DOBAD Package: EM Algorithm on a Partially Observed Linear Birth-Death-Immigration Process

Charles Doss

2009

Part I

Estimating Rates for Linear Birth-Death-Immigration chain via EM Algorithm

We are demonstrating the use of the `DOBAD` package's capability to do estimation of the rate parameters for a linear Birth-Death-Immigration (BDI) chain, given partial observations, via the Expectation-Maximization (EM) algorithm. Call the chain $\{X(t)\}_{t\in\mathbb{R}}$, and its birth rate la and its death rate μ. Derivations are neater if we only have two parameters. In this document, we consider a third parameter, ν, the immigration rate. We will denote $\theta = (la, \mu, \nu)$. 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 $X(t_i) = s_i$.

In order to use the EM algorithm, we need to be able to calculate $E(N_i^+|X_0 = a, X_T = b)$, $E(N_T^-|X_0 = a, X_T = b)$, and $E(R_T|X_0 = a, X_T = b)$, where N_i^+ is the number of jumps up starting in state i, for $i = 0, \ldots, N_T^+$, in the time interval $[0, T]$, N_T^- is the number of jumps down in the time interval $[0, T]$, and R_T is the total holding time in the interval $[0, T]$ (i.e. $R_T = \sum_{i=0}^{\infty} i d_T(i)$ where $d_T(i)$ is the time spent in state i in the interval $[0, T]$). This code uses the method of Doss et al. (2010) (generating functions) to calculate the latter two expectations, and Monte Carlo for the former, which is more complicated to calculate. Thus the algorithm is a Monte Carlo EM (MCEM) algorithm.

We will set up the true parameters and a true chain, and then "observe" it partially, and see how the EM does on that data. First, set up the true parameters.

```
> library(DOBAD)
> set.seed(1156);
> initstate=4;
> T=25;
> L <- .3
> mu <- .6
> beta.immig <- 1.2;
> dr <- 0.000001; #Need |dr| < |L-mu| always o/w get sqrt(negative). (numerical differ'n)
> n.fft <- 1024;
> trueParams <- c(L,mu);
> names(trueParams) <- c("lambda", "mu")
>
```

Now we get the “truth” and then observe the “data” as well as calculate some information about both.

```
> ##Get the "data"
> dat <- birth.death.simulant(t=T, lambda=L, m=mu, nu=L*beta.immig, X0=initstate);
> fullSummary <- BDsummaryStats(dat);
> fullSummary
```

```
   Nplus   Nminus Holdtime
19.00000 23.00000 49.45406
```

```
> names(fullSummary) <- c("Nplus", "Nminus", "Holdtime");
> MLEs.FullyObserved <- M.step.SC( EMsuffStats=fullSummary, T=T, beta.immig= beta.immig);
> #MLEs
> ###MLE.FullyObserved are NOT the MLE for the EM, but hopefully close as delta --> 0
>
> #delta <- 2
> #observetimes <- seq(0,T,delta)
> observetimes <- sort(runif(20,min=0,max=T))
> partialData <- getPartialData( observetimes, dat);
> T <- getTimes(partialData)[length(getTimes(partialData))]
> observedSummary <- BDsummaryStats.PO(partialData); observedSummary;
```

```
   Nplus   Nminus Holdtime
 7.00000 10.00000 48.45867
```

```
> param0 <- c(.8,.9,1.1); names(param0) <- c("lambdahat", "muhat", "nuhat");
> #param0
>
```

Note that the difference between `fullSummary` and `observedSummary` is some measure of the information we're missing. The EM algorithm aspires to the MLEs of the full data, ie `MLEs.FullyObserved`. Now we run the actual EM algorithm. This might take a while! The number of iterations can be lowered but for Confidence Intervals to compute without error, the estimates must be accurate.

The initial setting is to only do a single iteration of the EM, due to its slowness. Note that the `EM.BD` function is not available in the DOBAD namespace, it requires the prefix `DOBAD:::`. This choice was made due to its slowness at this time.

```
> ## ##############################################################################
> ## ##########################Do Generic Optimization
> ## logLike <- function(rates){
> ##   BDloglikelihood.PO(partialDat=partialData, L=exp(rates[1]), m=exp(rates[2]),
> ##                      nu=exp(rates[3]), n.fft=1024);
> ## }
> ## genericEstimates <- optim(param0, logLike,
> ##                           ##method="L-BFGS-B",
> ##                           ##lower=c(0.0001, 0.0001, .0001), upper=c(100,100,100),
> ##                           control=list(fnscale=-1))
> ## print(genericEstimates <- exp(genericEstimates$par))
> ## print(logLike(log(genericEstimates)))
> ## ##############################################################################
> ## ###########################End Generic Optimization
>
> #########RUN EM
> iters <- 1
> tol <- .0000005;
> ##myInitParamMat <- rbind(c(0, 1.46,.65),
> ##                         c(.43, .4, 1.3));
> myInitParamMat <- rbind(c(.25,.26,.15));
> emOuts <- DOBAD:::EM.BD(dat=partialData, init.params.mat=myInitParamMat, tol=tol, M=iters,
+                         dr=1e-07, n.fft=1024,
+                         alpha=.2, beta=.3, fracSimIncr=3, numMCs.i.start=20,
+                         outputBestHist=FALSE)
```

```
lambdahat     muhat     nuhat
     0.25      0.26      0.15
[1] "The estimators at step 1 are"
lambdahat     muhat     nuhat
     0.25      0.26      0.15
[1] "numMCs for next time is  20"
```

```
> bestparams <- sapply(emOuts, function(emOut){ emOut[[iters+1]]$newParams});
> print(bestparams)
```

```
                 V1
lambdahat 0.1828848
muhat     0.3740791
nuhat     0.2473082
```

```
> #loglikes <- apply( as.matrix(bestparams),2, function(param){logLike(param)});
> #print(loglikes);
> ########### end Run EM
>
>
> ##save.image("BD_EM_wImmigRnw.rsav");
```

References

Doss, C., Suchard, M., Holmes, I., Kato-Maeda, M., and Minin, V. (2010). Great Expectations: EM Algorithms for Discretely Observed Linear Birth-Death-Immigration Processes. *Arxiv preprint arXiv:1009.0893* .