# Computer lab 4: Markov Chain Monte Carlo

## Learning objectives

The main objective of this computer lab is to make the student familiar with major tools for Markov Chain Monte Carlo simulation in R.

After completing the lab the student shall be able to:

1. Evaluate properties of the Metropolis-Hastings algorithm in R.
2. Implement and apply the Gibbs sampling techniques in R.

## Recommended reading

Chapter 7 - 8 in Givens and Hoeting (2013)

## Assignment 1: Different versions of the Metropolis-Hastings algorithm

a.) The function normm is simulating from a normal with zero mean and unit variance

using a Metropolis algorithm with uniform proposal distribution. The exercise is to try the following alpha = 0.1,1 and 200 for 2000 iterations. Present trace plots of line type and histograms with interpretations. Tune the sampler by finding a value of alpha that gives an acceptance probability of 0.3. You have to modify the code in order to save the acceptance rate. Explain the code with comments.

normm<-function (Nsim, a)

{

vec <- vector("numeric", Nsim)

x <- 0

vec[1] <- x

for (i in 2:Nsim) {

innov <- runif(1, -a, a)

Xstar <- x + innov

aprob <- min(1, dnorm(Xstar)/dnorm(x))

u <- runif(1)

if (u < aprob)

x <- Xstar

vec[i] <- x

}

vec

}

b.) Function gammh is a Metropolis-Hastings independence sampling algorithm with normal proposal distribution with the same mean and variance as the desired gamma. Try a = 0.1, 2 and b = 0.01, 2. Present trace plots and histograms with interpretations. Explain the code with comments.

gammh<-function (Nsim, a, b)

{

mu <- a/b

sig <- sqrt(a/(b \* b))

vec <- vector("numeric", Nsim)

x <- a/b

vec[1] <- x

for (i in 2:Nsim) {

can <- rnorm(1, mu, sig)

aprob <- min(1, (dgamma(can, a, b)/dgamma(x,

a, b))/(dnorm(can, mu, sig)/dnorm(x,mu, sig)))

u <- runif(1)

if (u < aprob)

x <- can

vec[i] <- x

}

vec

}

## Assignment 2: The Gibbs sampling algorithm for the one-way random effects model

a). Implement R-code for the normal one-way random effects model with *k* levels. One way of parameterizing this model is:

The sampling should be performed from the full conditional distributions

where and can be set to 0.001 and . Compare your code with the results obtained with the Gibbs sampler implemented in the MCMChregress() function in the MCMCpack library. Provide results in Tables with mean, standard deviations and 95% credible intervals for the mean, the random effects and all variances for both your own code and the MCMChregress() analysis.

The data comes from the WinBUGS example Dyes where Yield is the response variable and random effect Batch has 6 levels:

Batch Yield (in grams)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1 1545 1440 1440 1520 1580

2 1540 1555 1490 1560 1495

3 1595 1550 1605 1510 1560

4 1445 1440 1595 1465 1545

5 1595 1630 1515 1635 1625

6 1520 1455 1450 1480 1445

## To hand in

A written report (preferably a Word or .pdf document) where you summarize your main findings in the assignments. Submit your report via Lisam before the deadline.