the first Stan developer, pointed out in correspondence that the exponential covariance length-scale parameter, ϕ in equation (3), requires an informative prior for model identifiability (CITATION?). Trangucci [2016] recommended, in particular, a sharp tailed prior distribution for the length-scale parameter, such as the normal or log-normal, to act as soft upper and lower bound constraints[3]. Even further, for practical computing time and convergence considerations, Trangucci [2016] said complex models such as spatial hierarchical models require proper priors for all β coefficients. Using the intial GLM estimates to inform coefficient prior distributions had exactly the intended effects.

Computational and Linear Algebra Motivated Techniques

For speed and efficiency, the Stan Users Manual recommends pure matrix algebra and vectors, over 'for loops' and scalars Stan Development Team [2016]. For example,

```
hit ~ bernoulli_logit(X*beta + Z)
```

is faster than

```
for (n in 1:N)
        hit[n] = bernoulli_logit(X[n]*beta[n] + Z[n]);
```

[3] "Without stronger priors on l, GP can act as a second constant term in your regression for large draws of length-scale and large draws of alpha."

Notice that $N \times 1$ column vectors `hit`, `beta`, and `Z` replace scalars `hit[n]`, `beta[n]`, and `Z[n]`; and $N \times p$ matix `X` replaces $1 \times p$ row vector `X[n]`.

Trangucci [2016] also suggested a QR factorization on covariate matrix $\boldsymbol{X}$, in the linear predictor, to increase computational efficiency. A QR factorization consists of factoring an $n \times p$ matrix into the product of an $n \times p$ orthogonal matrix $\boldsymbol{Q}$ and a $p \times p$ upper triangular matrix $\boldsymbol{R}$, such that $\boldsymbol{X} = \boldsymbol{QR}$.

$$\boldsymbol{X} = \boldsymbol{QR} \tag{10}$$

$$\boldsymbol{X\beta} = \boldsymbol{QR\beta} \tag{11}$$

To reparameterize for model fitting, let $\boldsymbol{\theta} = \boldsymbol{R\beta}$, so that $\boldsymbol{\beta} = \boldsymbol{R^{-1}\theta}$, which gives

$$\boldsymbol{X\beta} = \boldsymbol{Q\theta}, \text{ and} \tag{12}$$

$$\text{logit}(p_{ij}|\boldsymbol{s}_{ij}) = \boldsymbol{Q}_{ij}(\boldsymbol{s}_{ij})\boldsymbol{\theta}_j + w_{ij}. \tag{13}$$

Now prior information about $\boldsymbol{\beta}$ should be incorporated into and given in the prior distributions of $\boldsymbol{\theta}$. Consider non-informative prior distributions on p dimensional parameter vector $\boldsymbol{\beta}$,

$$\boldsymbol{\beta} \sim N(\boldsymbol{0}, \sigma^2 \boldsymbol{I}_p),$$

where $\boldsymbol{I}_p$ is the $p \times p$ identity matrix, and $\boldsymbol{0}$ is a $p \times 1$ zero vector. Notice the intended variance of the non-informative prior must be modified for $\boldsymbol{\theta}$.

$$\begin{aligned} \text{Var}(\boldsymbol{\theta}) &= \text{Var}(\boldsymbol{R\beta}) & (14)\\ &= \boldsymbol{R}\text{Var}(\boldsymbol{\beta})\boldsymbol{R}' & (15)\\ &= \boldsymbol{R}\sigma^2\boldsymbol{I}_p\boldsymbol{R}' & (16)\\ &= \sigma^2\boldsymbol{RR}' & (17)\end{aligned}$$

We add noise to the covariance matrix diagonal with the following snippet of code.

```
for (n in 1:N)
  Sigma[n, n] = Sigma[n, n] + 1e-6;
```

This added diagonal noise guarantees that the covariance matrix assembled by `cov_exp_quad(...)` remains numerically positive-definite Trangucci [2017]. This `cov_exp_quad(...)` function can generate numerically non-positive-definite matrices when operating at high dimensions.

Finally, Carpenter [2016] recommended a Cholesky decomposition and tactical reparameterization, noting the efficiency of a vectorized scalar approach.

```
L = cholesky_decompose(Sigma);
Z ~ normal(0, 1);
Z_mod = L * Z;
hit ~ bernoulli_logit(Q*theta + Z_mod);
```

The first line performs a Cholesky decomposition on the covariance matrix `Sigma`. A Cholesky decomposition factors symmetric matrix `Sigma` such that $\Sigma = \mathbf{LL}'$. The second, "vectorized scalar" line generates n standard normal random variables, by reusing `normal(0, 1)` for every element of `Z`. These two lines remove the dependence of random vector $\mathbf{Z}$, which must be generated, on unknown parameters to be estimated Trangucci [2017]. The third line transforms `Z` to have the desired distribution. Note that $\text{Var}(\mathbf{LZ}) = \mathbf{L}\text{I}_n\mathbf{L}' = \Sigma$, so that $\boldsymbol{LZ} \sim N(\mathbf{0}, \Sigma)$ as desired.

Evaluate Inverse you say?? Yes.

- $\text{logit}\{\text{EY(s)}\} = \boldsymbol{X}(s)\boldsymbol{\beta} + Z(s)$, with $Z(s) \sim MVN\{\mathbf{0}, \Sigma_s\}$

- $f(\boldsymbol{\beta}, \phi, \sigma^2, \boldsymbol{Z}|\boldsymbol{Y}) \propto f(\boldsymbol{Y}|\boldsymbol{\beta}, \phi, \sigma^2, \boldsymbol{Z})f(\boldsymbol{\beta})f(\boldsymbol{Z}|\phi, \sigma^2)f(\phi)f(\sigma^2)$

- M-H proposal, iteration i: $Z_{10,i}$

$$r = \frac{f(Z_{10,i}|\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}, \boldsymbol{\beta}_{i-1}, \phi_i, \sigma_i^2)}{f(Z_{10,i-1}|\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}, \boldsymbol{\beta}_{i-1}, \phi_i, \sigma_i^2)}$$

- Note: $f(z_1, z_2, z_3|\boldsymbol{Y}) = f(z_1|z_2, z_3, \boldsymbol{Y})f(z_2, z_3|\boldsymbol{Y})$. So...

$$r \propto \frac{f(\boldsymbol{Y}|\boldsymbol{\theta}_i)f(\boldsymbol{\beta})f(Z_{10,i}|\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}, \phi_i, \sigma_i^2)f(\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}|\phi_i, \sigma_i^2)f(\phi)f(\sigma^2)}{f(\boldsymbol{Y}|\boldsymbol{\theta}_{i-1})f(\boldsymbol{\beta})f(Z_{10,i-1}|\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}, \phi_i, \sigma_i^2)f(\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}|\phi_i, \sigma_i^2)f(\phi)f(\sigma^2)}$$

$$r \propto \frac{f(\boldsymbol{Y}|\boldsymbol{\theta}_i)f(Z_{10,i}|\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}, \phi_i, \sigma_i^2)}{f(\boldsymbol{Y}|\boldsymbol{\theta}_{i-1})f(Z_{10,i-1}|\boldsymbol{Z}_{1:9,i}, \boldsymbol{Z}_{11:n,i-1}, \phi_i, \sigma_i^2)}$$

- And $f(\boldsymbol{Z})$, $f(Z_i|\boldsymbol{Z}_{-i})$, etc. are $MVN\{\cdot, \Sigma^*\}$, where Σ^* either is, or is some function of, $\Sigma_{\boldsymbol{s}}$; with PDF kernal containing $\Sigma_{\boldsymbol{s}}^{*-1}$, (containing ϕ_i, σ_i^2).

1.3 Dimension Reduction; Predictive Process Models

Predictive process models (PPMs) provide a method for attempting to circumvent the “big N problem” in the case of Bayesian hierarchical models with latent Gaussian random effects. Numerous methods exist, for example [Cressie and Johannesson, 2008], (CITE OTHERS), but predictive process models provide a competitive modeling approach with computational advantages for hierarchical models with a Gaussian random field (GRF) at the second level of specification [Banerjee et al., 2008]. Latent GRFs prove challenging because they are only implicitly observed, through a binomial response in our case. This means the GRF parameters (hyperparameters) and their prior distributions comprise the third level of the hierarchical model. Gaussian PPMs achieve dimension reduction by projecting the original process onto a lower dimensioned subspace, at a set of locations called knots [Banerjee et al., 2008].

1.3.1 PPM Procedure

Consider again the SGLMM (5), and Gaussian random field $\boldsymbol{w}(\boldsymbol{s})$.

$$\text{logit}(p_{ij}|\boldsymbol{s}_{ij}) = \boldsymbol{X}_{ij}(s_{ij})\boldsymbol{\beta}_j + w(\boldsymbol{s}_{ij}) \tag{18}$$

$$\boldsymbol{w}(\boldsymbol{s})|\boldsymbol{\theta} \sim \text{GRF}(\boldsymbol{0}, \boldsymbol{C}(\boldsymbol{\theta})) \tag{19}$$

Define (20) as in (5), and (21) shows $n \times 1$ vector of random effects $\boldsymbol{w}$ at locations $\boldsymbol{s}$, conditioned on covariance parameters $\boldsymbol{\theta}$, constitutes a GRF. Let $n \times 1$ zero vector $\boldsymbol{0}$ be the stationary GRF mean, with $n \times n$, symmetric, positive-definite covariance matrix $\boldsymbol{C}(\boldsymbol{\theta})$. Let $C(\boldsymbol{s}_i, \boldsymbol{s}_j; \boldsymbol{\theta})$ denote the covariance of random effects at locations $\boldsymbol{s}_i$ and $\boldsymbol{s}_j$, so that $C(\boldsymbol{\theta}) = [C(\boldsymbol{s}_i, \boldsymbol{s}_j; \boldsymbol{\theta})]_{i,j=1}^{n}$.

To define the PPM, start with knot locations. Let $\boldsymbol{S}^* = \{\boldsymbol{s}_1^*, \ldots, \boldsymbol{s}_m^*\}$ be a set of $m < n$ chosen knot locations, which may or may not be a subset of observed locations. We denote knot location random effects with $m \times 1$ vector $\boldsymbol{w}^* = [w(\boldsymbol{s}_i^*)]_{i=1}^{m}$, and the $m \times m$ knot covariance matrix and its elements as $\boldsymbol{C}^*(\boldsymbol{\theta}) = \left[C(\boldsymbol{s}_i^*, \boldsymbol{s}_j^*)\right]_{i,j=1}^{m}$. The knot random effects form a distinct m-dimensional GRF.

$$\boldsymbol{w}^*|\boldsymbol{\theta} \sim \text{GRF}\{\boldsymbol{0}, \boldsymbol{C}^*(\boldsymbol{\theta})\} \tag{20}$$

The predictive process modelling procedure uses the m selected knots, the covariance structure of the parent process, and kriging to interpolate w at site $\boldsymbol{s}_0$ [Schabenberger and Gotway, 2004]; See Appendix ?? for kriging details. Let $\tilde{w}(\boldsymbol{s}_0)$ represent this interpolated random effect, and let $\boldsymbol{c}(\boldsymbol{s}_0; \boldsymbol{\theta}) = \left[C(\boldsymbol{s}_0, \boldsymbol{s}_j^*; \boldsymbol{\theta})\right]_{j=1}^{m}$ be an $m \times 1$ covariance vector giving the covariance of the $\boldsymbol{s}_0$ random effect with the knot random effects.

$$\tilde{w}(\boldsymbol{s}_0) = E[w(\boldsymbol{s}_0)|\boldsymbol{w}^*] \tag{21}$$

$$= \boldsymbol{c}^T(\boldsymbol{s}_0;\boldsymbol{\theta}) \cdot \boldsymbol{C}^{*-1}(\boldsymbol{\theta}) \cdot \boldsymbol{w}^* \tag{22}$$

For a GRF, the weights of linear combination (24) minimize the squared error loss function among all linear predictors [Schabenberger and Gotway, 2004]; and notice that the linear combination varies spatially. Accordingly, predictive process $\tilde{w}(\boldsymbol{s})$ defines another GRF and covariance matrix.

$$\tilde{\boldsymbol{w}}(\boldsymbol{s}) \sim \text{GRF}\{0, \tilde{C}(\cdot)\} \tag{23}$$

$$\tilde{C}(\boldsymbol{s}, \boldsymbol{s}';\boldsymbol{\theta}) = \boldsymbol{c}^T(\boldsymbol{s};\boldsymbol{\theta}) \cdot \boldsymbol{C}^{*-1}(\boldsymbol{\theta}) \cdot \boldsymbol{c}(\boldsymbol{s}';\boldsymbol{\theta}) \tag{24}$$

To reiterate, $m \times 1$ vector $\boldsymbol{c}(\boldsymbol{s};\boldsymbol{\theta}) = \left[C(\boldsymbol{s}, \boldsymbol{s}_j^*)\right]_{j=1}^m$ gives the covariance of the random effect at $\boldsymbol{s}$ with knot random effects. Finally, the predictive process model:

$$\text{logit}(p_{ij}|\boldsymbol{s}_{ij}) = \boldsymbol{X}_{ij}(\boldsymbol{s}_{ij})\boldsymbol{\beta}_j + \tilde{w}(\boldsymbol{s}) \tag{25}$$

1.3.2 Improved Predictive Process Models

1.4 Approximation; SPDE and INLA

Integrated Nested Laplace Approximation (INLA), a mathematically intensive and computationally geared approximation technique, works well for Bayesian hierarchical models with latent Gaussian **markov** random fields (GMRFs) [Rue and Martino, 2007]. A GMRF, by virtue of its sparse precision matrix, enables INLA's orders of magnitude faster approximation method. However, to use INLA for continuous domain spatial models with latent Gaussian random fields (GRFs), one must represent the GRF as a GMRF; and stochastic calculus provides a link. A particular stochastic partial derivative of a Matern GRF equals a Gaussian random field white noise process. This identity provides a platform on which to approximate a Matern GRF with a GMRF, in the form of a piecewise linear basis representation. The discrete representation consists of deterministic basis functions, defined by a triangulation of the domain, and GMRF weights. The basis representation has a sparse precision matrix, and thus qualifies for the INLA approximation and its computational advantages. Scientists use this GRF to GMRF translation process—projecting the SPDE onto the basis representation—known as Finite Element Method, extensively in other fields.

1.4.1 Gaussian Markov Random Fields

1.4.2 Stochastic Partial Differential Equation (SPDE)

The exponential convariance function is a member of the larger Matern family, defined by the following covariance function.

$$\mathrm{C}(h) = \frac{\sigma^2}{2^{\nu-1}\Gamma(\nu)}(\kappa h)^{\nu} K_{\nu}(\kappa h)$$

This parameterization includes range parameter $\kappa > 0$, smoothness parameter $\nu > 0$, scale parameter σ^2, and modified Bessel function $K_{\nu}(\cdot)$ [Schabenberger and Gotway, 2004]. While $\nu = 1/2$ defines the exponential covariance function, Whittle [1954] declared Matern($\nu = 1$) the "elementary correlation" function in two dimensions. Both functions yield a similarly decaying-with-distance spatial covariance, but a Matern random field solves the following SPDE [Whittle, 1954].

$$(\kappa^2 - \Delta)^{\alpha/2} x(\boldsymbol{s}) = \mathcal{W}(\boldsymbol{s}),$$

This includes $\Delta = \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2}$, the Laplace operator; spatial scale parameter κ, as in the Matern; smoothness parameter α; and Gaussian spatial white noise process $\mathcal{W}(\boldsymbol{s})$. The particular SPDE and Matern coupling dictates $\alpha = \nu + d/2$, where d = 2 for $\mathbb{R}^2$; and

$$\sigma^2 = \frac{\Gamma(\nu)}{\Gamma(\alpha)(4\pi)^{d/2}\kappa^{2\nu}}.$$

Based on Whittle [1954] and Mondal [2017] we use $\nu = 1$, which implies $\alpha = 2$. This specification simplifies the SPDE to

$$(\kappa^2 - \Delta) x(\boldsymbol{s}) = \mathcal{W}(\boldsymbol{s});$$

the Matern covariance to

$$\mathrm{C}(h) = \sigma^2(\kappa h) K_1(\kappa h);$$

and the variance to $\sigma^2 = \frac{1}{4\pi\kappa}$.

Piecewise Linear Basis Representation

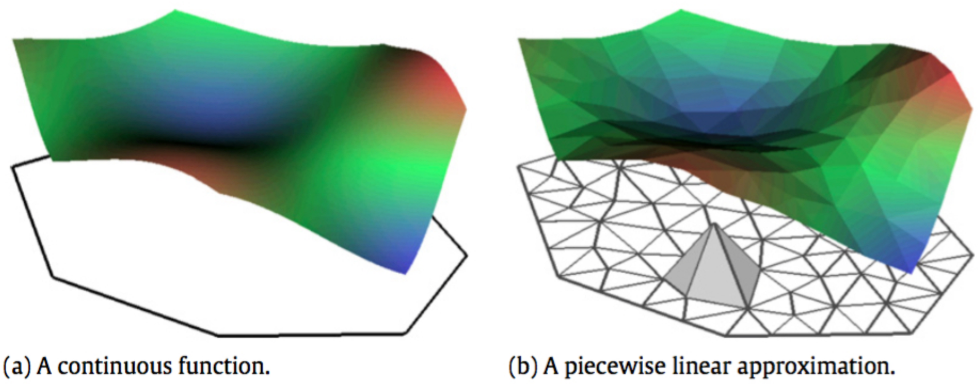
The next step, known as the Finite Element Method, projects the SPDE onto a piecewise linear basis representation [Simpson et al., 2012]. The basis represenation,

$$x(\boldsymbol{s}) = \sum_{k=1}^{n} \psi_k(\boldsymbol{s}) x_k,$$

contains deterministic basis functions, $\psi_k(\cdot)$; and weights $\boldsymbol{x} = \{x_1, \ldots, x_n\}$, which constitute a GMRF. The two combine so that the distribution of $x(\boldsymbol{s})$ approximates the Matern GRF that solves the SPDE, but retains a sparse precision matrix and its accordant computational advantages.

Deterministic Basis Function

A triangulation of the domain defines the deterministic component, $\psi_k(\boldsymbol{s})$, of the basis representation. Figure 12 [Simpson et al., 2012] illustrates how domain triangulation and determines a basis function.



(a) A continuous function.

(b) A piecewise linear approximation.

Figure 1: A Gaussian Markov random field, defined as the piecewise linear basis function $x(\boldsymbol{s}) = \sum_{k=1}^{n} \psi_k(\boldsymbol{s}) x_k$, approximates a Matern GRF. This image illustrates how a triangular mesh over the domain determines basis functions $\psi_k(\boldsymbol{s})$ [Simpson et al., 2012].

Keep in mind that a realization of a GRF is essentially a function, and Figure 12 shows how a discrete piecewise linear basis function can approximate a continuous function [Simpson et al., 2012]. We have $\psi_k(\boldsymbol{s}) = 1$ at the kth vertex, 0 at all other vertices, and surface function values for triangle interior points are linear combinations of the three home triangle vertices.

GMRF Weights

Stochastic calculus identities provide a way to calculate the weights in the basis representation. Define $\langle f, g\rangle = \int f(\boldsymbol{u})g(\boldsymbol{u})d\boldsymbol{u}$, and find weights $\boldsymbol{x}$ such that

$$\left[\left\langle \phi_k, (\kappa^2 - \Delta)^{\alpha/2}\boldsymbol{x}\right\rangle\right]_{k=1,\ldots,n} \overset{D}{=} \left[\langle \phi_k, \mathcal{W}\rangle\right]_{k=1,\ldots,n},$$

for a set of test functions ϕ_k [Lindgren et al., 2011]. The appropriate weight vector $\boldsymbol{x}$ gives the stochastic weak solution solution to the SPDE [Mao, 2007], Lindstrom [2014]; we use the Galerkin solution, with $\alpha = 2$ and $\phi_i = \psi_i$ [Lindgren et al., 2011]. Replace $\boldsymbol{x}$ with basis function representation $\Sigma_k \psi_k w_k$,

$$\left[\left\langle \phi_i, (\kappa^2 - \Delta)^{\alpha/2}\psi_j\right\rangle\right]_{i,j} \boldsymbol{w} \overset{D}{=} \left[\langle \phi_k, \mathcal{W}\rangle\right]_{k},$$

and let $\alpha = 2$ and $\phi_i = \psi_i$ as in the Galerkin solution:

$$\Big(\kappa^2[\langle \psi_i, \psi_j\rangle] + [\langle \psi_i, -\Delta\psi_j\rangle]\Big)\boldsymbol{w} \overset{D}{=} \Big[\langle \psi_k, \mathcal{W}\rangle\Big].$$

Let $\boldsymbol{C}_{i,j} = \langle \psi_i, \psi_j\rangle$, and $\boldsymbol{G}_{i,j} = \langle \psi_i, -\Delta\psi_j\rangle$, so that

$$\left(\kappa^2\boldsymbol{C} + \boldsymbol{G}\right)\boldsymbol{w} \overset{D}{=} N(\boldsymbol{0}, \boldsymbol{C}).$$

For $\boldsymbol{w} \sim N(\boldsymbol{0}, \boldsymbol{Q}^{-1})$, we have then

$$\boldsymbol{Q}_\kappa = \left(\kappa^2\boldsymbol{C} + \boldsymbol{G}\right)^T \boldsymbol{C}^{-1}\left(\kappa^2\boldsymbol{C} + \boldsymbol{G}\right).$$

However, $\boldsymbol{C}_{ij}^{-1}$ has a sparse precision matrix, so replace $\boldsymbol{C}$ with diagonal matrix $\widetilde{\boldsymbol{C}}$,

$$\widetilde{\boldsymbol{C}}_{i,i} = \langle \psi_i, \boldsymbol{1}\rangle = \int \psi_i(\boldsymbol{s})d\boldsymbol{s}.$$

Note that this solution provides the distribution of the weights x_k, not $x(\boldsymbol{s})$ itself, in the basis representation

$$x(\boldsymbol{s}) = \sum_{k=1}^{n} \psi_k(\boldsymbol{s})x_k.$$

With this representation achieved, via the SPDE link, we move on to the approximation procedure that we aim for.

1.4.3 Integrated Nested Laplace Approximations (INLA)

INLA proceeds through a carefully constructed and calibrated series of calculations and approximations, to acheive estimates of key quantites. These quantites include the maginal posterior distributions for latent field parameters, $p(x_i|\boldsymbol{y})$; and covariance hyperparameter posterior $p(\theta|\boldsymbol{y})$. This means we never obtain posteriors $p(\boldsymbol{x}|\boldsymbol{y})$ and $p(\boldsymbol{x},\boldsymbol{\theta}|\boldsymbol{y})$.

Step 1, Gaussian Approximation

The basic approach to estimating $p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$ is "matching the mode and curvature at the mode" of estimator $p_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$ to that of $p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$ [Rue and Held, 2005]. As mentioned, INLA requires Gaussian priors for all paramters except covariance hyperparameters; but, INLA also requires conditional independence, whereby $p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}) = \prod_i p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta})$. Our analysis satisfies this necessity, and therefore

$$p(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}) \propto \exp\left(-\frac{1}{2}\boldsymbol{x}^T\boldsymbol{Q}\boldsymbol{x} + \sum_i \log\, p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta})\right).$$

The Gaussian approximation takes the form

$$p_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y}) \propto \exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T(\boldsymbol{Q}+\mathrm{diag}(\boldsymbol{c}))(\boldsymbol{x}-\boldsymbol{\mu})\right),$$

where vectors $\boldsymbol{c}$ and $\boldsymbol{\mu}$ depend on second order Taylor expansions of $f(\boldsymbol{x}) = \sum_i \log\, p(y_i|\boldsymbol{x}_i,\boldsymbol{\theta})$ about the mode [Lindstrom, 2014]. A Newton-Raphson algorithm iteratively computes the mode and precision matrix until convergence [Rue et al., 2009]. Step 2 uses this Gaussian approximation, $p_G(\boldsymbol{x}|\boldsymbol{\theta},\boldsymbol{y})$.

Step 2, Laplace Approximation

This step begins with two sides of a familiar identity, and its subsequent rearrangement.

$$p(\boldsymbol{y},\boldsymbol{x}|\boldsymbol{\theta}) = p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta}) \tag{26}$$

$$p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta}) = p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})p(\boldsymbol{y}|\boldsymbol{\theta}) \tag{27}$$

$$\frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})} = p(\boldsymbol{y}|\boldsymbol{\theta}) \tag{28}$$

We use this formulation of $p(\boldsymbol{y}|\boldsymbol{\theta})$ next, in the familiar Bayesianian proporionality.

$$\begin{aligned} p(\theta|\boldsymbol{y}) &\propto p(\boldsymbol{y}|\boldsymbol{\theta})p(\boldsymbol{\theta}) && (29) \\ &\propto \frac{p(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta})p(\boldsymbol{x}|\boldsymbol{\theta})}{p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})} \cdot p(\boldsymbol{\theta}) && (30) \end{aligned}$$

For a given $\boldsymbol{\theta}$, let $\boldsymbol{x}_0 = \text{argmax}_x p(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})$. Then,

$$p(\boldsymbol{\theta}|\boldsymbol{y}) \approx \tilde{p}(\boldsymbol{\theta}|\boldsymbol{y}) \propto \frac{p(\boldsymbol{y}|\boldsymbol{x}_0,\boldsymbol{\theta})p(\boldsymbol{x}_0|\boldsymbol{\theta})}{p_G(\boldsymbol{x}_0|\boldsymbol{y},\boldsymbol{\theta})} \cdot p(\boldsymbol{\theta}),$$

where the Taylor approximation of $f(\boldsymbol{x}) = \sum_i \log p(y_i|x_i)$, in $p_G(\boldsymbol{x}|\boldsymbol{y},\boldsymbol{\theta})$, expands about $\boldsymbol{x}_0$. This approximation matches Tierney and Kadane [1986] Laplace approximation. Then, we have approximate maximum likelihood estimate $\hat{\boldsymbol{\theta}}_{\text{ML}} \approx \text{argmax}_\theta \tilde{p}(\boldsymbol{\theta}|\boldsymbol{y})$.

Step 3, Numerical Integration

Numerical Integration over $\boldsymbol{\theta}$, or elements of $\boldsymbol{\theta}$, gives $p(x_i|\boldsymbol{y})$ and $p(\theta_i|\boldsymbol{y})$.

$$p(x_j|\boldsymbol{y}) \approx \int p_{\text{G}}(x_j|\boldsymbol{\theta},\boldsymbol{y})\tilde{p}(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}$$

$$p(\theta_k|\boldsymbol{y}) \approx \int \tilde{p}(\boldsymbol{\theta}|\boldsymbol{y})d\boldsymbol{\theta}_{-k}$$

In summary, from a continuous domain the SPDE-INLA approxmation procedure provides, with improved speed, posterior estimates for $p(\boldsymbol{\theta}|\boldsymbol{y})$, $p(\theta_i|\boldsymbol{y})$, and $p(x_i|\boldsymbol{y})$. The R package `INLA` implements this procedure, with flexible SPDE specifications for domain triangulation.

1.4.4 Bayesian Inference in R-INLA

References

Waldo R Tobler. A computer movie simulating urban growth in the detroit region. Economic geography, 46 (sup1):234–240, 1970.

Sudipto Banerjee, Alan E Gelfand, Andrew O Finley, and Huiyan Sang. Gaussian predictive process models for large spatial data sets. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70 (4):825–848, 2008.

Alan E Gelfand, Peter Diggle, Peter Guttorp, and Montserrat Fuentes. Handbook of spatial statistics. CRC press, 2010.

Murali Haran. Gaussian random field models for spatial data. Handbook of Markov Chain Monte Carlo, pages 449–478, 2011.

Sudipto Banerjee, Bradley P Carlin, and Alan E Gelfand. Hierarchical modeling and analysis for spatial data. Crc Press, 2014.

Finn Lindgren, Håvard Rue, and Johan Lindström. An explicit link between gaussian fields and gaussian markov random fields: the stochastic partial differential equation approach. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 73(4):423–498, 2011.

Radford M Neal et al. Mcmc using hamiltonian dynamics. Handbook of Markov Chain Monte Carlo, 2: 113–162, 2011.

Nicholas Metropolis, Arianna W Rosenbluth, Marshall N Rosenbluth, Augusta H Teller, and Edward Teller. Equation of state calculations by fast computing machines. The journal of chemical physics, 21(6):1087–1092, 1953.

Berni J Alder and T.E. Wainwright. Studies in molecular dynamics. i. general method. The Journal of Chemical Physics, 31(2):459–466, 1959.

Simon Duane, Anthony D Kennedy, Brian J Pendleton, and Duncan Roweth. Hybrid monte carlo. Physics letters B, 195(2):216–222, 1987.

Radford M Neal. Priors for infinite networks. In Bayesian Learning for Neural Networks, pages 29–53. Springer, 1996.

Andrew Gelman. personal communication, November 2016.

Rob Trangucci. personal communication, November 2016.

Stan Development Team. Stan Modeling Language User's Guide and Reference Manual, Version 2.14.1, 2016. URL `http://mc-stan.org/`.

Rob Trangucci. Hierarchical gaussian processes in stan. StanCon Contributed Talks, 2017. Conference.

Bob Carpenter. personal communication, November 2016.

Noel Cressie and Gardar Johannesson. Fixed rank kriging for very large spatial data sets. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70(1):209–226, 2008.

Oliver Schabenberger and Carol A Gotway. Statistical methods for spatial data analysis. CRC press, 2004.

HÅvard Rue and Sara Martino. Approximate bayesian inference for hierarchical gaussian markov random field models. Journal of statistical planning and inference, 137(10):3177–3192, 2007.

Peter Whittle. On stationary processes in the plane. Biometrika, pages 434–449, 1954.

Debashis Mondal. Personal communication, February 2017.

Daniel Simpson, Finn Lindgren, and Håvard Rue. Think continuous: Markovian gaussian models in spatial statistics. Spatial Statistics, 1:16–29, 2012.

Xuerong Mao. Stochastic differential equations and applications. Elsevier, 2007.

Johan Lindstrom. Gaussian markov random fields. Lecture Slides, 2014. Pan-American Advanced Study Institute, Buzios.

Havard Rue and Leonhard Held. Gaussian Markov random fields: theory and applications. CRC Press, 2005.

Håvard Rue, Sara Martino, and Nicolas Chopin. Approximate bayesian inference for latent gaussian models by using integrated nested laplace approximations. Journal of the royal statistical society: Series b (statistical methodology), 71(2):319–392, 2009.

Luke Tierney and Joseph B Kadane. Accurate approximations for posterior moments and marginal densities. Journal of the american statistical association, 81(393):82–86, 1986.