spacious
An R package for analysis of large geostatistical spatial datasets

Ryan J. Parker[1], **Jo Eidsvik**[2], **Brian J. Reich**[1], **Benjamin Shaby**[3], and **Jarad Niemi**[4]

February 23, 2014

Spacious in an R package to estimate spatial covariance parameters using large geostatistical spatial datasets and to use these estimates for spatial prediction. To achieve efficient computation, spacious uses the parallelizable block composite likelihood approach of Eidsvik et al. (2013), which we briefly review in Section 2. In addition to the efficient maximum composite likelihood estimation with pthreads, spacious also provides GPU acceleration for maximum likelihood estimation of the full spatial likelihood. This GPU acceleration for the full likelihood, however, is currently limited by the amount of memory available on the GPU.

In this manual we begin in Section 1 showing how to install spacious using the default or parallel computing options with pthreads or CUDA. After a review of the block composite likelihood in Section 2, we then show how to implement this approach for parameter estimation in Section 3 and spatial prediction in Section 4. We end with Section 5 showing how spacious uses parallel computing to estimate models for moderate and large size data sets.

1 Installing the spacious package

The spacious package can be downloaded from CRAN and installed using the command:

```
> install.packages("spacious")
```

1.1 Enabling pthreads for parallelization with threads

The block composite likelihood in Section 2 operates on independent pairs of blocks, and thus computations for each pair of blocks can be run in parallel. Running these computations in parallel is supported with pthreads. The installation procedure searches for pthreads support by default. If pthreads is not enabled by default, you will need to add the location of pthreads to your LD_LIBRARY_PATH and run the command:

```
> install.packages("spacious", type="source")
```

The pthreads library is not natively supported by Windows and has not been tested for use with spacious.

[1]North Carolina State University
[2]Norwegian University of Science and Technology
[3]Penn State University
[4]Iowa State University

1.2 Enabling CUDA for GPU acceleration

Currently, `spacious` can accelerate matrix operations for maximum likelihood estimation of full models using `CUDA` and `CUBLAS`. To enable `CUDA` support, run the command:

```
> install.packages("spacious", type="source", configure.args="--with-cuda")
```

By default the installation searches for `CUDA_HOME` in `/usr/local/cuda`. If another location should be used, change `configure.args` to `--with-cuda=DIR`, where `DIR` specifies the location of `CUDA_HOME`. Support for `CUDA` has been tested with the 5.0 release.

2 Review of the block composite likelihood approach

We begin with the standard model for $\mathbf{Y} = [Y(\mathbf{s}_1), ..., Y(\mathbf{s}_n)]^T$, the vector of observations at spatial locations $\mathbf{s}_1, ..., \mathbf{s}_n \in \mathcal{D}$, where $\mathcal{D}$ is the spatial domain of interest. We assume Y is a Gaussian process with mean $\mathrm{E}[Y(\mathbf{s}_i)] = \mathbf{X}_i^T\boldsymbol{\beta}$ and

$$\mathrm{Cov}[Y(\mathbf{s}_i), Y(\mathbf{s}_j)] = \tau^2 I(\mathbf{s}_i = \mathbf{s}_j) + \sigma^2\rho(||\mathbf{s}_i - \mathbf{s}_j||; \phi, \nu),$$

where τ^2 is the nugget variance, σ^2 is the partial sill, and ρ is the Matérn correlation function with spatial range ϕ and smoothness ν. `Spacious` can also fit the exponential covariance with smoothness $\nu = 0.5$, and thus $\rho(||\mathbf{s}_i - \mathbf{s}_j||; \phi, 0.5) = \exp(-||\mathbf{s}_i - \mathbf{s}_j||/\phi)$.

The full likelihood is proportional to

$$|\boldsymbol{\Sigma}(\boldsymbol{\theta})|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})^T\boldsymbol{\Sigma}(\boldsymbol{\theta})^{-1}(\mathbf{Y} - \mathbf{X}\boldsymbol{\beta})\right],$$

where $\boldsymbol{\Sigma}(\boldsymbol{\theta})$ is the $n \times n$ covariance matrix involving spatial covariance parameters $\boldsymbol{\theta} = (\tau^2, \sigma^2, \phi, \nu)$ and $\mathbf{X}$ is the $n \times p$ design matrix comprised of covariate vectors $\mathbf{X}_i$. The bottleneck in computing the maximum likelihood estimate is clear: evaluating the likelihood requires $\mathcal{O}(n^3)$ matrix operations.

To avoid working with large matrices while preserving the local spatial structure, Eidsvik et al. (2013) partition $\mathcal{D}$ into M blocks/subregions $\mathcal{B}_1, ..., \mathcal{B}_M$, with $\mathbf{Y}_j$ denoting the vector of observations in $\mathcal{B}_j$. The block composite likelihood is simply the product of joint likelihoods for pairs of adjacent blocks

$$\prod_{j \sim k} |\boldsymbol{\Sigma}_{jk}(\boldsymbol{\theta})|^{-1/2} \exp\left[-\frac{1}{2}(\mathbf{Y}_{jk} - \mathbf{X}_{jk}\boldsymbol{\beta})^T\boldsymbol{\Sigma}_{jk}(\boldsymbol{\theta})^{-1}(\mathbf{Y}_{jk} - \mathbf{X}_{jk}\boldsymbol{\beta})\right],$$

where $\mathbf{Y}_{jk} = (\mathbf{Y}_j^T, \mathbf{Y}_k^T)^T$, $\mathbf{X}_{jk}$ and $\boldsymbol{\Sigma}_{jk}(\boldsymbol{\theta})$ are the corresponding design and covariance matrices, and $j \sim k$ indicates that regions $\mathcal{B}_j$ and $\mathcal{B}_k$ are adjacent. Assuming that the number of observations in each block is fixed and the number of pairs of adjacent blocks increases linearly as the sample size increases, then the computational complexity for evaluating the likelihood is $\mathcal{O}(n)$, a dramatic improvement over the $\mathcal{O}(n^3)$ evaluation of the full likelihood.

3 Parameter estimation

We illustrate `spacious` using the temperature anomaly data included in the `spacious` package. These data are described in detail by Klein Tank et al. (2002) and can be downloaded from `http://www.ecad.eu`. The object `anom.2011` contains $n = 1,375$ observations observed at locations of longitude `lon` and latitude `lat` with covariate elevation `elev`. Shown in Figure 1, the code below plots the response, `anom`, which is 2011 annual mean temperature anomaly (computed with respect to the 1961-1990 average) at the $n \times 2$ matrix of spatial coordinates, `S`.

```
> library(spacious)
> data(anom.2011)
> S <- cbind(anom.2011$lon, anom.2011$lat)
>
> library(ggmap)
> euro <- get_map(location=colMeans(S), zoom=4, maptype="satellite")
>
> ggmap(euro) +
      geom_point(aes(lon, lat, color=anom), shape=15, size=1.5, data=anom.2011) +
      scale_colour_gradient(low = "green", high="red")
```

The dataset also includes regularly-spaced prediction locations and elevations in the `anom.pred.grid` object.

3.1 The basic `spacious` call

The main call is

```
spacious(formula, data, S, cov = "exp", cov.inits, B, neighbors,
         fixed = list(smoothness = 0.5),
         blocks = list(type = "cluster"),
         verbose = FALSE, tol = 1e-8, maxIter = 100,
         nthreads = 1, gpu = FALSE, engine = "C")
```

The response and mean trend are specified by `formula`, which follows the usual `response` $\sim$ `sum of predictors` notation. The optional data frame `data` can be used to specify the variables in `formula`. The fit with longitude, latitude, and elevation as predictors with default exponential covariance and blocking options is

```
> fit <- spacious(anom ~ lon + lat + elev, S=S, data=anom.2011)
> summary(fit)
Coefficients:
   Estimate Std Err P-value
b0  1.84659 0.19667    0.00
b1  0.00034 0.00200    0.87
b2 -0.00567 0.00378    0.13
b3 -0.00001 0.00003    0.73
Spatial parameters
             Estimate Std Err
Nugget          0.319   0.008
Partial Sill    0.055   0.010
Range           3.072   0.845
```

Figure 2 gives the convergence plot that results from `plot(fit)`. This plots the value of each spatial covariance parameter at each iteration of the optimization routine. When the model fit converges, these trace plots will plateau for all covariance parameters. Model convergence can be assessed with the variable `fit$convergence`. When `fit$convergence` is `FALSE`, this plot may help identify which parameter(s) are not converging.

Figure 1: Temperature anomaly data in Celcius.

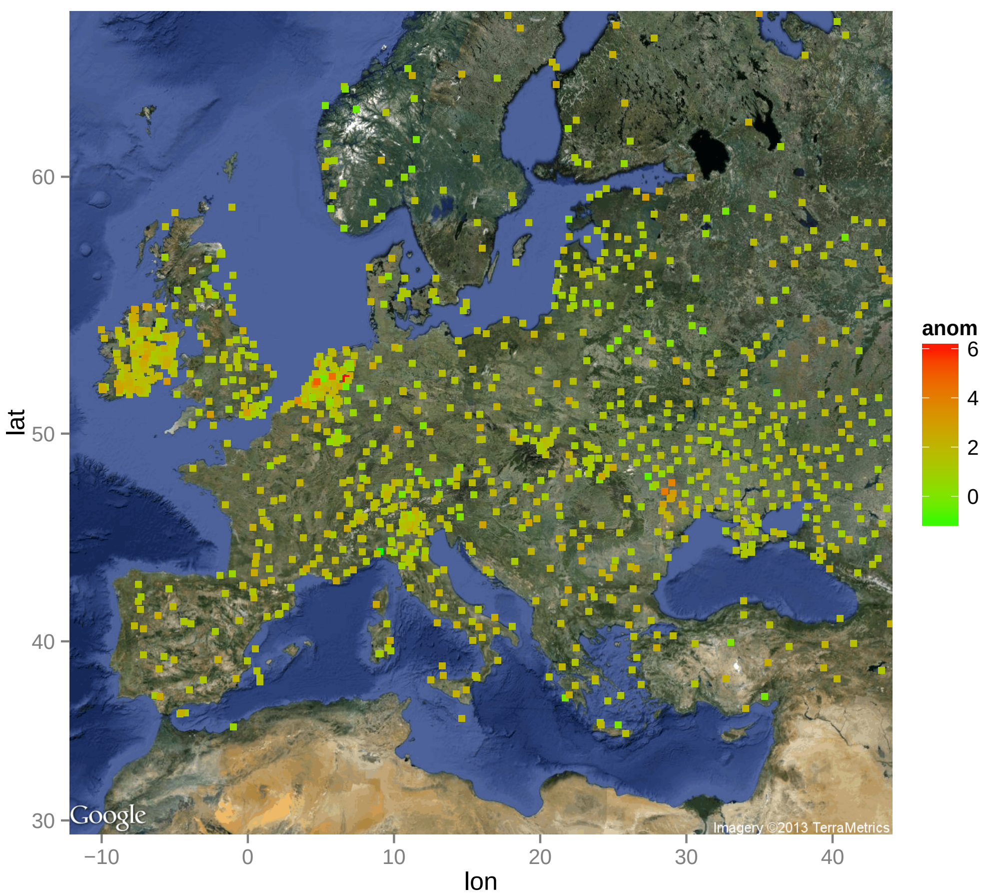
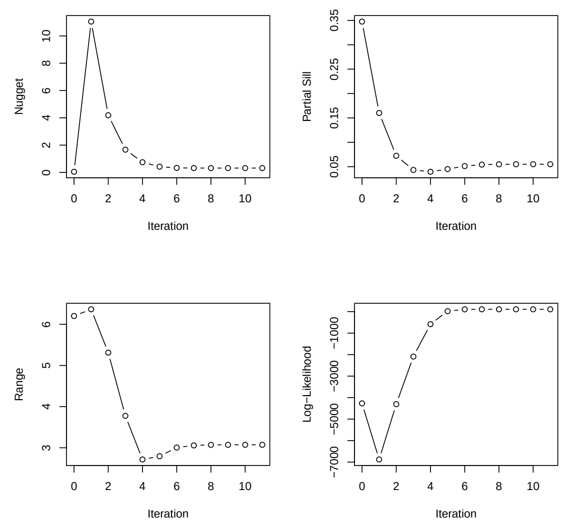


Figure 2: Convergence plot for the temperature anomaly data.



3.2 Specifying the spatial covariance function

The default spatial covariance function is exponential covariance with the nugget, partial sill, and range to be estimated. It is also possible to fix (rather than estimate) some of the parameters using the `fixed` option, which is a list with three possible entries: `nugget`, `psill`, and `range`. For example, `fixed = list(nugget=0)` specifies a spatial covariance with no nugget ($\tau^2 = 0$). Removing the nugget can lead to computational problems due to singular covariance matrices and should be used with caution. Here is the fit with $\phi = 5$,

```
> fit <- spacious(anom ~ lon + lat + elev, S=S, data=anom.2011,
                  fixed=list(range=0.20))
> summary(fit)
Coefficients:
    Estimate  Std Err P-value
b0  1.823724 0.231828 3.6e-15
b1  0.000493 0.002418 8.4e-01
b2 -0.005209 0.004436 2.4e-01
b3 -0.000014 0.000029 6.3e-01
Spatial parameters
             Estimate Std Err
Nugget          0.325       0
Partial Sill    0.053       0
Range           5.000       0
```

`Spacious` can also accommodate the Matérn covariance using the `cov="matern"` option. In this case the smoothness parameter ν must be specified as fixed. For example, the code below fits a Matérn covariance with $\nu = 1$,

```
> fit <- spacious(anom ~ lon + lat + elev, S=S, data=anom.2011,
```

```
                     cov="matern", fixed=list(smoothness=1))
> summary(fit)
Coefficients:
    Estimate  Std Err P-value
b0  1.845695 0.189774    0.00
b1  0.000307 0.001922    0.87
b2 -0.005639 0.003645    0.12
b3 -0.000011 0.000029    0.70
Spatial parameters
             Estimate Std Err
Nugget          0.325       0
Partial Sill    0.048       0
Range           2.067       0
Smoothness      1.000       0
```

Initial values for the spatial covariance parameters can be specified using the `cov.inits` option, which is a list of the same form as `fixed`. The defaults are

```
cov.inits = list(nugget = 0.8*var(Y),
                 psill = 0.2*var(Y),
                 range = quantile(dist(S),0.1))
```

It is highly recommended to try several initial values to ensure they all lead to the same solution.

3.3 Specifying the blocking structure

Selecting the blocking structure that forms the block composite likelihood is a crucial step in applying `spacious`. Selecting one or two blocks gives the statistically optimal, but computationally inefficient, full likelihood. Selecting a large number of blocks improves computational speed with a trade-off in statistically efficiency. The blocking structure is determined by the `blocks` argument, which is specified as a list with two elements: `type` and `nblocks`. The `type` argument can be either `"cluster"` or `"regular"`, and `nblocks` gives the number of blocks (M). If `type = "cluster"`, then the block groups are determined by k-means clustering of the observation locations, `S`. If `type = "regular"`, then blocks are taken to be a regular $\sqrt{M} \times \sqrt{M}$ rectangular grid covering the range of `S`. Clearly, if `type = "regular"`, then `nblocks` should be a square number. The default is cluster blocks with approximately $n/50$ blocks.

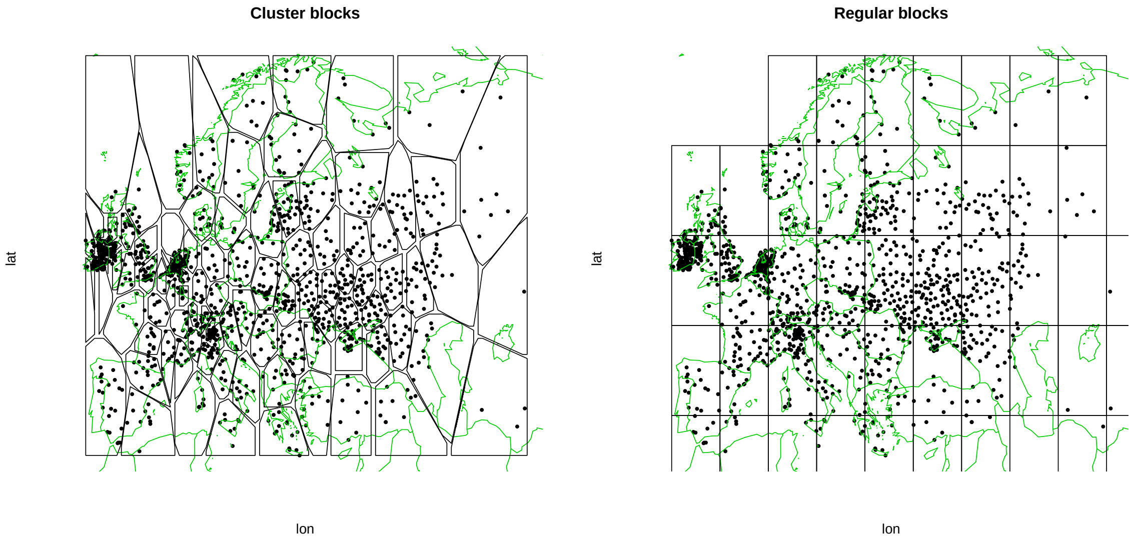
The following code fits the composite likelihood with $M = 100$ blocks using both clustered and regular blocks:

```
> library(maps)
> fit_c <- spacious(anom ~ lon + lat + elev, S=S, data=anom.2011,
                    blocks=list(type="cluster", nblocks=100))
> fit_r <- spacious(anom ~ lon + lat + elev, S=S, data=anom.2011,
                    blocks=list(type="regular", nblocks=100))

> plot(S, cex=0.5, pch=19, axes=FALSE, main="Cluster blocks")
> map("world", add=TRUE, col=3)
> plot(fit_c$grid, add=TRUE)

```

Figure 3: Blocking structures for the temperature data.




```
> plot(S, cex=0.5, pch=19, axes=FALSE, main="Regular blocks")
> map("world", add=TRUE, col=3)
> plot(fit_r$grid, add=TRUE)
```

The output `B` gives the cluster label assigned to each observation, and is plotted in Figure 3. For these data, the regular blocks have many observations in two blocks in Ireland and the Netherlands, which slows computation.

Alternatively, it is possible to specify the blocking structure directly using the `B` and `neighbors` arguments. The input `B` is a vector of n block numbers (e.g., `B[4] = 3` implies that the fourth observation is from block number 3), and `neighbors` is an $M \times 2$ matrix specifying the adjacency structure of the blocks (e.g., if a row of `neighbors` equals `c(4,5)` then blocks 4 and 5 are neighbors).

4 Spatial prediction

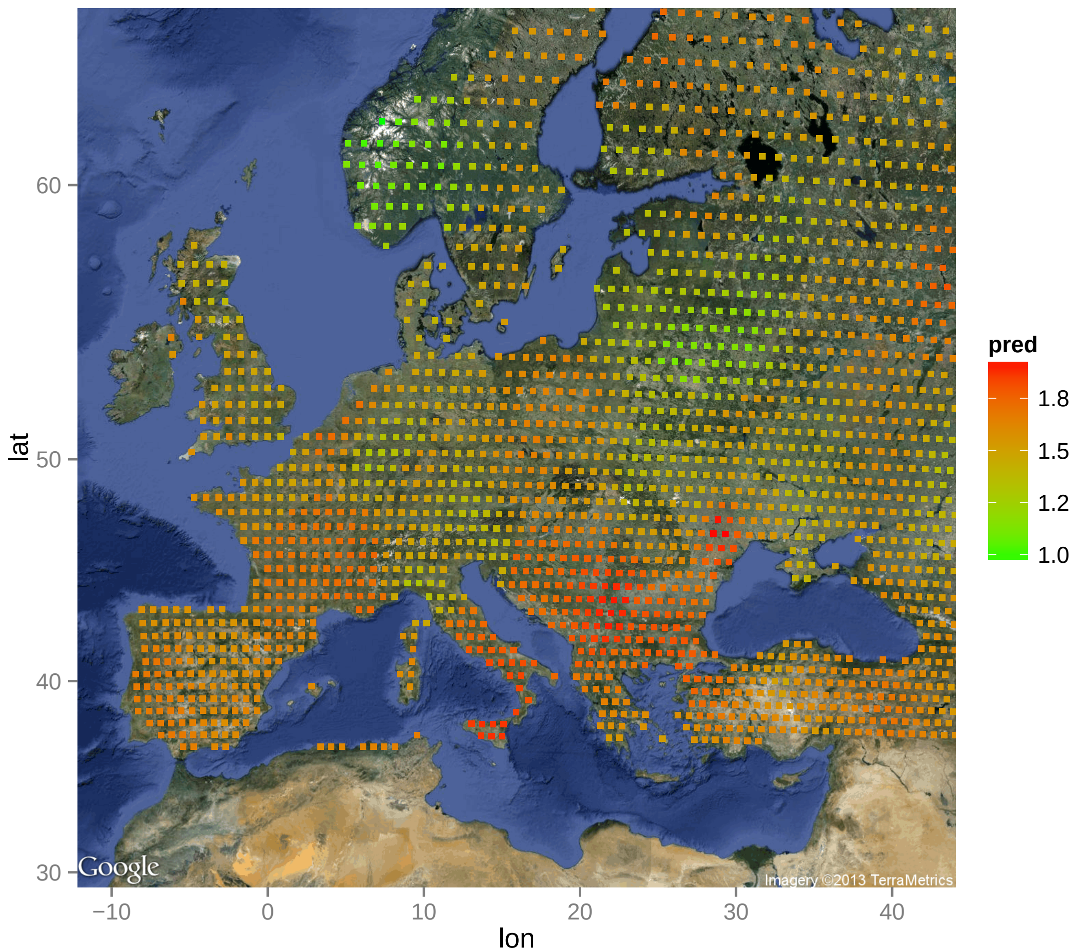
Spatial prediction is carried out using the `predict` function. The required arguments are the `spacious` fit, the prediction locations, `newS`, and any covariates, `newdata`. The code below generates fitted values at locations `Sp`, and plots the fitted values using ggmaps (Figure 4).

```
> fit  <- spacious(anom ~ lon + lat + elev, S=S, data=anom.2011)
> Sp   <- cbind(anom.pred.grid$lon, anom.pred.grid$lat)
> pred <- predict(fit, newdata=anom.pred.grid, newS=Sp)$y

> ggmap(euro) +
      geom_point(aes(lon, lat, color=pred), shape=15, size=1.25,
                 data=data.frame(anom.pred.grid, pred=pred)) +
      scale_colour_gradient(low = "green", high="red")
```

By default, `predict` uses block prediction (Eidsvik et al., 2013) with blocks specified by the `spacious` object. Alternatively, the `opts` argument can be used to perform local prediction where

Figure 4: Spatial predictions for the temperature (Celcius) anomaly data.



only the closest m locations to each prediction point are used for prediction (so that taking $m = n$ gives the usual Kriging predictions). The code below uses the nearest $m = 25$ observations for prediction, which gives similar results to the block predictions.

```
> pred.local <- predict(fit, newdata=anom.pred.grid, newS=Sp,
                        opts=list(type="local", num=25))$y
> pred[1:5]
[1] 1.663 1.651 1.616 1.610 1.614
> pred.local[1:5]
[1] 1.660 1.648 1.605 1.595 1.599
```

Prediction intervals can also be obtained by adding the argument `interval="prediction"` and optionally specifying confidence `level` (default is 0.95). For example, the 95% prediction intervals for the first five prediction locations are

```
> predict(fit, newdata=anom.pred.grid[1:5,], newS=Sp[1:5,], interval="prediction")
      y     sd    lwr   upr
1 1.663 0.5953 0.4958 2.829
2 1.651 0.5942 0.4860 2.815
3 1.616 0.5975 0.4445 2.787
4 1.610 0.6000 0.4342 2.786
5 1.614 0.6077 0.4234 2.805
```

5 Estimation with parallel computing for large data sets

Acknowledgments

References

Eidsvik, J., Shaby, B. A., Reich, B. J., Wheeler, M., and Niemi, J. (2013), "Estimation and prediction in spatial models with block composite likelihoods," *Journal of Computational and Graphical Statistics*, in press.

Klein Tank, A., Wijngaard, J., Können, G., Böhm, R., Demarée, G., Gocheva, A., Mileta, M., Pashiardis, S., Hejkrlik, L., Kern-Hansen, C., et al. (2002), "Daily dataset of 20th-century surface air temperature and precipitation series for the European Climate Assessment," *International Journal of Climatology*, 22, 1441–1453.