DOBAD Package: Gibbs Sampling MCMC of Linear Birth-Death Chain with Partial Data

Charles Doss

September 2009
Part I

Estimating Rates for Linear Birth-Death chain via Gibbs Sampler MCMC by Exact Conditional Simulation

We are demonstrating the use of the DOBAD package’s capability to do Bayesian estimation of the rate parameters for a linear Birth-Death chain, given partial observations, using the methods of Doss et al. (2010). Call the chain \( \{X(t)\}_{t \in \mathbb{R}} \), and its birth rate \( l_a \) and its death rate \( \mu \). We fix \( \beta \in \mathbb{R} \) and constrain \( \nu \), the immigration rate, to be \( \nu = \beta l_a \). We will denote \( \theta = (l_a, \mu) \). The data is the value of the process at a finite number of discrete time points. That is, for some fixed times \( 0 = t_0, t_1, \ldots, t_n \), we see the state of the process, \( X(t_i) \). Thus the data, \( D \), is 2 parts: a vector of the times \( t_i \), \( i = 0, \ldots, n \) and a vector of states at each of those times, \( s_i \), for \( i = 0, \ldots, n \) (where \( X(t_i) = s_i \). The gamma prior is the conjugate prior if we observed the chain continuously instead of partially. The way we proceed, then, is to use independent Gamma priors on the \( \lambda \) and \( \mu \) and augment the state space for our MCMC to include the entire chain \( \{X_t\}_{t \in [0, t_n]} \) by conditionally sampling \( \{X_t\}_{t \in [0, t_n]} | \theta | D \).

First we generate the underlying process and the “data”, set our prior parameters, and compute some summary statistics of the fully observed and partially observed processes.

```r
> library(DOBAD)

> ####### Generate the "data"

> initstate=7;
> set.seed(112);
> T=5; L <- .2; mu <- .4;
> beta.immig <- .987;
> trueParams <- c(L,mu,beta.immig); names(trueParams) <- c("lambda", "mu","beta") #for saving
```

1
> dr <- 0.0000000001; #Need |dr| < |L-mu| always o/w get sqrt(negative).
> n.fft <- 1024;
> delta <- 1;#play with. or make observation intervals distinct
> dat <- birth.death.simulant(t=T, lambda=L, mu=mu, nu=L*beta.immig, X0=initstate);
> fullSummary <- BDsummaryStats(dat); fullSummary

<table>
<thead>
<tr>
<th>Nplus</th>
<th>Nminus</th>
<th>Holdtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>12.00000</td>
<td>14.00000</td>
<td>26.73947</td>
</tr>
</tbody>
</table>

> #fullSummary <- BDsummaryStats(dat[[1]]); fullSummary
> MLEs <- M.step.SC( EMsuffStats=fullSummary, T=T, beta.immig= beta.immig); MLEs

<table>
<thead>
<tr>
<th>lambdahat</th>
<th>muhat</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.3788540</td>
<td>0.5235706</td>
</tr>
</tbody>
</table>

> partialData <- getPartialData( seq(0,T,delta), dat);
> observedSummary <- BDsummaryStats.PO(partialData); observedSummary;

<table>
<thead>
<tr>
<th>Nplus</th>
<th>Nminus</th>
<th>Holdtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>5</td>
<td>28</td>
</tr>
</tbody>
</table>

> ##Bayesian parameters
> L.mean <- 1; M.mean <- 1.1;
> aL <- .02;
> bL <- aL / L.mean
> aM <- .022;
> bM <- aM / M.mean;
> print(paste("Variances are", aL/bL^2, "and", aM/bM^2))

[1] "Variances are 50 and 55"

> N=10
> burn=0
Now we run the MCMC. It is set to run only a few iterations, obviously not enough for estimation, but enough to demonstrate the code. You may want to change the sweave code to run more iterations.

```r
> ##Rprof(file="mcmc.rprofout")
> timer <- system.time(theMCMC <- BD.MCMC.SC(Lguess=L.mean, Mguess=M.mean,
+ alpha.L=aL, beta.L=bL, # mean
+ alpha.M=aM, beta.M=bM, #mean of
+ beta.immig=beta.immig,
+ data= partialData,
+ burnIn=burn, N=N));
> ##Rprof(NULL)
> #theMCMC
> mean(theMCMC[,1]); #lambda
[1] 0.722452

> mean(theMCMC[,2]); #mu
[1] 0.9170729

> L;
[1] 0.2

> mu;
[1] 0.4

> timer;

        user  system elapsed
       0.489   0.003   0.492

> options(continue=" "); ##undo the setting we changed at top
```
> hist(theMCMC[,1], freq=FALSE, breaks=20,
    xlab="Lambda", ylab = "Density",
    main="Posterior of Lambda")
> Lmean <- mean(theMCMC[,1])
> abline(col="red", v=Lmean)
> abline(col="purple", v=L.mean)
> #text(col="red", y=-.3, x=Lmean, labels = "L")
> x <- seq(from=0,to=1, by=.01);
> y <- dgamma(x, shape=aL, rate=bL)
> lines(x,y, col="blue")

> hist(theMCMC[,2], freq=FALSE, breaks=20,
    xlab="Mu", ylab = "Density",
    main="Posterior of Mu")
> Mmean <- mean(theMCMC[,2])
> abline(col="red", v=Mmean)
> abline(col="purple", v=M.mean)
> x <- seq(from=0,to=1, by=.01);
> y <- dgamma(x, shape=aM, rate=bM)
> lines(x,y, col="blue")

References

Figure 1: Posterior Density Estimation of Lambda
Figure 2: Posterior Density Estimation of Mu