Illustration of Adaptive MCMC

Andrew O. Finley and Sudipto Banerjee

September 5, 2014

1 Overview

The `adaptMetropGibb` function in **spBayes** generates MCMC samples for a continuous random vector using an adaptive Metropolis within Gibbs algorithm (see Roberts and Rosenthal, 2006; Rosenthal, 2007). We illustrate this function by fitting a simple univariate spatial regression model to some synthetic data.

In the code block below, we generate some synthetic data then set our priors. We use a logit transformation to project ϕ to the real-line and log transform σ^2 and τ^2. These transformations necessitate computing their respective Jacobians in the log target density.

```
> set.seed(1)
> n <- 50
> x <- runif(n, 0, 100)
> y <- runif(n, 0, 100)
> D <- as.matrix(dist(cbind(x, y)))
> phi <- 3/50
> sigmasq <- 50
> tausq <- 20
> mu <- 150
> s <- (sigmasq * exp(-phi * D))
> w <- mvrnorm(1, rep(0, n), s)
> Y <- mvrnorm(1, rep(mu, n) + w, tausq * diag(n))
> X <- as.matrix(rep(1, length(Y)))

> a.sig <- 2
> b.sig <- 100
> a.tau <- 2
> b.tau <- 100
> a.phi <- 3/500
> b.phi <- 3/3
> logit <- function(theta, a, b) {
+     log((theta - a)/(b - theta))
+ }
> logit.inv <- function(z, a, b) {
```

```
+     b - (b - a)/(1 + exp(z))
+ }
> target <- function(theta) {
+     mu.cand <- theta[1]
+     sigmasq.cand <- exp(theta[2])
+     tausq.cand <- exp(theta[3])
+     phi.cand <- logit.inv(theta[4], a.phi, b.phi)
+     Sigma <- sigmasq.cand * exp(-phi.cand * D) +
+         tausq.cand * diag(n)
+     SigmaInv <- chol2inv(chol(Sigma))
+     logDetSigma <- determinant(Sigma, log = TRUE)$modulus[1]
+     out <- (-(a.sig + 1) * log(sigmasq.cand) - b.sig/sigmasq.cand -
+         (a.tau + 1) * log(tausq.cand) - b.tau/tausq.cand +
+         log(sigmasq.cand) + log(tausq.cand) + log(phi.cand -
+         a.phi) + log(b.phi - phi.cand) - 0.5 * logDetSigma -
+         0.5 * (t(Y - X %*% mu.cand) %*% SigmaInv %*%
+             (Y - X %*% mu.cand)))
+     return(out)
+ }
> inits <- c(0, log(1000), log(1000), logit(3/10, a.phi,
+     b.phi))
> metrop.out <- adaptMetropGibbs(ltd = target, starting = inits,
+     batch = 500, batch.length = 25, report = 100)

-------------------------------------------------
                Sampling
-------------------------------------------------
Batch: 100 of 500
Metropolis batch acceptance rate:
0.560        0.320        0.520        0.400
-------------------------------------------------
Batch: 200 of 500
Metropolis batch acceptance rate:
0.320        0.600        0.440        0.520
-------------------------------------------------
Batch: 300 of 500
Metropolis batch acceptance rate:
0.480        0.520        0.440        0.520
-------------------------------------------------
Batch: 400 of 500
Metropolis batch acceptance rate:
0.360        0.480        0.560        0.600
-------------------------------------------------

> summary(mcmc(metrop.out$acceptance))

Iterations = 1:500
```

```
Thinning interval = 1
Number of chains = 1
Sample size per chain = 500

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

       Mean      SD Naive SE Time-series SE
[1,] 0.5665 0.19746 0.008831       0.066404
[2,] 0.4464 0.10873 0.004863       0.007258
[3,] 0.4512 0.09877 0.004417       0.005732
[4,] 0.4839 0.12735 0.005695       0.017598

2. Quantiles for each variable:

     2.5%  25%  50%  75% 97.5%
var1 0.20 0.40 0.56 0.68 0.941
var2 0.24 0.36 0.44 0.52 0.640
var3 0.28 0.40 0.44 0.52 0.640
var4 0.24 0.40 0.48 0.56 0.720
```

```
> plot(mcmc(metrop.out$acceptance))
```

```
> p.samples <- metrop.out$p.theta.samples
> p.samples[, 2] <- exp(metrop.out$p.theta.samples[,
+     2])
> p.samples[, 3] <- exp(metrop.out$p.theta.samples[,
+     3])
> p.samples[, 4] <- 3/logit.inv(metrop.out$p.theta.samples[,
+     4], a.phi, b.phi)
> colnames(p.samples) <- c("mu", "sigma.sq", "tau.sq",
+     "effective range")
> plot(mcmc(p.samples), smooth = FALSE, density = FALSE)
```

```
> n.samples <- nrow(p.samples)
> burn.in <- as.integer(0.25 * n.samples)
> p.samples <- mcmc(p.samples[burn.in:n.samples, ])
> summary(p.samples)
```

```
Iterations = 1:9376
Thinning interval = 1
Number of chains = 1
Sample size per chain = 9376

1. Empirical mean and standard deviation for each variable,
```

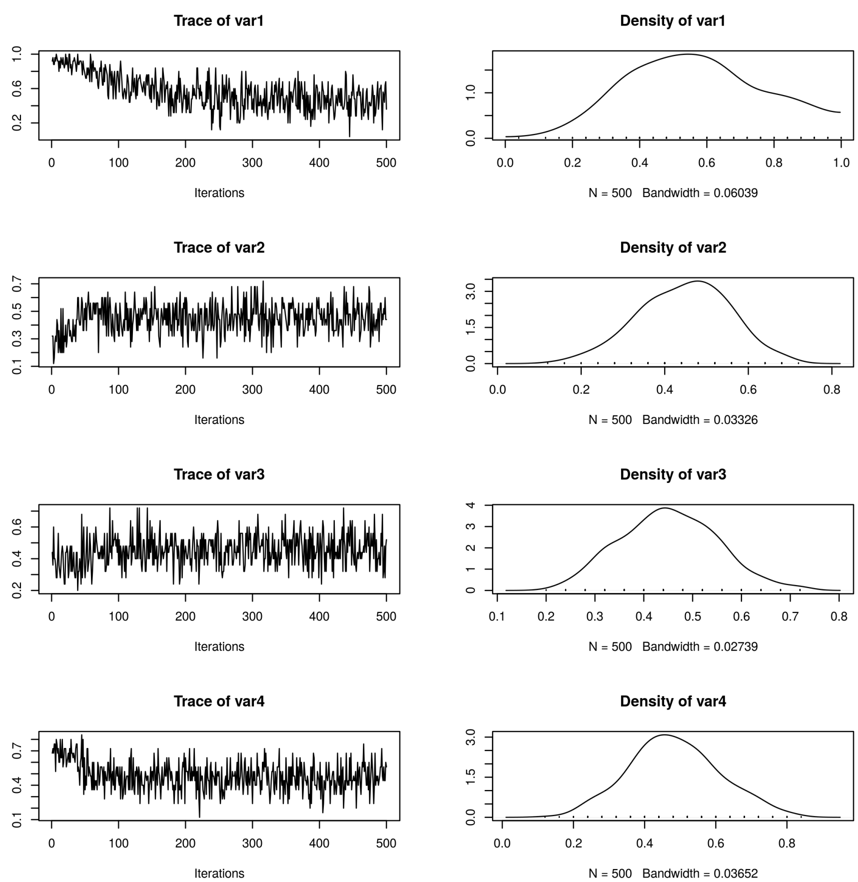

Figure 1: Metropolis acceptance trace plots.

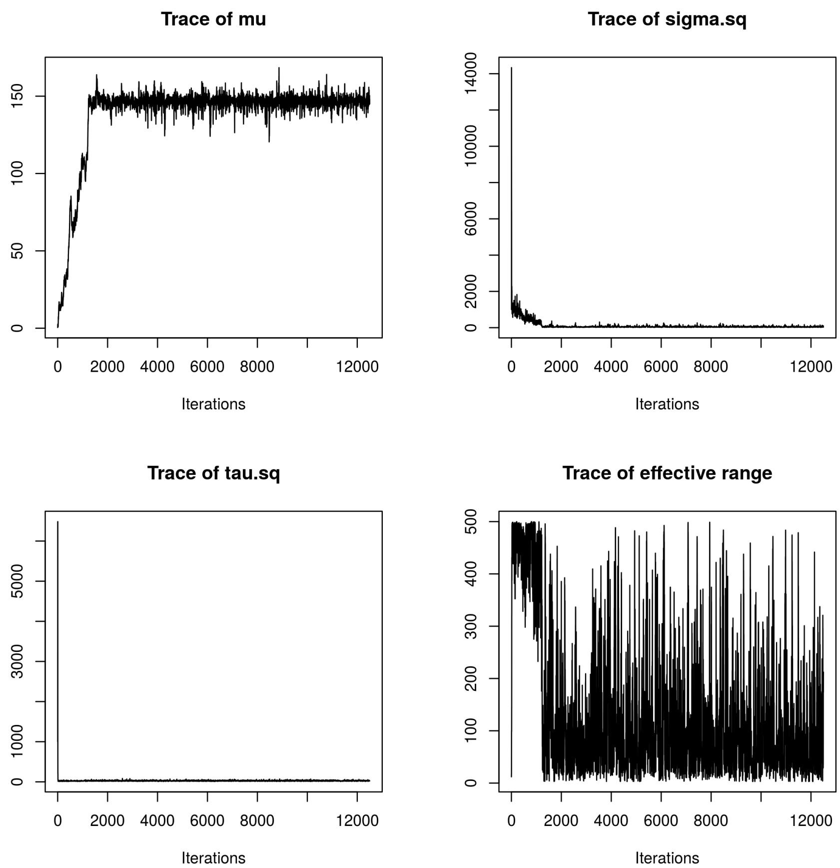

Figure 2: MCMC trace plots.

```
   plus standard error of the mean:

                  Mean     SD Naive SE Time-series SE
mu              146.42  3.745  0.03868         0.1143
sigma.sq         49.50 28.985  0.29934         1.2746
tau.sq           27.14  8.991  0.09285         0.2396
effective range  96.41 92.391  0.95416         4.3259

2. Quantiles for each variable:

                   2.5%    25%    50%    75%  97.5%
mu              137.487 144.87 146.72 148.40 153.54
sigma.sq         18.411  31.46  41.89  57.99 132.06
tau.sq           13.472  20.62  25.66  32.24  47.98
effective range   5.639  34.62  63.89 123.47 374.48

> plot(p.samples, density = FALSE)
```

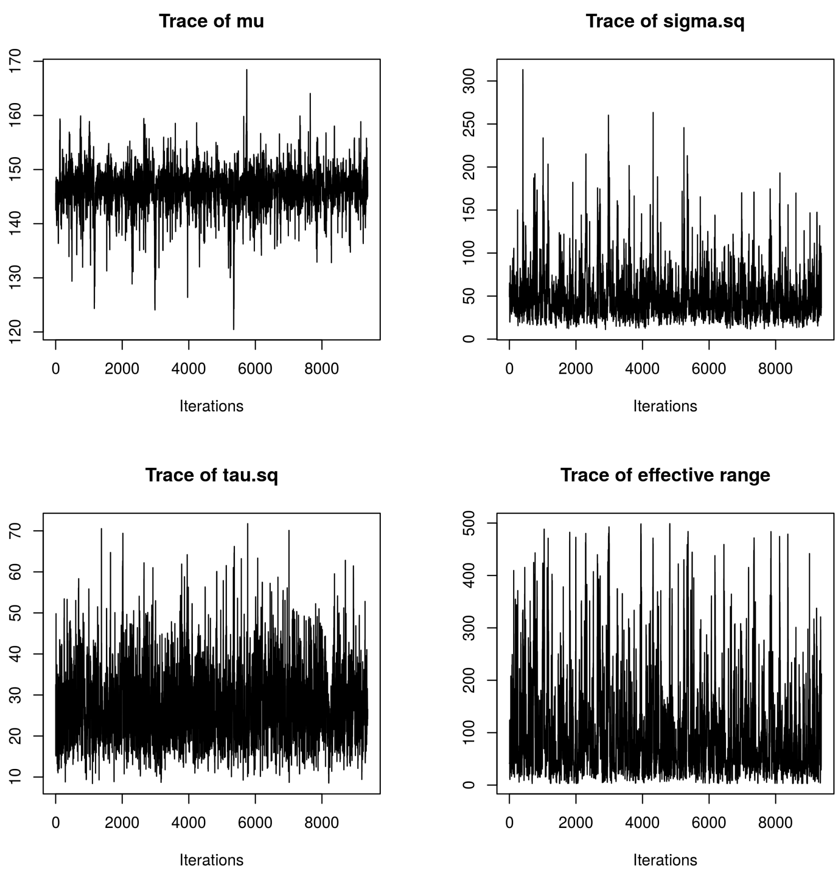

Figure 3: MCMC trace plots post burn-in.

2 References

Roberts G.O. and Rosenthal J.S. (2006). Examples of Adaptive MCMC. http://probability.ca/jeff/ftpdir/adaptex.pdf Preprint.

Rosenthal J.S. (2007). AMCMC: An R interface for adaptive MCMC. *Computational Statistics and Data Analysis*, 51:5467-5470.