Journal of Statistical Software

January 2015, Volume 63, Issue 12. *http://www.jstatsoft.org/*

Model-Based Geostatistics the Easy Way

Patrick E. Brown
Cancer Care Ontario and University of Toronto

Abstract

This paper briefly describes geostatistical models for Gaussian and non-Gaussian data and demonstrates the **geostatsp** and **dieasemapping** packages for performing inference using these models. Making use of R's spatial data types, and raster objects in particular, makes spatial analyses using geostatistical models simple and convenient. Examples using real data are shown for Gaussian spatial data, binomially distributed spatial data, a log-Gaussian Cox process, and an area-level model for case counts.

Keywords: spatial statistics, geostatistics, R, INLA, Bayesian inference, kriging.

1. Introduction

In the past two decades spatial statistics has gradually become a mature and established branch of statistics with a suite of well defined models and proven inference methodologies capable of addressing a wide range of practical problems. The capability of R (R Core Team 2014) to store, manipulate, and display spatial data has similarly improved, and as a result spatial methodologies which were formerly only accessible to the specialist are available to the wider statistical community. This paper demonstrates model fitting for Gaussian, non-Gaussian, and point process data using the **geostatsp** and **diseasemapping** packages, with R's spatial data classes being used to make spatial data analysis simple and the software intuitive.

1.1. Models and methods

Models and theory for Gaussian spatial data were first espoused by Matheron (1962) and popularized by Cressie (1993). Writing $U(s)$ as the value of a Gaussian random field U at location s, the basic (stationary) geostatistical model is characterized by the joint multivariate normal distribution

$$[U(s_1) \dots U(s_N)]^\top \sim \text{MVN}(0, \Sigma).$$

The entries of the covariance matrix Σ are determined by a spatial correlation function ρ with

$$\Sigma_{ij} = \text{cov}[U(s_i), U(s_j)] = \sigma^2 \rho[(s_i - s_j)/\phi, \theta].$$

Here ϕ is a scale parameter controlling the rate at which correlation decays with distance, and θ is a vector of possible additional parameters (controlling directional effects, for example). An isotropic process has correlation being a function of distance with $\rho[(s_i - s_j)/\phi] = \rho_0(||s_i - s_j||/\phi)$.

Various parametric functions have been used for ρ, and Stein (1999) makes a compelling case for the Matérn correlation function described in Appendix A. An isotropic process with a Matérn correlation has a single additional parameter κ controlling the differentiability of the process. Two additional covariance parameters commonly used refer to geometric anisotropy, and comprise an angle of rotation indicating a preferred direction and a ratio parameter giving the ratio of the ranges on the two axes.

The parametrisation of the Matérn is different in each of the **geoR** (Ribeiro and Diggle 2001), **RandomFields** (Schlather, Malinowski, Menck, Oesting, and Strokorb 2015) and **geostatsp** packages. The specification of the Matérn in Appendix A, and in use in the **geostatsp** package, has the property that when varying κ the correlation at a distance ϕ stays fairly close to 0.14, or $\rho[(0, \phi)/\phi, \kappa] \approx 0.14$. A Matérn with $\kappa = \infty$ is a Gaussian density with ϕ being two standard deviations. The term 'practical range' is used at times to describe ϕ as defined here, interpreting ϕ as a distance beyond which correlation is 'small' is a manner analogous to interpreting the Gaussian density as being 'small' beyond two standard deviations.

The anisotropy angle refers to rotation of the coordinates anti-clockwise by the specified amount prior to calculating distances, which has the effect that the contours of the correlation function appear rotated clockwise by this amount. The anisotropy ratio is the amount the Y coordinates are divided by by following rotation, with large values making the Y coordinates smaller and increasing the correlation in the Y direction (of the rotated coordinates).

Gaussian data

Data Y_i observed at location s_i with covariates $X(s_i)$ is often modelled with the linear geostatistical model (LGM):

$$\begin{aligned} Y_i|U(s_i) \sim & \text{N}(\lambda(s_i), \tau^2) \\ \lambda(s_i) = & \mu + \beta X(s_i) + U(s_i). \end{aligned} \tag{1}$$

Although method-of-moments estimation of the covariance parameters ϕ, σ and τ is still common, Stein (1999) makes a thorough argument for using maximum likelihood estimates (MLEs). Writing $\psi = (\mu, \beta, \sigma, \tau, \phi)$, the MLEs $\hat{\psi}$ are the quantities which maximize the likelihood $pr(Y_1 \dots Y_N; \psi)$. The Y_i are jointly multivariate normal and the likelihood is tractable, albeit requiring the inversion of an N by N matrix, and numerical optimizers such as the `optim` function can be used to find $\hat{\psi}$.

Spatial prediction usually involves covering the study region with a large number of regularly spaced points $g_\ell; \ell = 1 \dots L$ and mapping estimates of $\bar{U} = [U(g_1) \dots U(g_L)]$ or $\bar{\lambda} = [\lambda(g_1) \dots \lambda(g_L)]$. As the model is linear and Gaussian, the conditional distribution $[\bar{U}|Y]$ is multivariate normal with closed form expressions for the conditional mean and variance. The MLEs $\hat{\psi}$ are used to calculate these expressions, hence the uncertainty in these parameter estimates is ignored (see Diggle and Ribeiro 2006).

Non-Gaussian data

When the observed data Y_i are non-Gaussian, the model above is extended to the generalized linear geostatistical model (GLGM) used by Diggle, Moyeed, and Tawn (1998) and further described in Diggle and Ribeiro (2006). Consider a distribution f (i.e., Binomial or Weibull) with a mean parameter λ and possibly additional parameters ν. Writing $g(\cdot)$ as a link function (i.e., log or logit), the GLGM takes the form

$$\begin{aligned} Y_i|U(s_i) \sim & f[\lambda(s_i), \nu] \\ g[\lambda(s_i)] = & \mu + \beta X(s_i) + U(s_i) \\ \text{cov}[U(s_i), U(s_j)] = & \sigma^2 \rho[(s_i - s_j)/\phi, \theta]. \end{aligned} \tag{2}$$

The combination of non-Gaussian data and an unobserved latent variable make the likelihood function intractable and computing the MLEs difficult. Bayesian inference using Markov chain Monte Carlo (MCMC) algorithms has become the most common method for making statistical inference with GLGMs, as was done in Diggle *et al.* (1998). Bayesian inference requires specifying prior distributions for the model parameters μ, β, σ and ϕ, with the posterior distributions $\pi(\phi|Y)$ and $\pi[U(s)|Y]$ forming the basis of inference.

The integrated nested Laplace approximation (INLA) algorithm of Rue, Martino, and Chopin (2009) is an alternative to MCMC for performing Bayesian with latent Gaussian models. MCMC's principal drawback is the requirement that chains of posterior samples must be monitored and assessed for convergence and mixing, and obtaining a set of reliable posterior samples from a MCMC algorithm can be difficult and require a specialized skill set to accomplish. INLA is much easier to use in this regard, and although it's maximisation step and numerical integration can sometimes require judicious choices of starting values and tuning parameters it is in general less labor-intensive to use than MCMC.

An additional recent development which has facilitated the implementation of the GLGM is the Markov random field approximation to the Matérn correlation function developed by Lindgren, Rue, and Lindström (2011). When the number of spatial locations N is large, inverting the variance matrix Σ can be time consuming or numerically unstable. Lindgren *et al.* (2011) use Gaussian Markov random fields (GMRF's) to derive a simple expression for Σ^{-1} for Matérn correlations using various forms of stochastic partial differential equations. The **geostatsp** package makes use of the Matérn approximation of GMRF's on grids of square cells with $\kappa = 1$ or 2. Although real datasets will rarely be sampled on a square lattice, the continuous surface $U(s)$ can be well approximated by superimposing a fine lattice over the study region and assigning each data point to a cell. The fact that many (or most) of the cells will not have data observed in them is not problematic for INLA. This combination of INLA with the Markov random field approximation has been to estimate spatial variation in risk for Lupus in the city of Toronto, Canada from case incident locations by Li, Brown, Rue, al Maini, and Fortin (2012), and for assessing the effect of cancer risk of ambient radiation near a nuclear power facility using time-to-event data from a retrospective cohort in Jiang, Brown, Rue, and Shimakura (2014).

Lindgren *et al.* (2011) derive a GMRF approximation for the Matérn using an irregular lattice with triangular basis functions, which has a number of advantages over the grid cell approach. This approximation is implemented in the **INLA** software, and incorporation of this feature into **geostatsp** is work in progress.

1.2. Spatial statistics and R

The **sp** package (see Bivand, Pebesma, and Gómez-Rubio 2013) and **raster** package (Hijmans 2014) provide an excellent set of facilities for storing, manipulating, and visualising spatial data. The **sp** package provides `SpatialPointsDataFrame` and `SpatialPolygonsDataFrame` objects for storing point and polygon data respectively, and are compatible with many of the standard data formats most geographical information systems (GIS) uses. The **raster** package provides similar tools for raster data, which are pixelated images or rectangular lattices. The **rgdal** (Bivand, Keitt, and Rowlingson 2014) package provides a set of tools for reading spatial data from various formats into R, such as ESRI shapefiles for point and polygon data, and GeoTIFF files for raster data. These three packages (along with **spdep**, Bivand 2014, and others) have made R fully compatible with GIS software and R fulfils many of the criteria for it to be called a GIS in its own right.

The venerable **geoR** package (see Diggle and Ribeiro 2006) has provided tools for likelihood-based inference since 2000, and is one of the very few software packages for spatial analysis which accommodates all of: the Matérn correlation function; covariates; Maximum Likelihood Estimation; geometric anisotropy; and the Box-Cox transform. Since **geoR** predates the **sp** and **raster** packages, it has its own spatial data types.

For Bayesian inference, the excellent **INLA** (Rue, Martino, Lindgren, Simpson, and Riebler 2013) package developed by the authors of Rue *et al.* (2009) and Lindgren *et al.* (2011) implements INLA for a wide variety of models, including spatial Gaussian Markov random field models. **INLA** has been designed with flexibility of model specification being a priority, a job **INLA** accomplishes to an astonishing degree albeit at the cost rendering some tasks relatively complex in comparison to other packages. One such example is specifying a Matérn correlation function, with spatial locations being specified as grid cell indexes rather than coordinates. A considerable amount of code can sometimes be necessary for converting **INLA** results from a spatial model into a format which can be mapped.

The **geostatsp** package provides a set of user-friendly functions for Gaussian spatial models and an easy interface to **INLA** for fitting non-Gaussian models, resulting in a powerful set of tools for model-based geostatistical analyses in R. Response variables and covariates are specified with formulas, with data provided as `Raster` or `SpatialPointsDataFrame` objects. The interface to **INLA** has more complex set of routines underlying it, with observations being allocated to cells in a Markov random field and linear combinations of parameters and latent variables for predicted spatial surfaces being defined. The spatial predictions obtained from these packages are raster objects, making them easy to display and overlay on background maps.

2. Model-based geostatistics through examples

The **geostatsp** and **diseasemapping** packages described in this paper are available from the Comprehensive R Archive Network at `http://CRAN.R-project.org/` and R-Forge at `http://R-Forge.R-project.org/R/?group_id=312`. They both depend on the **INLA** package obtainable from `http://R-INLA.org/`.

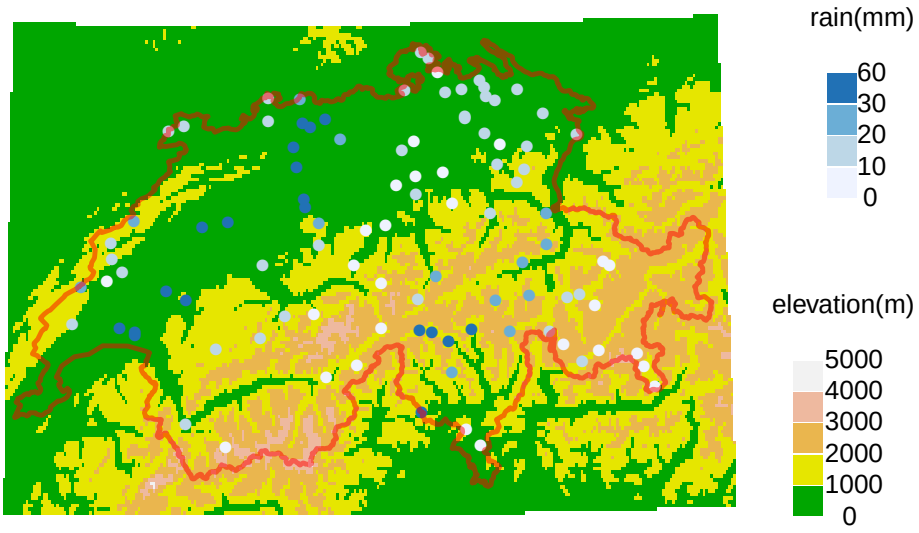


Figure 1: Swiss rainfall data (colored blue points and top legend) with elevation (background colors and bottom legend).

2.1. Maximum likelihood estimation and kriging

The Swiss rainfall dataset (see Diggle and Ribeiro 2006, 5.4.7) is a classic case study in Gaussian geostatistics. Loading of the **geostatsp** package and executing `data("swissRain")` makes available the following objects: a `SpatialPointsDataFrame` named `swissRain` of rain values at a number of points, a `SpatialPolygonsDataFrame` named `swissBorder` of the border of Switzerland; and a `Raster` object `swissAltitude` containing elevation values for Switzerland. These three objects are plotted in Figure 1.

Using the linear geostatistical model in (1) with these data would have the rainfall measurements being the Y_i, elevation values being $X(s)$, and $\lambda(s)$ as the unknown true rainfall surface. Either Bayesian or Frequentist inference could be used to fit the model, with the former possible in a manner similar to the example in the subsequent section. Frequentist inference is accomplished with the `lgm` function in the **geostatsp** package, which in turn calls `likfitLgm` for estimating the model parameters and `krige` for computing conditional means and variances of $U(s)$ and $\lambda(s)$. The Swiss rainfall data is fit with the code below.

```
R> names(swissRain)

[1] "ID"   "rain"

R> names(swissAltitude)

[1] "CHE_alt"

R> swissFit <- lgm(rain ~ CHE_alt, swissRain, grid = 120,
+    covariates = swissAltitude, shape = 1, fixShape = TRUE, boxcox = 0.5,
+    fixBoxcox = TRUE, aniso = TRUE)
R> names(swissFit)
```

	Estimate	Std. error	CI 0.025	CI 0.975	Estimated
(Intercept)	4.86	1.29	2.32	7.39	true
Elev'n per 1000m	0.28	0.37	−0.45	1.01	true
range, km	0.06		0.03	0.11	true
sdNugget	0.95		0.73	1.24	true
anisoAngleDegrees	37.00		31.74	42.27	true
anisoRatio	7.48		3.94	14.19	true
shape	1.00				false
boxcox	0.50				false
sdSpatial	2.97		1.89	4.68	true

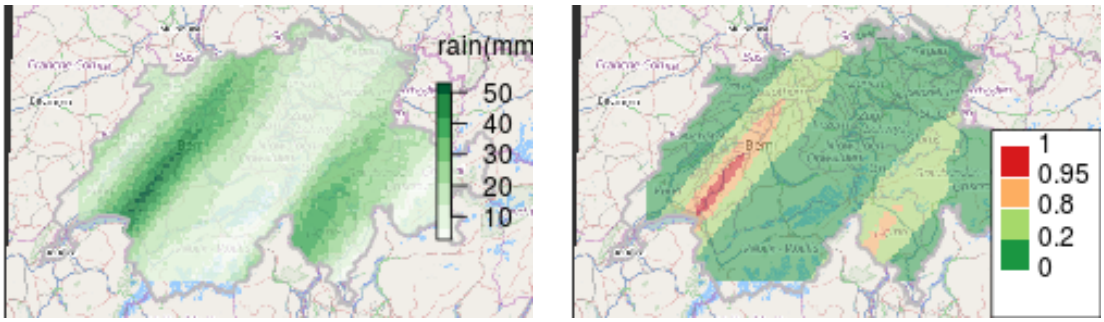
Table 1: Swiss rainfall parameter estimates, standard errors and confidence intervals obtained from a linear geostatistical model and the `lgm` function.

```
[1] "predict"  "param"    "varParam" "optim"
[5] "data"     "model"    "summary"
```

The `data` and `covariates` arguments contain the data required for fitting the model, with the fixed effects $\beta X(s)$ specified by `formula`. The variables listed in `formula` refer to names in either the `swissRain` or `swissAltitude` objects, and are not the names of the objects themselves. Variables in the right hand side of `formula` can refer to either: the name of a vector of values contained in the `data` argument; the name of a layer in a `Raster` object (a single layer, brick or stack) passed as `covariates`; or the name of one of the elements if `covariates` is a list of `Raster` objects. The latter is useful when covariate rasters have different resolutions and projections. If a covariate is a column in `data`, it will not be included in the predicted values for $\lambda(s)$.

The argument `grid = 120` specifies that spatial prediction should be done on a raster with 120 cells in the X dimension, with this raster having square cells covering the bounding box of `swissRain`. The `grid` argument can alternatively be supplied as a `Raster` object. A Matérn spatial correlation function with shape parameter fixed at 1 and a Box-Cox transform with parameter fixed at 0.5 (a square-root transform) are used. The `aniso = TRUE` argument allows for geometric anisotropy in the correlation function. Additional function arguments are `param` and `parscale`, starting values and parameter scaling values passed from `lgm` to `likfitLgm` and ultimately the numerical optimizer `optim`. The Swiss data has spatial locations expressed in a UTM projection, with coordinates in metres and consequently a spatial range parameter likely to be in the hundreds of thousands. The default scaling of 1 in `optim` would be ineffective and arguments on the order of `param = c(range = 10^5)` and `parscale = c(range = 10^4)` are in order. The default starting value and scale which `likfitLgm` sets for the range parameter are 1/20 and 1/200 of the diagonal distance of the bounding box of `data`.

The `swissFit` object produced by `lgm` is a list with elements including `predict`, a `RasterStack` of spatial predictions and standard errors, and `summary`, a table of parameter estimates and confidence intervals. Table 1 shows the `summary` component, with the range parameter converted to kilometres. The standard deviation parameters σ and τ are displayed in the `sdSpatial` and `sdNugget` rows respectively. Confidence intervals for the covariance parameters are derived from the observed information matrix, and will be missing if any of the



(a) Predicted rainfall $\mathrm{E}(\lambda(s)|Y)$ (b) Exceedance probabilities $pr(\lambda(s) > 30|Y)$

Figure 2: Conditional expectations and probabilities obtaind from fitting a linear geostatistical model to the Swiss rainfall using the `lgm` function.

estimated parameters are on a boundary. Notice the 'Estimated' column indicating that the Matérn shape parameter and Box-Cox transformation parameter were not estimated from the data.

Spatial predictions of the rainfall surface $\lambda(s)$ and the spatial random effect $U(s)$ are contained in the `RasterStack` element of `swissFit$predict`, which has the following layers:

```
R> names(swissFit$predict)

[1] "space"           "random"          "predict.boxcox"
[4] "krigeSd"         "predict"
```

Using the notation in (1), these layes are (in the order given above): the predicted fixed effects $\hat{\mu} + \hat{\beta}X(s)$; the kriged random effects $\mathrm{E}[U(s)|Y]$; the predicted rainfall surface $\mathrm{E}[\lambda(s)|Y]$ on the Box-Cox transformed scale; the prediction standard deviation $\mathrm{sd}[U(s)|Y]$; and predicted rainfall on the natural scale $\mathrm{E}\{[\alpha\lambda(s) + 1]^{1/\alpha}|Y\}$ with α being the Box-Cox transformation parameter. Figure 2a shows the predicted rainfall values (on the natural scale), and results from the command `plot(swissFit$predict[["predict"]])`. Notice the strong directionality is consistent with an angle of rotation of 37° and a ratio of the major to minor axes of 7.5.

Figure 2b shows the conditional probabilities that rainfall exceeds 30mm, computed with

```
R> exc30 <- excProb(swissFit, 30, nuggetInPrediction = TRUE)
```

The `excProb` function uses `pnorm` with means from the `predict.boxcox` layer and standard deviations from `krigeSd`, calculating probabilities of exceeding the Box-Cox transform of 30. The `nuggetInPrediction` argument can be set to `TRUE` to compute probabilities of new observations Y_i exceeding a threshold, with `FALSE` specifying exceedance probabilities for $\lambda(s)$.

The `data` component of `swissFit` provides all the values necessary for further analysis such as conditional simulation or re-estimation of model parameters. This `SpatialPointsDataFrame` contains all covariates $X(s_i)$, observed data Y_i, and residuals $Y_i - X(s_i)\hat{\beta}$ (the latter on the Box-Cox transformed scale if appropriate). Conditional simulation is required for making inference on non-linear functions of the latent process (such as total area above a threshold),

and is advisable when making inference on the latent process with a Box-Cox transformed model. Using a wrapper for the `RFsimulate` function in `RandomFields`, a sample from the conditional distribution $[U|Y]$ is obtained with

```
R> oneSim <- geostatsp::RFsimulate(model = swissFit$param,
+    data = swissFit$data["resid"], err.model = swissFit$param["nugget"],
+    x = raster(extent(swissRain), nrow = 10, ncol = 10))
```

As a final note on the Gaussian geostatistical model, consider the comparison between the **geoR** package (see Diggle and Ribeiro 2006) and **geostatsp** below. The code below estimates the shape and Box-Cox parameters for an isotropic model. Notice the specification of scaling factors for parameters `myscale`, given as a `control` argument.

```
R> swissRain$alt <- raster::extract(swissAltitude, swissRain)
R> library("geoR")
R> swiss2 <- as.geodata(swissRain, data.col = "rain", covar.col = "alt")
R> myscale <- c(range = 1000, shape = 1, boxcox = 1, nugget = 0.1)
R> geoRres <- likfit(swiss2, ini.cov.pars = c(1, 10000), kappa = 0.2,
+    trend = ~alt, lambda = 0.5, fix.lambda = FALSE, fix.nugget = FALSE,
+    fix.kappa = FALSE, lik.method = "REML", message = FALSE,
+    control = list(parscale = myscale[c("range", "nugget", "shape",
+        "boxcox")]))
```

The same model is fit with `lgm` with:

```
R> swissFit2 <- lgm(rain ~ CHE_alt, swissRain, grid = 90,
+    covariates = swissAltitude, shape = 0.2, fixShape = FALSE,
+    boxcox = 0.5, reml = TRUE, fixBoxcox = FALSE, parscale = myscale)
```

The two sets of parameter estimates are comparable, as shown below.

```
          (Intercept)  CHE_alt range nugget boxcox shape variance
geostatsp        6.57 0.000153 54820      0  0.592 0.959     14.4
geoR             6.63 0.000160 52900      0  0.595 1.005     14.5
```

2.2. Generalized linear geostatistical models

The Loaloa data (see Diggle and Ribeiro 2006, 7.6.4) shown in Figure 3 contains the locations of villages where subjects were tested for a tropical disease, with the (binomially distributed) number of positive samples and total sample size being recorded. These data are accessible with `data("loaloa")` in the **geostatsp** package, which contains a `SpatialPointsDataFrame` (named `loaloa`), and raster images for elevation (`elevationLoa`), vegetation index (`eviLoa`) and land type (`ltLoa`). Land type is shown as background values in Figure 3.

The generalized linear geostatistical model from (2) would be suitable for these data with f being a binomial distribution and g being a logit link function. The surface $X(s)$ is multivariate and have values for land type, vegetation index, and elevation. The model can be fit to these data using the `glgm` function in the **geostatsp** package.

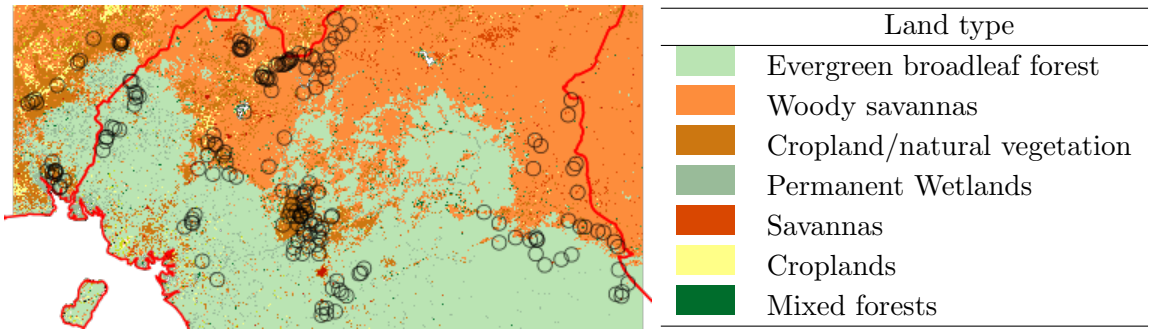


Figure 3: Village locations in the Loaloa dataset (∘) with land type shown as background colors.

As there is more than one covariate, the three rasters containing covariates are grouped together in a list. The elevation covariate is to be fit as a linear effect with a change point at 750m, with two variables `elLow` and `elHigh` being the negative and positive portions of elevation data minus 750. These two rasters are created with

```
R> elevationLoa <- elevationLoa - 750
R> elevLow <- reclassify(elevationLoa, c(0, Inf, 0))
R> elevHigh <- reclassify(elevationLoa, c(-Inf, 0, 0))
```

Land types with a very small number of observations are merged with more populated land types, with: savannas (9) changed to woody savannas (8); wetlands (5) and mixed forests (11) changed to forest (2); and croplands (12) and urban areas (13) changed to crop/natural mosaic (14).

```
R> rcl <- rbind(c(9, 8), c(5, 2), c(11, 2), c(12, 14), c(13, 14))
R> ltLoaRe <- reclassify(ltLoa, rcl)
R> levels(ltLoaRe) = levels(ltLoa)
```

The following code creates the corresponding list of rasters, note that they may have different extents, resolutions or projections.

```
R> covList <- list(elLow = elevLow, elHigh = elevHigh, land = ltLoaRe,
+    evi = eviLoa)
```

The call to `glgm` appears below. As with `lgm`, spatial predictions will be made on a grid as specified by the `grid` argument. It can be specified as a `raster` object though in this case a square grid with 150 cells in the X direction is used. The Markov random field implicitly assumes $U(s)$ takes values of zero outside of the study region, and this effect can be partially negated by adding a buffer (in this case of 50km) around the study region where $U(s)$ will be evaluated but the values in these cells are not returned. Variables listed in the `formula` argument can be contained in either the first argument or in `covariates`. The number of samples taken per village is passed as the `Ntrials` argument, and the `y` variable in the formula is the number of positive samples. The argument `shape` specifies the (fixed) shape parameter κ of the Matérn correlation function.

	Mean	0.025 quantile	0.975 quantile
(Intercept)	$-2.18e+00$	$-3.50e+00$	$-8.62e-01$
factor(land)Woody savannas	$-4.38e-01$	$-7.92e-01$	$-8.92e-02$
factor(land)Cropland/natural	$-2.57e-01$	$-5.91e-01$	$7.20e-02$
evi	$2.63e-04$	$-7.88e-06$	$5.35e-04$
elHigh	$-3.55e-03$	$-4.88e-03$	$-2.18e-03$
elLow	$2.74e-03$	$1.45e-03$	$3.94e-03$
range	$4.22e+04$	$2.67e+04$	$6.46e+04$
sd	$9.88e-01$	$7.80e-01$	$1.25e+00$

Table 2: Posterior expectations and quantiles of model parameters obtained by fitting a generalized linear geostatistical model to the Loaloa dataset using the `glgm` function.

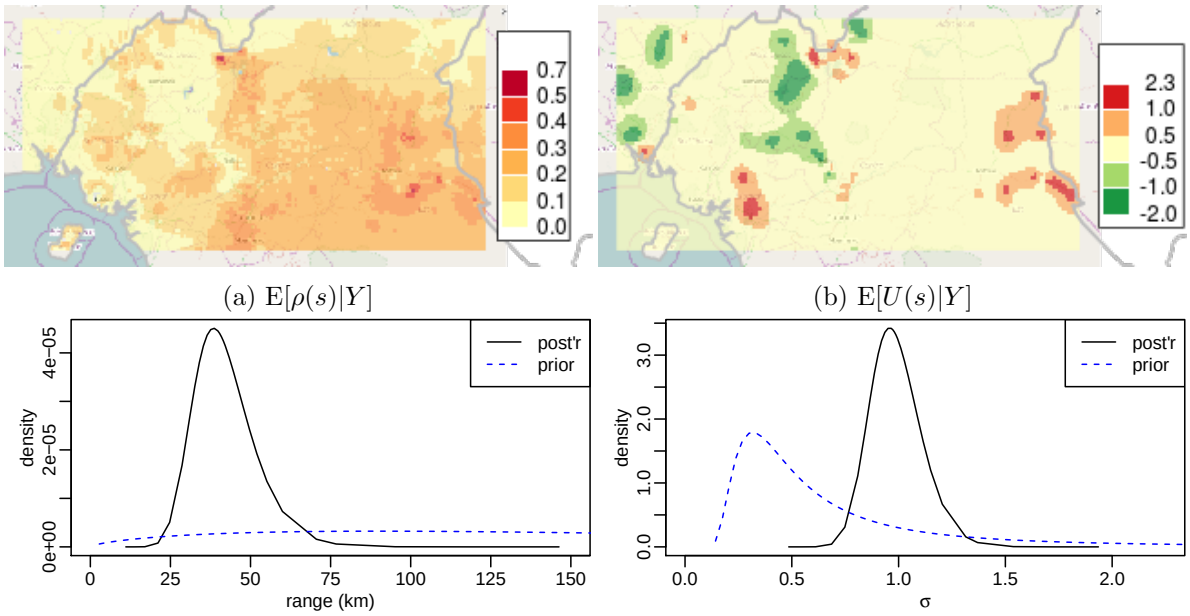
```
R> names(loaloa)

[1] "N"         "y"         "villageID"

R> loaFit = glgm(formula = y ~ factor(land) + evi + elHigh + elLow,
+    data = loaloa, grid = 150, covariates = covList, family = "binomial",
+    Ntrials = loaloa$N, shape = 1, buffer = 50000,
+    priorCI = list(sd = c(0.2, 4), range = c(20000, 5e+05)))
```

Bayesian inference requires prior distributions, and the priors for the spatial covariance parameters are specified by the `priorCI` argument. Prior 95% intervals for σ and ϕ are specified, and `glgm` creates gamma priors for the precision $1/\sigma^2$ and scaled range parameter ϕ/δ (with δ being the cell size) having the 95% intervals specified. Priors other than the gamma are possible (though currently unimplemented in **geostatsp**). The INLA methodology requires priors to be continuous, but are otherwise unrestricted. The **INLA** software specifies that priors are set for log precisions, with prior distributions available including the log-gamma and normal. Incorporating additional priors into **INLA** or **geostatsp** would be relatively straightforward. Priors for the remaining parameters can be specified with `inla` arguments such as `control.fixed = list(prec.intercept = 0.01)`.

The result of the `glgm` function is a list with elements: `inla` for the raw results from **INLA**; `parameters` containing parameter prior and posterior distributions; and `raster` containing the posterior means of the random effects and fitted values. Table 2 contains posterior means and quantiles of the model parameters, taken from the object `loaFit$parameters$summary`.

The component `loaFit$raster` is a `RasterStack` with posterior means, standard deviations, and quantiles for the random effects $U(s)$ and the predicted values on the link scale $g[\lambda(s)]$. The posterior means of $\lambda(s)$ are contained in the layer `"predict.invlogit"`. Figures 4a and b involve the commands `plot(loaFit$raster[["predict.invlogit"]])` and `plot(loaFit$raster[["random.mean"]])`. Figures 4c and d show the prior and posterior distributions of σ and ϕ.

Markov chain Monte Carlo (MCMC) methods are an alternative (and more established) method for fitting spatial models to non-Gaussian data. The **geoRglm** (Christensen and Ribeiro 2002) package provides an excellent set of functions for fitting generalized linear geostatistical models with MCMC, with Matérn correlation functions and geometric anisotropy



(a) E[$\rho(s)|Y$] (b) E[$U(s)|Y$]

(c) Range parameter ϕ (d) Standard deviation σ

Figure 4: Posterior means for spatial surfaces and posterior distributions of model parameters obtained from the `glgm` function.

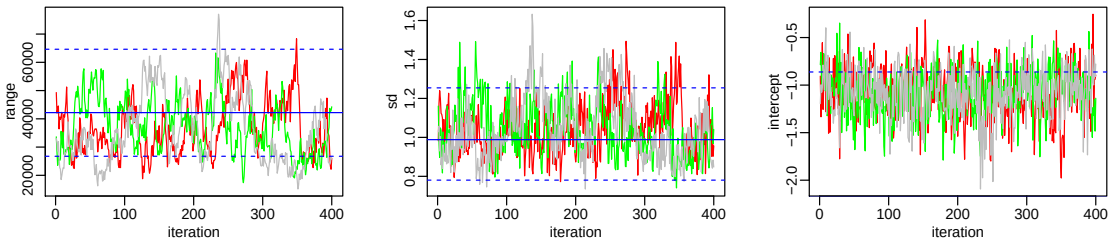
being available. MCMC is more labor intensive and computationally intensive to use than INLA, but is able to produce joint posterior samples. The **geoRglm** package does not use the GMRF approximation, which has advantages and disadvantages. Geometric anisotropy is straightforward withouth the GMRF approximation, and non-integer shape parameters can be used. Readers unfamiliar with MCMC methods are advised to skip ahead to Section 2.3, as the following paragraphs will presuppose a good deal of familiarity with MCMC.

The first step before fitting the Loaloa model using **geoRglm** is to create a new `geodata` object, copying over the values of the covariates extracted from the rasters by `glgm`.

```
R> library("geoRglm")
R> loaNoMissing <- loaloa[as.integer(rownames(loaFit$inla$.args$data)), ]
R> loa2 <- as.geodata(loaNoMissing, data.col = "y")
R> loa2$covariate <- loaFit$inla$.args$data[,
+    c("evi", "elLow", "elHigh", "land")]
R> loa2$covariate$evi <- loa2$covariate$evi - 4000
```

Next, the model and prior distributions are specified. The `model.glm.control` function specifies the fixed effects portion of the model and the Matérn correlation. The prior for the range parameter is taken from the `glgm` output, though notice the difference in parametrisations for the range by factor of $\sqrt{8}$.

```
R> model.10 <- model.glm.control(kappa = 1, cov.model = "matern", trend.d =
+    trend.spatial(~ 1 + elLow + elHigh + evi + factor(land), loa2))
R> phiSeq <- seq(1 * 1000, 100 * 1000, len = 1001)
R> phiValues <- approx(loaFit$param$range$prior, xout = phiSeq * sqrt(8))$y
```



(a) Range (b) Standard deviation (c) Intercept

Figure 5: Trace plots for MCMC samples obtained from fitting the Loaloa data with the **geoRglm** package. Shown are traces for three chains (red, green, grey lines) and as the posterior mean (blue solid line) and 2.5% and 97.5% quantiles obtained from `glgm`.

```
R> phiValues <- phiValues/sum(phiValues)
R> prior.10 <- prior.glm.control(sigmasq.prior = "sc.inv.chisq",
+    df.sigmasq = 1.5, sigmasq = 0.5, phi.prior = phiValues,
+    phi.discrete = phiSeq)
```

Control parameters for the MCMC run are created next. The number of iterations, thinning and burn-in, and scaling parameters are specified by `mcmc.control`,

```
R> mcmc.10 <- mcmc.control(S.scale = 0.004, n.iter = 4e+05,
+    S.start = raster::extract(loaFit$raster[["random.mode"]],
+        loaNoMissing), phi.start = 40 * 1000/sqrt(8), phi.scale = 10,
+        thin = 1000, burn.in = 10000)
R> mcmc.10$S.start[is.na(mcmc.10$S.start)] <- 0
```

The MCMC run is accomplished with the `binom.krige.bayes` function. The code below defines a function to run a single chain, and subsequently runs three chains in parallel. Trace plots for three of the model parameters are show in Figure 5, along with the posterior means and quantiles from INLA. The lack of mixing in the range parameter, despite thinning by a factor of 1000, gives an indication of the perseverance and skill often required when using MCMC for spatial problems.

```
R> oneChain <- function(phiMult) {
+    set.seed(100 * phiMult)
+    mcmc.10$phiStart = mcmc.10$phiStart * phiMult
+    binom.krige.bayes(loa2, units.m = loaNoMissing$N,
+      model = model.10, prior = prior.10, mcmc.input = mcmc.10,
+      output = output.glm.control(messages = FALSE))
+  }
R> library("parallel")
R> test.10 <- mccollect(list(mcparallel(oneChain(0.8)),
+    mcparallel(oneChain(1)), mcparallel(oneChain(1.2))))
```

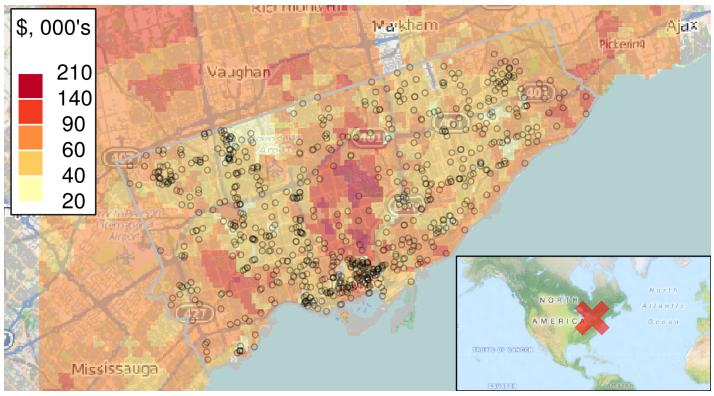


Figure 6: Murder locations in Toronto, Canada (1990–2013) with median household income in 2006 (background colors).

2.3. Log-Gaussian Cox processes

The log-Gaussian Cox process (LGCP) is closely related to the GLGM and is a model suitable for describing the data on murder locations in the city of Toronto, Canada in Figure 6. These data are from the years 1990 to 2013 and appear in the Toronto Star newspaper (`http://www.thestar.com/news/crime/torontohomicidemap.html`). The `murder` dataset in **geostatsp** contains these locations, as well as raster images for median household income (`torontoIncome`), population density (`torontoPdens`), and ambient light (`torontoNight`).

The LGCP (see Møller, Syversveen, and Waagepetersen 1998) is a spatial point process model with the event locations $\{P_i; i = 1 \dots N\}$ being independently distributed conditional on a random log-Gaussian spatial random field $\lambda(s)$. Allowing for a vector of covariates $X(s)$ at location s (in this case light, income, and population density), a LGCP model for the murder locations is

$$\begin{aligned} \{P_i; i = 1 \dots N\} | U(\cdot) &\sim \text{Poisson process}[\lambda(\cdot)] \\ \log[\lambda(s)] &= \mu + X(s)\beta + U(s) \\ \text{cov}[U(s_i), U(s_j)] &= \sigma^2 \text{Matérn}(||s_i - s_j||/\phi; \kappa). \end{aligned}$$

Using a Gaussian Markov random field approximation for $U(s)$, with $U(s)$ being piecewise constant, reduces the inferential problem to modelling the count of points within cells with a Poisson distribution. An improved methodology for fitting LGCP's, using the previously mentioned triangular bases on an irregular lattice, is available in **INLA** (see Rue *et al.* 2013). Currently this method is not implemented in `geostatsp`, and the results below use the GMRF on a square lattice.

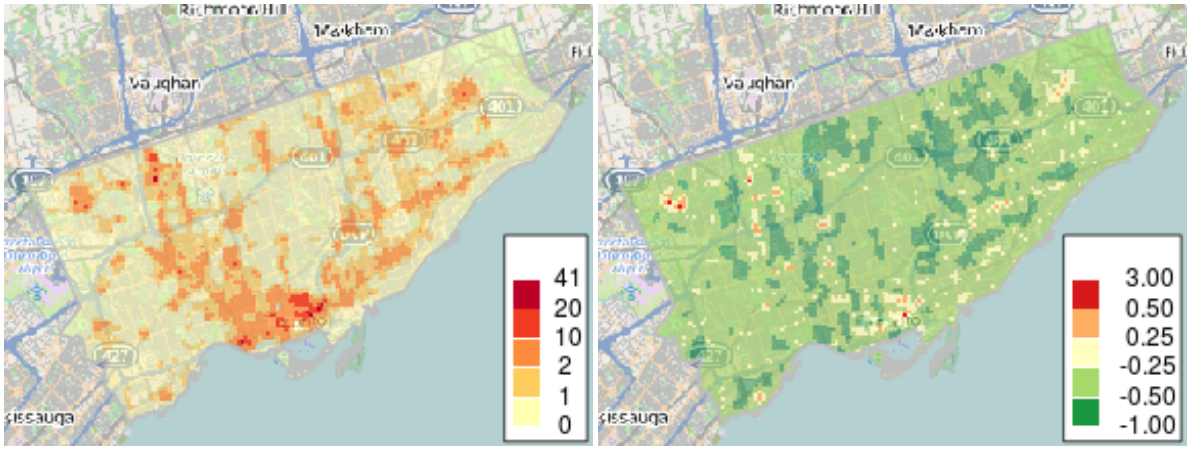
The `lgcp` function in **geostatsp** operates very similarly to `glgm`. A list of covariates is first created, with income, population density, and ambient light transformed so as to make them roughly symmetrically distributed.

```
R> covList <- list(loginc = log(torontoIncome), logpop = log(torontoPdens),
+    loglight = log(torontoNight))
```

Next, the `lgcp` function is called. Here `murder` is a `SpatialPoints` object, and the locations themselves are the response. The model formula is one sided, specifying the covariates. The

	Mean	0.025 quantile	0.975 quantile
`(Intercept)`	−5.07	−9.80	−0.31
`log(income)`	−1.32	−1.73	−0.92
`log(pop'n)`	0.76	0.63	0.90
`log(light)`	0.74	0.51	0.97
`range`	649.55	480.39	860.58
`sd`	0.83	0.71	0.94

Table 3: Posterior means and quantiles of model parameters obtained from fitting the murder data with `lgcp`.



(a) $\mathrm{E}(\lambda(s)|Y)$ in cases/km^2 (b) $\mathrm{E}(U(s)|Y)$

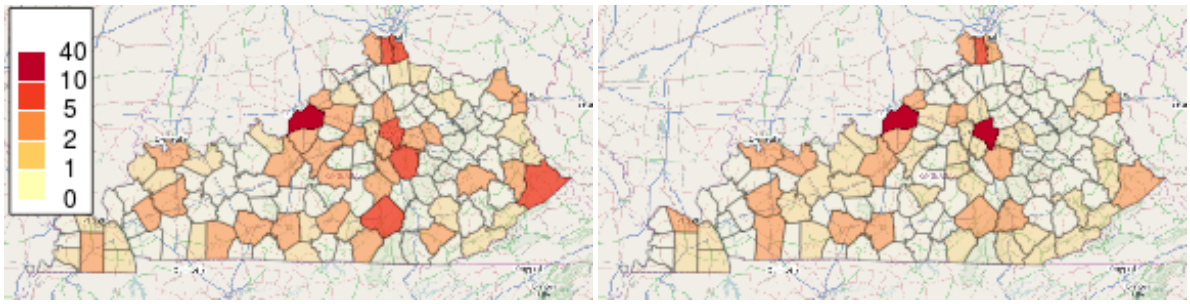
Figure 7: Posterior means of spatial surfaces obtained from fitting the Toronto murder data with the `lgcp` function.

remainder of the arguments are as in `glgm`, and as with `glgm` arguments can be passed directly to INLA.

```
R> murderFit <- lgcp(formula = ~loginc + logpop + loglight, data = murder,
+    grid = 150, covariates = covList, shape = 2, buffer = 4000, priorCI =
+    list(range = c(400, 10000), sd = c(0.02, 2)), border = torontoBorder)
```

Parameter posterior distributions are shown in Table 3. The `raster` component of the results contains posterior distributions for $U(s)$ and $\log[\lambda(s)]$, with the posterior mean of $\lambda(s)$ contained in `murderFit$raster[["predict.exp"]]` and plotted in Figure 7.

As with the generalized linear geostatistical model in the previous section, MCMC is an alternative to using INLA for LGCP's which avoids many of INLA's limitations. The most accessible software for using MCMC with LGCP's is the **lgcp** package described in Taylor, Davies, Rowlingson, and Diggle (2013). As was noted before, MCMC is more computationally intensive and labor intensive than INLA, and requires more than a moderate amount of specialist knowledge to use reliably. One situation where MCMC is required is the case where the point location data are not directly observable. Taylor, Davies, Rowlingson, and Diggle



(a) Observed (b) Expected

Figure 8: Case counts for Larynx cancer in Kentucky.

(2015) show how the **lgcp** package allows for LGCP's to be a latent variable in a hierarchical model, with the LGCP inference nested within a data augmentation algorithm.

2.4. The Besag York and Mollié model

The final model which will be demonstrated is the Besag, York, and Mollié (1991) model (BYM), useful for modelling disease case counts in polygons. Figure 8a shows the number of Larynx cancer cases in each county of Kentucky in a single year, and Figure 8b shows the count that should be expected given the population of each age and sex group in the counties and the US national rates for Larynx cancer. As the case count in a county is often zero or one, the Standardized Mortality Ratio (observed divided by expected) would be expected to be a poor estimator of underlying risk and a spatial random effects model with a Poisson-distributed response variable would be more useful. The BYM model models the case count Y_i of region i, given the expected count E_i and covariates X_i, as

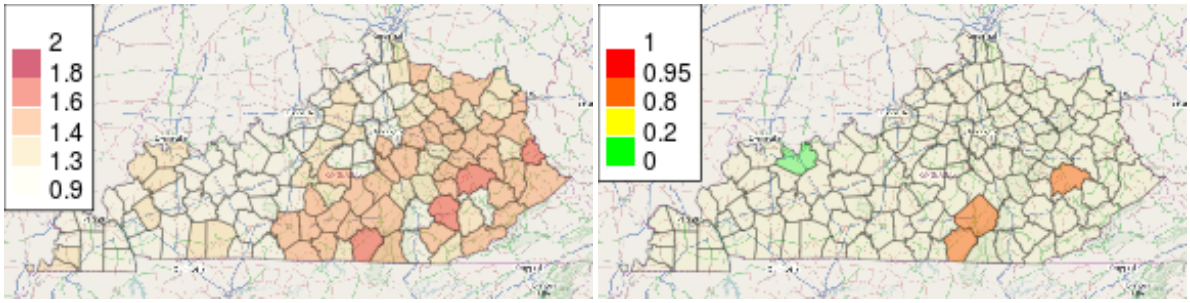
$$\begin{aligned} Y_i \sim & \text{Poisson}(E_i\lambda_i) \\ \log(\lambda_i) = & \mu + X_i\beta + U_i \\ U_i = & W_i + V_i \\ V_i \sim & \text{i.i.d. N}(0, \tau^2) \\ W_i|\{W_j; j \neq i\} \sim & \text{N}(\text{mean}\{W_j; j \sim i\}, \sigma^2/|j \sim i|) \end{aligned}$$

Here W follows a Markov random field model on the irregular lattice of regions, with $j \sim i$ referring to regions i and j being neighbors (sharing a common boundary line). Including the spatially independent term V in the model allows for flexibility in the spatial dependence of U, with τ being larger than σ resulting in a rough surface and σ being larger creating a smoother surface.

The `kentucky` dataset in the **diseasemapping** package contains a `SpatialPolygonsDataFrame` of the counties of Kentucky, and includes the population by age and sex group and the proportion of individuals living in poverty. The `larynx` object is a case file, with one row per individual with larynx cancer in a single year and a variable denoting their county of residence. The `cancerRates` and `getSMR` functions in **diseasemapping** can be used to generate observed and expected counts for each county with the following code. The observed counts and expected counts are shown in Figure 8.

	Mean	0.025 quantile	0.975 quantile
μ	0.11	−0.38	0.61
poverty	0.01	−0.02	0.03
σ	0.19	0.08	0.42
τ	0.20	0.08	0.41

Table 4: Posterior means and 95% credible intervals for model parameters obtained from fitting the Kentucky cancer data with the `bym` function.



(a) $E[\lambda(s)|Y]$ (b) $pr[\lambda(s) > 1.35|Y]$

Figure 9: Posterior means and probabilities obtained from fitting the the Kentucky larynx cancer data with the `bym` function.

```
R> library("diseasemapping")
R> data("kentucky")
R> larynxRates <- cancerRates("USA", year = 1998:2002, site = "Larynx")
R> kentucky <- getSMR(kentucky, larynxRates, larynx, regionCode = "County")
```

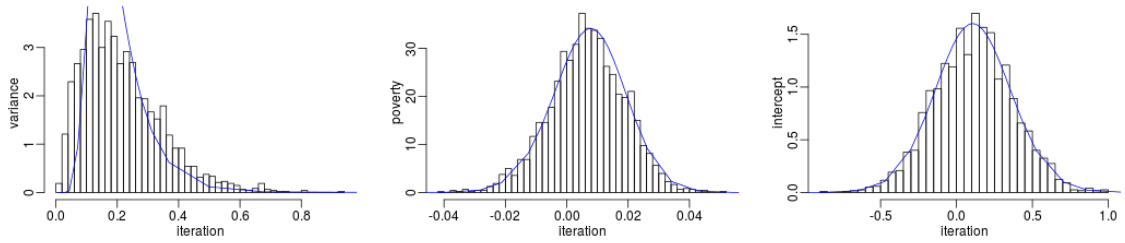
The `bym` function performs Bayesian inference function for the BYM model. It takes as it's arguments the `SpatialPolygonsDataFrame` containing the regional boundaries and variables, as well as the model formula and the prior 95% intervals for σ and τ.

```
R> kBYM <- bym(formula = observed ~ offset(logExpected) + poverty, data =
+    kentucky, priorCI = list(sdSpatial = c(0.1, 5), sdIndep = c(0.1, 5)))
```

The result has a component for the INLA results (`inla`), the parameter posterior distributions (`parameters`), and a `SpatialPolygonsDataFrame` with the spatial results. The posterior means and quantiles of the parameters are given in Table 4, obtained from `kBYM$parameters$summary`.

Figure 9a shows the posterior mean of relative risk λ_i and can be produced with `spplot(kBYM$data, "fitted.exp")`. Figure 9b shows that posterior probability that each county has a cancer rate more than 30% in excess of the US national rates, obtained by numerically integrating the marginal posterior using the and is computed from the marginal posterior distributions `inla` provides. The `excProb` function called below performs the integration using the `trapz` function in the **pracma** package (Borchers 2014).

```
R> kBYM$data$excProb <- excProb(kBYM$inla$marginals.fitted.bym, log(1.3))
```



(a) σ (b) β (c) μ

Figure 10: Histograms of posterior samples (black bars) obtained from fitting the Kentucky cancer data with `OpenBUGS` (black bars), along with posterior densities from the `bym` function (blue lines).

Fitting the BYM model is much less computationally intensive than is the case for LGCP's and the GLGM, and MCMC has a long history of being used with the BYM model (see Gilks, Richardson, and Spiegelhalter 1996). OpenBUGS (Sturtz, Ligges, and Gelman 2005) is a flexible and popular tool for running MCMC, and is able to fit the BYM model. The **glmmbugs** and **R2OpenBUGS** packages (see Brown and Zhou 2010; Sturtz *et al.* 2005) can be used to fit the BYM model with a minimum amount of effort, providing a simple interface between R and OpenBUGS . First, model files are prepared and starting values computed. The `priors` argument creates gamma priors for the standard deviation parameters which are not entirely dissimilar from the priors used by the `bym` function. The `glmmBUGS` function requires priors to be specified for standard deviations, though it would be possible to edit the model file manually to set a posterior for the precision parameter in order to replicate **INLA**'s results.

```
R> library("spdep")
R> kAdjMat <- poly2nb(kentucky, row.names = as.character(kentucky$County))
R> library("glmmBUGS")
R> kBYMbugs <- glmmBUGS(observed + logExpected ~ poverty,
+    data = as.data.frame(kentucky), effects = "County", family = "poisson",
+    spatial = kAdjMat, modelFile = "kentuckyBYM.txt", initFile = "kInit.R",
+    priors = c(SDCountySpatial = "dggamma(5.46,42,0.555)",
+      SDCounty = "dggamma(5.46,42,0.555)"))
```

Second, starting values loaded. The file `kInit.R` contains code for a function to generate random starting values, and users are encouraged to edit this file prior to sourcing it.

```
R> startingValues <- kBYMbugs$startingValues
R> source("kInit.R")
```

Finally, OpenBUGS is run.

```
R> library("R2OpenBUGS")
R> kResult <- bugs(kBYMbugs$ragged, inits = getInits,
+    model.file = "kentuckyBYM.txt", parameters = names(getInits()),
+    n.chain = 3, n.iter = 1000, n.burnin = 200, n.thin = 200)
```

Figure 10 shows marginal posterior distributions for three of the model parameters. The spatial variance parameter has a slightly different posterior from `bym`, due to the differences in the prior distribution.

2.5. A short simulation study

This section aims to illustrate the ease with which spatial simulation studies can be carried out using the **geostatsp** package. Gaussian data is simulated and inference is carried out both with Maximum Likelihood Estimation using the `likfit` function, and Bayesian inference using `glgm`.

Before simulating data, spatial covariates must be created and model parameters specified. The following code defines two simple covariates as sloping north to south and east to west respectively on a square area measuring 10 units across.

```
R> covariates <- brick(xmn = 0, ymn = 0, xmx = 10, ymx = 10,
+    ncols = 200, nrows = 200, nl = 2)
R> values(covariates)[, 1] <- rep(seq(0, 1, len = nrow(covariates)),
+    ncol(covariates))
R> values(covariates)[, 2] <- rep(seq(0, 1, len = nrow(covariates)),
+    rep(nrow(covariates), ncol(covariates)))
R> names(covariates) <- c("cov1", "cov2")
```

Next, a spatial covariance structure is specified with $\sigma = 2$, $\phi = 2.5$, $\tau = 1/2$ and $\kappa = 2$ is specified.

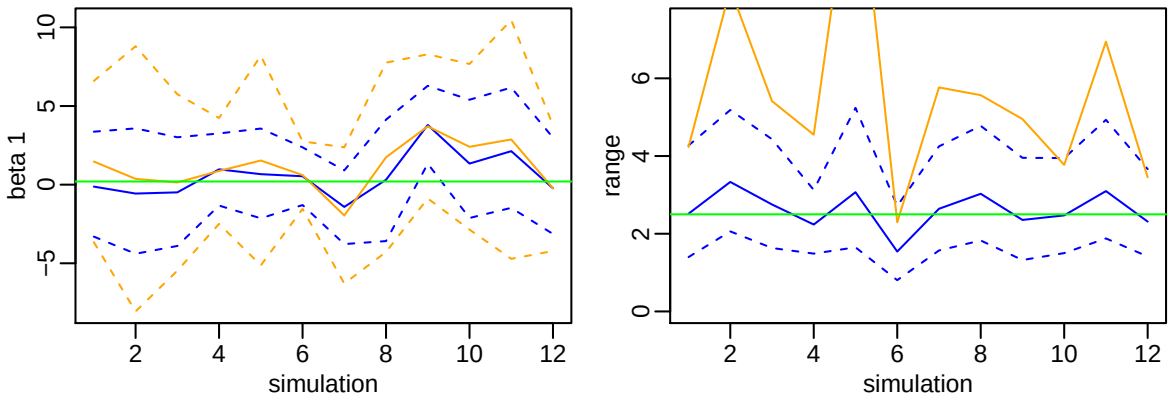
```
R> myModel <- c(intercept = 0.5, variance = 2^2, nugget = 0.5^2,
+    range = 2.5, shape = 2, cov1 = 0.2, cov2 = -0.5)
```

The `RFsimulate` function in **geostatsp**, a wrapper for the function of the same name in **RandomFields** (Schlather *et al.* 2015), is used to simulate a $U(s)$ surface as a raster with the same resolution and dimension as `cov1`. An intercept and the two covariates are added to create a $\lambda(s)$.

Next, points in the study region are simulated at random. Observations at these locations are created by extracting values of $\lambda(s)$ and adding random normal noise with standard deviation 0.5.

```
R> Npoints <- 50
R> myPoints <- SpatialPoints(cbind(runif(Npoints, 0, 10),
+    runif(Npoints, 0, 10)))
R> myPoints <- SpatialPointsDataFrame(myPoints,
+    as.data.frame(extract(covariates, myPoints)))
R> myPoints$fixed <- myModel["intercept"] + drop(as.matrix(data.frame(
+    myPoints))[, names(covariates)] %*% myModel[names(covariates)])
R> myPoints$U <- RFsimulate(myPoints, model = myModel)$sim1
R> myPoints$y <- myPoints$fixed + myPoints$U + rnorm(length(myPoints),
+    0, sqrt(myModel["nugget"]))
```

MLEs are computed with **geostatsp**'s `lgm` function,



(a) β_1 (b) range

Figure 11: True parameters values (—) , Bayesian (—) and maximum likelihood (—) parameter estimates (—) and 95% confidence limits (- - -).

```
R> fitMLE <- lgm(y ~ cov1 + cov2, myPoints, grid = 10,
+    covariates = covariates, shape = 1, fixShape = TRUE)
R> fitMLE$summary["range", "estimate"]

[1] 3.51
```

Bayesian posteriors computed with **geostatsp**'s `glgm`:

```
R> fitBayes <- glgm(formula = y ~ cov1 + cov2, data = myPoints,
+    grid = 30, buffer = 3, covariates = covariates, shape = 1,
+    priorCI = list(range = c(0.15, 10), sd = c(0.1, 10)))
R> fitBayes$parameters$summary["range", "mean"]

[1] 2.47
```

Note the data are simulated with shape parameter $\kappa = 2$ and the fitted model deliberately misspecified $\kappa = 1$. Also, the Markov random field approximation uses a significantly coarser grid than the data are simulated on (30 cells across versus 100).

Figure 11 shows the parameter estimates for 12 simulations. Figure 11a contains the posterior mean and 95% credible interval for the first β coefficient obtained from Bayesian inference, and the MLE and 95% confidence interval obtained from Frequentist inference. Notice the estimates and intervals are nearly identical. Figure 11b shows the Bayesian posterior mean and credible interval for the range parameter ϕ along with the MLE. Confidence intervals for the range are not produced by `lgm`. The MLE and the posterior mean differ, sometimes substantially, but tend to identify the true value despite κ being misspecified.

3. Discussion

The **geostatsp** and **diseasemapping** packages remove much of the drudgery involved in fitting spatial random effects models with **INLA** (Rue *et al.* 2013) or **geoR** (Diggle and Ribeiro

2006). The creation of the lattice for the Markov Random Field in **INLA** in particular is time consuming to code, and translation of **INLA** results to maps and interpretable parameters is not always straightforward. The use of `Raster` objects for covariates and default values able to accommodate UTM spatial coordinates (with values in the thousands or hundreds of thousands) do away with the need to modify and reformat data prior to its analysis.

These packages provide a mechanism for fitting geostatistical models using R spatial data types: `SpatialPointsDataFrame`s, `SpatialPolygonsDataFrame`s, and `Raster`s. The advantage of working with these data types in place of matrices and vectors of coordinates is two-fold. First, data from various sources can be easily downloaded and included in these analyses. NASA provides a wide variety of satellite data, including elevation and vegetation indices, which can be loaded into R using the **raster** package. Census data are often available as Shapefiles which can be read using **rgdal** . Much of the geographic data from sources such as these have coordinates in a longitude-latitude projection, and geostatistical analyses involving Euclidean distances require coordinates on a metre-based (or UTM) projection. Converting coordinates is easily accomplished with the `spTransform` and `projectRaster` functions, and many additional functions in the **sp** and **raster** packages are available for data manipulation and processing.

A second advantage accruing from the use of R spatial objects is the ease with which results can be exported to GIS software or plotted with background map images in R. The maps presented in this paper have required projecting the result to a longitude-latitude coordinate reference system with the `projectRaster` function, downloading background layers from Openstreetmap.org with the **mapmisc** package, and obtaining city names and locations with **geonames**. While it is possible to improve on the maps presented here using GIS software, code for R generated maps can be incorporated in **Sweave** and **knitr** scripts thereby allowing any manual GIS map creation to be reserved for final drafts of documents.

The **geostastp** package could be improved by incorporating several more of the facilities in **INLA** and work towards this is ongoing. Replacing the grid of square cells in the MRF approximation with the irregular lattices in Lindgren *et al.* (2011) would increase speed and accuracy of the approximation, though spatial predictions could still be made on rasters and the inner workings of the approximation could remain hidden from the user. The **INLA** software allows for non-parametric effects of covariates, it would be possible to specify non-parametric effects in `glgm` though they would not as of yet be included when making spatial predictions. Also, **INLA** can fit a variety of spatio-temporal models and a simple user-friendly interface to fitting spatio-temporal data would certainly be possible.

Acknowledgments

Background maps are from Openstreetmap.org. The author is funded by the Natural Sciences and Engineering Research Council of Canada.

References

Besag J, York J, Mollié A (1991). "Bayesian Image Restoration, with Two Applications In Spatial Statistics." *Annals of the Institute of Statistical Mathematics*, **43**(1), 1–20.

Bivand R, Keitt T, Rowlingson B (2014). ***rgdal**: Bindings for the Geospatial Data Abstraction Library*. R package version 0.9-1, URL http://CRAN.R-project.org/package=rgdal.

Bivand RS (2014). ***spdep**: Spatial Dependence: Weighting Schemes, Statistics and Models*. R package version 0.5-77, URL http://CRAN.R-project.org/package=spdep.

Bivand RS, Pebesma E, Gómez-Rubio V (2013). *Applied Spatial Data Analysis with R*. 2nd edition. Springer-Verlag, New York. URL http://www.asdar-book.org/.

Borchers HW (2014). ***pracma**: Practical Numerical Math Functions*. R package version 1.6.1, URL http://CRAN.R-project.org/package=pracma.

Brown PE, Zhou L (2010). "MCMC for Generalized Linear Mixed Models with **glmmBUGS**." *The R Journal*, **2**(1), 13–17. URL http://journal.R-project.org/.

Christensen OF, Ribeiro PJ (2002). "**geoRglm**: A Package for Generalised Linear Spatial Models." *R News*, **2**(2), 26–28. URL http://CRAN.R-project.org/doc/Rnews.

Cressie N (1993). *Statistics For Spatial Data*. John Wiley & Sons.

Diggle PJ, Moyeed RA, Tawn JA (1998). "Model-Based Geostatistics." *Applied Statistics*, **47**, 299–350.

Diggle PJ, Ribeiro PJ (2006). *Model-Based Geostatistics*. Springer-Verlag, New York.

Gilks WR, Richardson S, Spiegelhalter DJ (1996). *Markov Chain Monte Carlo in Practice*. Chapman Hall/CRC, New York.

Hijmans RJ (2014). ***raster**: Geographic Data Analysis and Modeling*. R package version 2.2-31, URL http://CRAN.R-project.org/package=raster.

Jiang H, Brown PE, Rue H, Shimakura S (2014). "Geostatistical Survival Models for Environmental Risk Assessment With Large Retrospective Cohorts." *Journal of the Royal Statistical Society A*, **177**(3), 679–695.

Li Y, Brown PE, Rue H, al Maini M, Fortin P (2012). "Spatial Modelling of Lupus Incidence Over 40 Years With Changes In Census Areas." *Journal of the Royal Statistical Society C*, **61**, 99–115.

Lindgren F, Rue H, Lindström J (2011). "An Explicit Link between Gaussian Fields and Gaussian Markov Random Fields: The Stochastic Partial Differential Equation Approach." *Journal of the Royal Statistical Society B*, **73**(4), 423–498.

Matheron G (1962). *Traité de Géostatistique Appliquée*. Editions Technip.

Møller J, Syversveen AR, Waagepetersen RP (1998). "Log Gaussian Cox Processes." *Scandinavian Journal of Statistics*, **25**(3), 451–482.

R Core Team (2014). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

Ribeiro PJ, Diggle PJ (2001). "**geoR**: A Package For Geostatistical Analysis." *R News*, **1**(2), 15–18. URL http://CRAN.R-project.org/doc/Rnews/.

Rue H, Martino S, Chopin N (2009). "Approximate Bayesian Inference for Latent Gaussian Models by Using Integrated Nested Laplace Approximations." *Journal of the Royal Statistical Society B*, **71**(2), 319–392.

Rue H, Martino S, Lindgren F, Simpson D, Riebler A (2013). ***INLA**: Functions Which Allow to Perform Full Bayesian Analysis of Latent Gaussian Models Using Integrated Nested Laplace Approximation*. R package, URL http://R-Inla.org.

Schlather M, Malinowski A, Menck PJ, Oesting M, Strokorb K (2015). "Analysis, Simulation and Prediction of Multivariate Random Fields with Package **RandomFields**." *Journal of Statistical Software*, **63**(8), 1–25. URL http://www.jstatsoft.org/v63/i08/.

Stein ML (1999). *Interpolation of Spatial Data: Some Theory for Kriging*. Springer-Verlag, New York.

Sturtz S, Ligges U, Gelman A (2005). "**R2WinBUGS**: A Package for Running WinBUGS from R." *Journal of Statistical Software*, **12**(3), 1–16. URL http://www.jstatsoft.org/v12/i03/.

Taylor BM, Davies TM, Rowlingson BS, Diggle PJ (2013). "**lgcp**: An R Package for Inference with Spatial and Spatio-Temporal Log-Gaussian Cox Processes." *Journal of Statistical Software*, **52**(4), 1–40. URL http://www.jstatsoft.org/v52/i04/.

Taylor BM, Davies TM, Rowlingson BS, Diggle PJ (2015). "Bayesian Inference and Data Augmentation Schemes for Spatial, Spatiotemporal and Multivariate Log-Gaussian Cox Processes in R." *Journal of Statistical Software*, **63**(7), 1–48. URL http://www.jstatsoft.org/v63/i07/.

Wikipedia (2013). "Matérn Covariance Function — Wikipedia, The Free Encyclopedia." Accessed 2014-07-16, URL http://en.wikipedia.org/wiki/MatÃl'rn_covariance_function.

A. Matérn correlation

There are several parametrisations of the Matérn correlation function, and the `range` parameter in `lgm` and `glgm` corresponds to ϕ in

$$\rho(h;\phi,\kappa) = \frac{1}{\Gamma(\kappa)2^{\kappa-1}} \left(\frac{\sqrt{8\kappa}||h||}{\phi} \right)^{\kappa} K_{\kappa} \left(\sqrt{8\kappa}||h||/\phi \right).$$

$\Gamma(\cdot)$ is a gamma function and K_{κ} is a modified Bessel function of the second kind of order κ. Figure 12 shows plots of the Matérn for various values of κ and all with $\phi = 1$. Notice that, with the possible exception of $\kappa = 0.1$, the correlations intersect (more or less) at $||h|| = 1$. A not inaccurate interpretation of the range parameter ϕ in this parametrization is it is the distance beyond which correlation is both fairly small (< 0.14), and decaying fairly slowly regardless of the shape parameter κ.

The **geostatsp** package has a `matern` function which implements the parametrization above, though it may be helpful to consider the function below. Figure 12 is produced with this code.

```
R> mymatern <- function(u, phi, kappa) {
+    uscale <- sqrt(8 * kappa) * u/phi
+    res <- (1/(gamma(kappa) * 2^(kappa - 1))) * uscale^kappa *
+      besselK(uscale, kappa)
+    res[u == 0] <- 1
+    res
+  }
```

Wikipedia (2013) and the 'matern' model in the **RandomFields** package define the range parameter as $\phi_1 = \phi/2$. Diggle and Ribeiro (2006), the **geoR** package, and the `whittle` model in **RandomFields** have a range parameter $\phi_2 = \phi/\sqrt{8\kappa}$. It is also common to define the Matérn with a scale parameter in place of the range, with the scale parameter being $\alpha = 1/\phi_2$. Lindgren *et al.* (2011) use either the scale α or the range ϕ. The `Range` parameter produced by `inla` is $\phi\delta$, with δ being the length of the sides of the grid cells, as confirmed below.

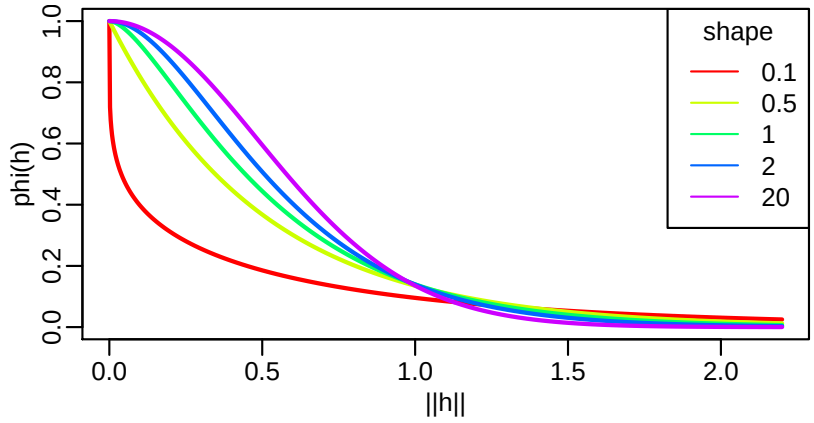


Figure 12: Matérn correlation functions with $\phi = 1$ and various values of the shape parameter κ

```
R> c(loaFit$inla$summary.hyperpar["Range for space", "mode"] *
+    xres(loaFit$raster), loaFit$par$summary["range", "mode"])

[1] 38489 38489
```

Affiliation:

Patrick Brown
Prevention and Cancer Control
Cancer Care Ontario
620 University Avenue
Toronto, ON, M5G 2L7, Canada
E-mail: patrick.brown@utoronto.ca
URL: http://pbrown.ca/

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 63, Issue 12 *Submitted:* 2013-06-06
January 2015 *Accepted:* 2014-10-22