LaplacesDemon Examples

Byron Hall
STATISTICAT, LLC

Abstract

The **LaplacesDemon** package in R enables Bayesian inference with any Bayesian model, provided the user specifies the likelihood. This vignette provides examples of how to specify different model forms.

Keywords:~Bayesian, Bayesian Inference, Laplace's Demon, LaplacesDemon, R, STATISTICAT.

A formal introduction to Laplace's Demon is provided in an accompanying vignette entitled "**LaplacesDemon** Tutorial", and an introduction to Bayesian inference is provided in the "Bayesian Inference" vignette.

The purpose of this document is to provide users of the **LaplacesDemon** package (Hall 2011) with examples of a variety of Bayesian methods. To conserve space, the examples are not worked out in detail, but provide necessary materials for using the various methodologies. Necessary materials include data (which is often simulated), initial values, and the `Model` function. This document is expected to grow over time as examples of more methods become included. Contributed examples are welcome. Please send contributed examples in a similar format in an email to `statisticat@gmail.com` for review.

1. Normal, Multilevel

This is Gelman's school example (Gelman, Carlin, Stern, and Rubin 2004). Note this does not converge as quickly as the example using Gibbs sampling in the **R2WinBUGS** package (Gelman 2009), an R (R Development Core Team 2010) package on CRAN, but also note that with further sampling, Laplace's Demon provides a better answer (higher `ESS`, etc.), even though the **R2WinBUGS** example uses of 3 chains for more indications of convergence.

1.1. Data

```
J <- 8
y <- c(28.4, 7.9, -2.8, 6.8, -0.6, 0.6, 18.0, 12.2)
```

```
sd <- c(14.9, 10.2, 16.3, 11.0, 9.4, 11.4, 10.4, 17.6)
parm.names <- 2*J+2
for (j in 1:J) {parm.names[j] <- paste("theta[",j,"]",sep="")}
parm.names[J+1] <- paste("theta.mu[",j,"]",sep="")
parm.names[J+2] <- paste("log.theta.sigma[",j,"]",sep="")
MyData <- list(J=J, parm.names=parm.names, sd=sd, y=y)
```

1.2. Initial Values

```
Initial.Values <- rep(0,J+2)
```

1.3. Model

```
Model <- function(parm, MyData)
     {
     ### Prior Parameters
     theta.mu <- parm[J+1]
     log.theta.sigma <- parm[J+2]
     tau.alpha <- 1.0E-3
     tau.beta <- 1.0E-3
     ### Parameters
     tau <- theta <- rep(0,J)
     for (j in 1:J) {theta[j] <- parm[j]; tau[j] <- sd[j]^2}
     ### Log Prior Densities
     tau.prior <- theta.prior <- rep(0,J)
     for (j in 1:J) {
          tau.prior[j] <- dgamma(tau[j], tau.alpha, tau.beta, log=TRUE)
          theta.prior[j] <- dnorm(theta[j], theta.mu,
               exp(log.theta.sigma), log=TRUE)}
     ### Log-Posterior
     LL <- sum(dnorm(y, theta, 1/sqrt(tau), log=TRUE))
     LP <- LL + sum(theta.prior) + sum(tau.prior)
     Modelout <- list(LP=LP, Dev=-2*LL, Monitor=exp(log.theta.sigma),
          yhat=theta)
     return(Modelout)
     }
```

2. Linear Regression

2.1. Data

```
N <- 10000
J <- 5
X <- matrix(1,N,J)
```

```
for (j in 2:J) {X[,j] <- rnorm(N,runif(1,-3,3),runif(1,0.1,1))}
beta <- runif(J,-3,3)
e <- rnorm(N,0,0.1)
y <- beta %*% t(X) + e
parm.names <- rep(NA, J+1)
for (j in 1:J) {parm.names[j] <- paste("beta[",j,"]",sep="")}
parm.names[J+1] <- "log.tau"
MyData <- list(J=J, X=X, parm.names=parm.names, y=t(y))
```

2.2. Initial Values

```
Initial.Values <- c(rep(0,J), log(1))
```

2.3. Model

```
Model <- function(parm, Data)
     {
     ### Prior Parameters
     beta.mu <- rep(0,J)
     beta.tau <- rep(1.0E-3,J)
     tau.alpha <- 1.0E-3
     tau.beta <- 1.0E-3
     ### Parameters
     beta <- rep(0,J)
     for (j in 1:J) {beta[j] <- parm[j]}
     tau <- exp(parm[J+1])
     ### Log Prior Densities
     beta.prior <- rep(0,J)
     for (j in 1:J) {
          beta.prior[j] <- dnorm(beta[j], beta.mu[j],
               1/sqrt(beta.tau[j]), log=TRUE)}
     tau.prior <- dgamma(tau, tau.alpha, tau.beta, log=TRUE)
     ### Log-Posterior
     mu <- beta %*% t(X)
     LL <- sum(dnorm(y, mu, 1/sqrt(tau), log=TRUE))
     LP <- LL + sum(beta.prior) + tau.prior
     Modelout <- list(LP=LP, Dev=-2*LL, Monitor=c(tau,mu[1]), yhat=mu)
     return(Modelout)
     }
```

3. Poisson Regression

3.1. Data

```
N <- 10000
```

```
J <- 5
X <- matrix(1,N,J)
for (j in 2:J) {X[,j] <- rnorm(N,runif(1,-3,3),runif(1,0.1,1))}
beta <- runif(J,-3,3)
e <- rnorm(N,0,0.1)
y <- exp(beta %*% t(X)) + e
parm.names <- rep(NA,J+1)
for (j in 1:J) {parm.names[j] <- paste("beta[",j,"]",sep="")}
parm.names[J+1] <- "log.tau"
MyData <- list(J=J, X=X, parm.names=parm.names, y=t(y))
```

3.2. Initial Values

```
Initial.Values <- rep(0,J)
```

3.3. Model

```
Model <- function(parm, MyData)
     {
     ### Prior Parameters
     beta.mu <- rep(0,J)
     beta.tau <- rep(1.0E-3,J)
     ### Parameters
     beta <- rep(0,J)
     for (j in 1:J) {beta[j] <- parm[j]}
     ### Log Prior Densities
     beta.prior <- rep(0,j)
     for (j in 1:J) {
          beta.prior[j] <- dnorm(beta[j], beta.mu[j],
               1/sqrt(beta.tau[j]), log=TRUE)}
     ### Log-Posterior
     lambda <- exp(beta %*% t(X))
     LL <- sum(dpois(y, lambda, log=TRUE))
     LP <- LL + sum(beta.prior)
     Modelout <- list(LP=LP, Dev=-2*LL, Monitor=c(lambda[1:2]), yhat=lambda)
     return(Modelout)
     }
```

References

Gelman A (2009). ***R2WinBUGS**: Running WinBUGS and OpenBUGS from R / S-PLUS.* R package version 2.1-16, URL http://www.R-project.org/package=R2WinBUGS.

Gelman A, Carlin J, Stern H, Rubin D (2004). *Bayesian Data Analysis*. 2nd edition. Chapman & Hall, Boca Raton, FL.

Hall B (2011). ***LaplacesDemon**: Software for Bayesian Inference.* R package version 11.01.18, URL http://www.R-project.org/package=LaplacesDemon.

R Development Core Team (2010). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org.

Affiliation:

Byron Hall
STATISTICAT, LLC
Farmington, CT
E-mail: statisticat@gmail.com
URL: http://www.statisticat.com/laplacesdemon.html