BDSEM Package: inferring MLE rates of discretely observed birth-death-shift processes

Jason Xu

September 4, 2014

1 Transition probabilities of birth-death-shift process

Here we demonstrate how to use the software to obtain and verify the accuracy of numerically computed transition probabilities of the BDS process. We walk through an illustration that is an analogous, simplified version of the transition probability comparison experiment in the main paper.

First, we specify desired birth, shift, and death rates λ, ν, μ as inputs. Then given an initial population size i, corresponding to an initial state of the process $\mathbf{X}(0) = (i, 0)$, we want to compute the transition probability $p_{(i,0),(k,l)(t)}$ accurately for any time interval length t. We do this for a list of different interval lengths using the function `getTrans.timeList`. This function implements our generating function approach to computing transition probabilities, and in addition to the arguments already mentioned, this requires inputs `s1.seq`, `s2.seq`, vectors of imaginary numbers evenly spaced around the unit circle. Their length determines the maximum k, l values for which transition probabilities are computed; accuracy improves with finer sequences, and their length defined by `gridLength` below should be a power of 2 for best results.

```
library(bdsem)
gridLength = 16
s1.seq <- exp(2 * pi * (0+1i) * seq(from = 0, to = (gridLength - 1))/gridLength)
s2.seq <- exp(2 * pi * (0+1i) * seq(from = 0, to = (gridLength - 1))/gridLength)
tList <- c(0.5, 2, 6, 10)
transProbs <- getTrans.timeList(tList, 0.0188, 0.00268, 0.0147, 10, s1.seq, s2.seq, 0.5)
```

The results are stored in `transProbs`, a list of matrices, where each entry contains a matrix corresponds to a time interval length from `tList`. The k, l entry of the matrix is then the transition probability $p_{(i,0),(k-1,l-1)(t)}$

For instance, in our example the transition probability $p_{(10,0),(8,0)}(6)$ is contained in

```
transProbs[[3]][9, 1]

## [1] 0.05284
```

since $t = 6$ is the fourth entry in `tList`.

Next, we can check these probabilities by simulating from the BDS process. We simulate `nSims` realizations of the process for length t, for all values t in `tList`. Each simulation begins at $\mathbf{X}(0) = (i, 0)$, and transition probabilities $p_{(i,0),(k,l)(t)}$ are computed by empirically computing the proportion of times the simulation ends with $\mathbf{X}(t) = (k, l)$.

These empirical transition probabilities via simulation serve as a ground truth benchmark for comparison, and are computed using `transProbs.MC`. The results are stored in the same format as returned by `getTrans.timeList`

To compare with previous methods, we can also compute the probabilities of one birth, death, or shift, as well as the probability of no event occurring, according to the frequent monitoring method: note this next section will take a while, even with a relatively small number of simulations

```
# nSims = 5000 #increase number of simulations for more precise estimates: omitted for
# speed
nSims = 5
transProbs.MC <- getTrans.MC(nSims, tList, 0.02, 0.008, 0.015, 10)
transProbs.FM <- getTrans.FreqMon(tList, 0.02, 0.008, 0.015, 10)
```

It is easy to compare entries of transProbs and transProbs.MC to verify that our method calculates accurate values. We can also plot the transition probabilities defined under frequent monitoring as computed in all three cases, similar to the more detailed figure in the main paper.

```
## Warning:  package 'plotrix' was built under R version 2.15.3
```