Hierarchical Modeling for Non-Gaussian Spatial Data in R

Andrew O. Finley and Sudipto Banerjee

March 6, 2013

We make use of several libraries in the following example session, including:

- `library(spBayes)`
- `library(fields)`
- `library(MBA)`

The non-Gaussian outcome data we commonly encounter are binary, rates, or counts which can often be modeled using either a binomial or Poisson generalized linear model (GLM). The function `spGLM` fits the Poisson and binomial model using the log and logit link function, respectively.

1 Poisson regression with spatial random effects

Here we illustrate the use of `spGLM` to fit a Poisson generalized linear mixed model with spatially dependent random effects. Data from this model are generated in the code block below.

```
> n <- 50
> coords <- cbind(runif(n, 0, 1), runif(n, 0, 1))
> phi <- 3/0.5
> sigma.sq <- 2
> R <- exp(-phi * iDist(coords))
> w <- mvrnorm(1, rep(0, n), sigma.sq * R)
> beta.0 <- 0.1
> y <- rpois(n, exp(beta.0 + w))
```

Assuming there is no spatial dependence we might fit a simple non-spatial GLM using

```
> pois.nonsp <- glm(y ~ 1, family = "poisson")
> beta.starting <- coefficients(pois.nonsp)
> beta.tuning <- t(chol(vcov(pois.nonsp)))
```

These coefficients and the Cholesky square root of the parameters' estimated covariances will be used as starting values and Metropolis sampler tuning values in the subsequent call to `spGLM`. In addition to the regression coefficients we specify starting values for the spatial range `phi` and variance `sigma.sq` as well as the random spatial effects `w`.

Here posterior inference is based on three MCMC chains each of length 15,000. The code to generate the first of these chains is given below.

```
> n.batch <- 300
> batch.length <- 50
> n.samples <- n.batch * batch.length
> pois.sp.chain.1 <- spGLM(y ~ 1, family = "poisson", coords = coords,
```

```
+     starting = list(beta = beta.starting, phi = 3/0.5, sigma.sq = 1,
+         w = 0), tuning = list(beta = 0.1, phi = 0.5, sigma.sq = 0.1,
+         w = 0.1), priors = list("beta.Flat", phi.Unif = c(3/1, 3/0.1),
+         sigma.sq.IG = c(2, 1)), amcmc = list(n.batch = n.batch,
+         batch.length = batch.length, accept.rate = 0.43), cov.model = "exponential",
+     verbose = TRUE, n.report = 500)

----------------------------------------
        General model description
----------------------------------------
Model fit with 50 observations.

Number of covariates 1 (including intercept if specified).

Using the exponential spatial correlation model.

Number of MCMC samples 15000.

Priors and hyperpriors:
        beta flat.

        sigma.sq IG hyperpriors shape=2.00000 and scale=1.00000

        phi Unif hyperpriors a=3.00000 and b=30.00000

Adaptive Metropolis with target acceptance rate: 43.0
------------------------------------------------
                Sampling
------------------------------------------------
Sampled: 15000 of 15000, 100.00%
------------------------------------------------
```

Again we use the `coda` package's `plot` function to plot chain trace plots, Figure 1.

```
> samps <- mcmc.list(pois.sp.chain.1$p.beta.theta.samples, pois.sp.chain.2$p.beta.theta.samples,
+     pois.sp.chain.3$p.beta.theta.samples)
> plot(samps)
```

The convergence of multiple chains can be assessed using diagnostics detailed in Gelman and Rubin (1992). The `gelman.diag` function in the **coda** package calculates the "potential scale reduction factor" for each each parameter, along with the associated upper and lower confidence limits. Approximate convergence is diagnosed when the upper confidence limit is close to 1. We can also plot the Gelman and Rubin's shrink factor versus the number of MCMC samples, here again convergence is diagnosed when the upper confidence limit remain close to 1. Figure 2 suggests we should discard the first ∼10,000 samples as burn in prior to summarizing the parameters' posterior distributions.

```
> print(gelman.diag(samps))

Potential scale reduction factors:

            Point est. Upper C.I.
(Intercept)       1.09       1.20
sigma.sq          1.02       1.05
```

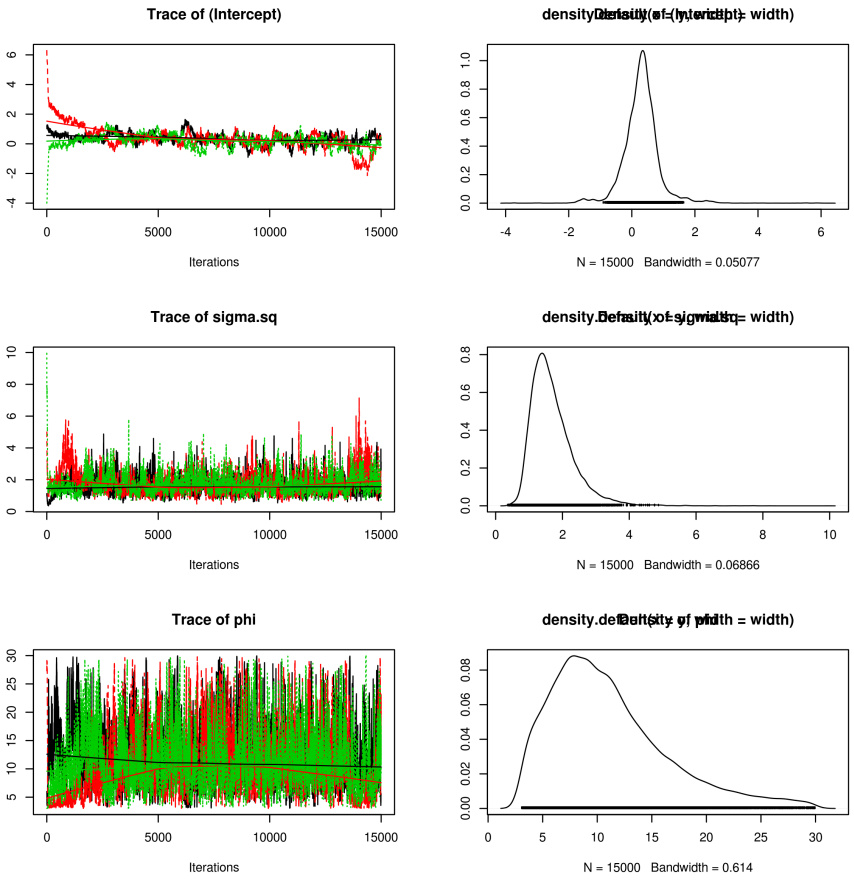

Figure 1: MCMC chain trace plots.

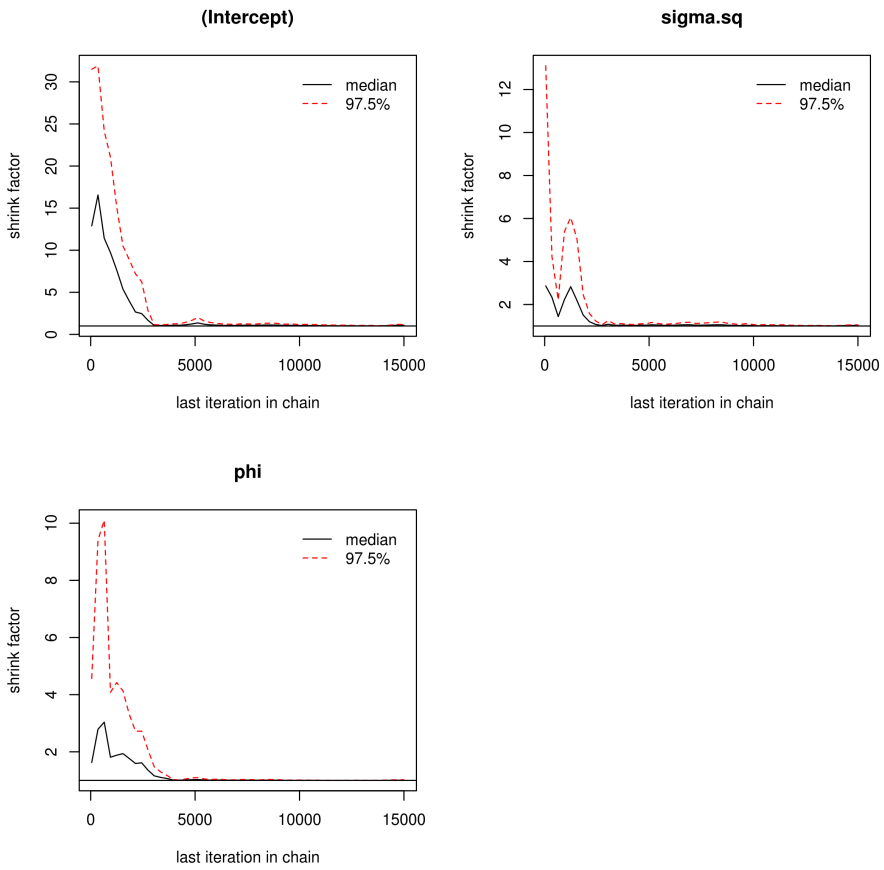

Figure 2: MCMC chain convergence diagnostics.

```
phi                1.01       1.02

Multivariate psrf

1.03
```

```
> gelman.plot(samps)
> burn.in <- 10000
> print(round(summary(window(samps, start = burn.in))$quantiles[,
+     c(3, 1, 5)], 2))
```

```
             50%  2.5% 97.5%
(Intercept) 0.18 -1.44  0.82
sigma.sq    1.58  0.87  3.41
phi         9.89  3.69 23.98
```

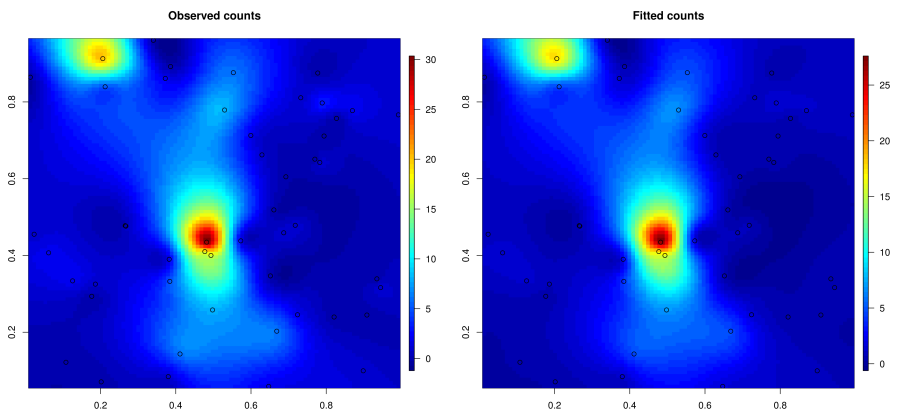

Figure 3: Observed and estimated counts.

Given the post burn in samples, we can also generate surfaces of the estimated counts Figure 3.

```
> samps <- as.matrix(window(samps, start = burn.in))
> w <- cbind(pois.sp.chain.1$p.w.samples[, burn.in:n.samples], pois.sp.chain.2$p.w.samples[,
+     burn.in:n.samples], pois.sp.chain.3$p.w.samples[, burn.in:n.samples])
> beta.0.hat <- mean(samps[, "(Intercept)"])
> w.hat <- apply(w, 1, mean)
> y.hat <- exp(beta.0.hat + w.hat)
> par(mfrow = c(1, 2))
> surf <- mba.surf(cbind(coords, y), no.X = 100, no.Y = 100, extend = TRUE)$xyz.est
> image.plot(surf, main = "Observed counts")
> points(coords)
> surf <- mba.surf(cbind(coords, y.hat), no.X = 100, no.Y = 100, extend = TRUE)$xyz.est
> image.plot(surf, main = "Fitted counts")
> points(coords)
```

Given the posterior samples of the model parameters, we use composition sampling to draw from the posterior predictive distribution of any new location. The code block below constructs a grid to define the prediction locations then calls `spPredict`. For each new location's posterior predictive samples we can draw a corresponding vector of realizations from `rpois`. These realization can then summarized to assess predictive performance. For example Figure 4 illustrates the median prediction over the domain.

```
> pred.coords <- as.matrix(expand.grid(seq(0.01, 0.99, length.out = 20),
+     seq(0.01, 0.99, length.out = 20)))
> pred.covars <- as.matrix(rep(1, nrow(pred.coords)))
> pois.pred <- spPredict(pois.sp.chain.1, start = 10000, thin = 10,
+     pred.coords = pred.coords, pred.covars = pred.covars, verbose = FALSE)
> y.pred <- apply(pois.pred$p.y.predictive.samples, 1, median)

> par(mfrow = c(1, 2))
> surf <- mba.surf(cbind(coords, y), no.X = 100, no.Y = 100, extend = TRUE)$xyz.est
> image.plot(surf, main = "Observed counts")
> points(coords)
> surf <- mba.surf(cbind(pred.coords, y.pred), no.X = 100, no.Y = 100,
```

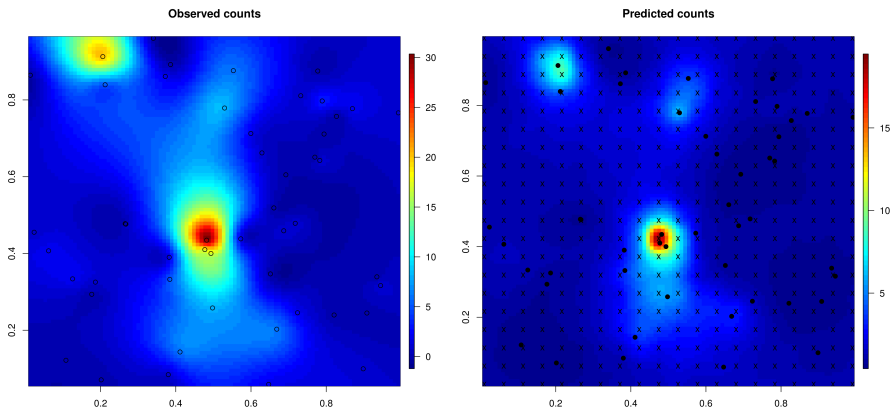

Figure 4: Observed and predicted counts.

```
+     extend = TRUE)$xyz.est
> image.plot(surf, main = "Predicted counts")
> points(pred.coords, pch = "x", cex = 1)
> points(coords, pch = 19, cex = 1)
```

2 Binomial regression with spatial random effects

Now let's consider the case where observations are either zero or one at locations. For example, the data could be the presence or absence of a species at a sample location or perhaps a positive or negative response to some experimental manipulation. This example illustrate a Binomial generalized linear mixed model with spatially dependent random effects. Here too, we consider different number of trials across locations.

```
> coords <- as.matrix(expand.grid(seq(0, 100, length.out = 8), seq(0,
+     100, length.out = 8)))
> n <- nrow(coords)
> phi <- 3/50
> sigma.sq <- 2
> R <- sigma.sq * exp(-phi * as.matrix(dist(coords)))
> w <- mvrnorm(1, rep(0, n), R)
> x <- as.matrix(rep(1, n))
> beta <- 0.1
> p <- 1/(1 + exp(-(x %*% beta + w)))
> weights <- rep(1, n)
> weights[coords[, 1] > mean(coords[, 1])] <- 10
> y <- rbinom(n, size = weights, prob = p)

> fit <- glm((y/weights) ~ x - 1, weights = weights, family = "binomial")
> beta.starting <- coefficients(fit)
> beta.tuning <- t(chol(vcov(fit)))
> n.batch <- 200
> batch.length <- 50
> n.samples <- n.batch * batch.length
```

```
> m.1 <- spGLM(y ~ 1, family = "binomial", coords = coords, weights = weights,
+     starting = list(beta = beta.starting, phi = 0.06, sigma.sq = 1,
+         w = 0), tuning = list(beta = beta.tuning, phi = 0.5, sigma.sq = 0.5,
+         w = 0.5), priors = list(beta.Normal = list(0, 10), phi.Unif = c(0.03,
+         0.3), sigma.sq.IG = c(2, 1)), amcmc = list(n.batch = n.batch,
+         batch.length = batch.length, accept.rate = 0.43), cov.model = "exponential",
+     verbose = TRUE, n.report = 50)

----------------------------------------
        General model description
----------------------------------------
Model fit with 64 observations.

Number of covariates 1 (including intercept if specified).

Using the exponential spatial correlation model.

Number of MCMC samples 10000.

Priors and hyperpriors:
        beta normal:
                mu:        0.000
                sd:        10.000


        sigma.sq IG hyperpriors shape=2.00000 and scale=1.00000

        phi Unif hyperpriors a=0.03000 and b=0.30000

Adaptive Metropolis with target acceptance rate: 43.0
------------------------------------------------
                Sampling
------------------------------------------------
Batch: 50 of 200, 25.00%
        parameter        acceptance        tuning
        beta[0]                72.0                0.18334
        sigma.sq        38.0                0.44792
        phi                74.0                0.83265
------------------------------------------------
Batch: 100 of 200, 50.00%
        parameter        acceptance        tuning
        beta[0]                58.0                0.25759
        sigma.sq        42.0                0.43035
        phi                56.0                1.34562
------------------------------------------------
Batch: 150 of 200, 75.00%
        parameter        acceptance        tuning
        beta[0]                46.0                0.27904
        sigma.sq        28.0                0.47561
        phi                54.0                2.04798
------------------------------------------------
Sampled: 10000 of 10000, 100.00%
```

```
------------------------------------------------

> burn.in <- 0.9 * n.samples
> sub.samps <- burn.in:n.samples
> print(summary(window(m.1$p.beta.theta.samples, start = burn.in)))

Iterations = 9000:10000
Thinning interval = 1
Number of chains = 1
Sample size per chain = 1001

1. Empirical mean and standard deviation for each variable,
   plus standard error of the mean:

               Mean      SD Naive SE Time-series SE
(Intercept) -0.5566 0.15581 0.004925       0.017730
sigma.sq     0.6148 0.22684 0.007170       0.030316
phi          0.1998 0.06122 0.001935       0.006436

2. Quantiles for each variable:

                2.5%     25%     50%     75%   97.5%
(Intercept) -0.82374 -0.6598 -0.5615 -0.4497 -0.2757
sigma.sq     0.27533  0.4540  0.5685  0.7395  1.1728
phi          0.09322  0.1520  0.2009  0.2567  0.2950

> beta.hat <- m.1$p.beta.theta.samples[sub.samps, "(Intercept)"]
> w.hat <- m.1$p.w.samples[, sub.samps]
> p.hat <- 1/(1 + exp(-(x %*% beta.hat + w.hat)))
> y.hat <- apply(p.hat, 2, function(x) {
+     rbinom(n, size = weights, prob = p)
+ })
> y.hat.mu <- apply(y.hat, 1, mean)
> y.hat.var <- apply(y.hat, 1, var)
> par(mfrow = c(1, 2))
> surf <- mba.surf(cbind(coords, y.hat.mu), no.X = 100, no.Y = 100,
+     extend = TRUE)$xyz.est
> image.plot(surf, main = "Estimated number of successful trials")
> text(coords, label = paste("(", y, ")", sep = ""))
> surf <- mba.surf(cbind(coords, y.hat.var), no.X = 100, no.Y = 100,
+     extend = TRUE)$xyz.est
> image.plot(surf, main = "Variance of estimated number of successful trials\n(observed # of trials)")
> text(coords, label = paste("(", weights, ")", sep = ""))
```

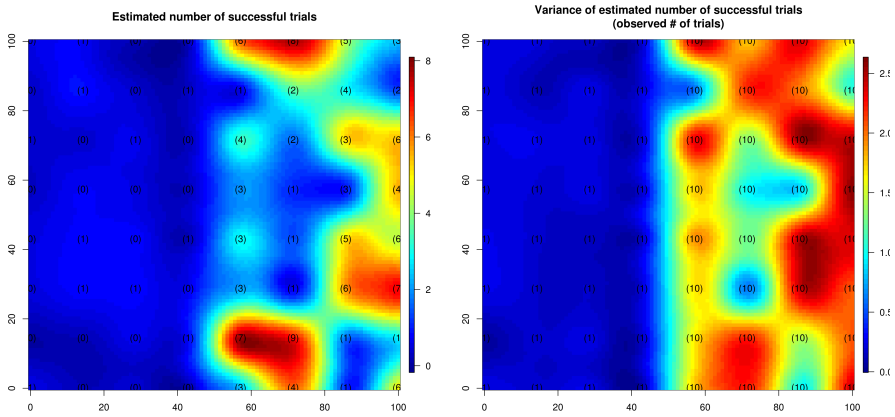

Figure 5: Mean of the posterior success rate and associated variance. Number values indicate number of successful trials (left) and total number of trials (right) at each location.

3 References

Banerjee, S., Carlin, B.P., and Gelfand, A.E. (2004). *Hierarchical Modeling and Analysis for Spatial Data*, Boca Raton, FL: Chapman and Hall/CRC Press.

Bivand, R.B., Pebesma, E.J., and Gómez-Rubio, V. (2008). *Applied Spatial Data Analysis with R*, UseR! Series, Springer.

Brooks, S.P. and Gelman, A. (1998). General Methods for Monitoring Convergence of Iterative Simulations. *Journal of Computational and Graphical Statistics*, 7:434–455.

Diggle, P.J. and Riberio, P.J. (2007). *Model-based Geostatistics*, Series in Statistics, Springer.

Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B. (2004). *Bayesian Data Analysis*. Second Edition. Boca Raton, FL: Chapman and Hall/CRC Press.

Gelman, A and Rubin, D.B. (1992). Inference from iterative simulation using multiple sequences. *Statistical Science*, 7:457–511.