DIC Write-up

July 18, 2018

User-created functions are in the Appendix.

This write-up details the comparison of using latent space models (LSM), stochastic block models (SBM), and exponential random graph models (ERGM) on sample datasets. We compare model-fit using deviance information criterion (DIC). The sample datasets used are Zachary's Karate club, Sampson's monks, and florentine. The parameters below are used for each model. To implement the models the `latent` package for LSMs, the `CIDnetworks` package for SBMs, and `BERGM` package for ERGMs.

```
burnin = 500
sample_size = 2000
interval = 5
```

The following formulas and settings were used for the models. We use mutuality in the Sampson data since it is a directed network. We use two blocks for the SBMs.

```
######### Florentine data ############

#LSM
flo_LSM <- ergmm(flomarriage ~ edges + kstar(2),
               control=ergmm.control(burnin=burnin,
                                     sample.size= sample_size,
                                     interval=interval))
#SBM
flo_SBM<- CID.Gibbs(flo_adj, components = c(SBM(2)),
                    draws = sample_size, burnin = burnin,
                    thin = interval)
#ERGM
flo_ERGM <- bergm(flomarriage ~ edges + kstar(2),
               burn.in = burnin,
               aux.iters = sample_size,
               main.iters = sample_size,
               gamma = 1)

######### Karate data  ############

#LSM
zach_LSM <- ergmm(zach ~ edges + kstar(2),
                 control=ergmm.control(burnin=burnin,
                                       sample.size= sample_size,
#SBM
zach_SBM<- CID.Gibbs(flo_adj, components = c(SBM(2)),
                     draws = sample_size, burnin = burnin,
                     thin = interval)
#ERGM
zach_ERGM <- bergm(zach ~ edges + kstar(2),
                  burn.in = burnin,
                  aux.iters = sample_size,
                  main.iters = sample_size,
                  gamma = 1)

######### Sampson data ############

#LSM
samplike_LSM <- ergmm(samplike~edges + mutual,
                  control=ergmm.control(burnin=burnin,
                                        sample.size= sample_size,
                                        interval=interval))
#SBM
samplike_SBM<- CID.Gibbs(samplike_adj, components = c(SBM(2)),
                         draws = sample_size, burnin = burnin,
                         thin = interval)
#ERGM
samplike_ERGM <- bergm(samplike~edges + mutual,
                   burn.in = burnin,
                   aux.iters = sample_size,
                   main.iters = sample_size,
                   gamma = 1)
```

In the DIC measures below the LSM performs the worst, with the worst score for the florentine and karate club data and only second place for the Sampson data. The ERGM performs second best overall with a substantially better score than the other models for the Sampson data, but nearly as bad on the karate club data as the LSM and average score for the florentine data. The SBM performs the best overall, with the best score for both the florentine and karate club data. Although, the SBM performs the worst for the Sampson data.

DIC

	florentine	zach	samplike	average
LSM	112.00	391.38	301.51	268.30
SBM	**-264.90**	**106.81**	318.69	**53.53**
ERGM	-32.25	360.91	**-71.21**	85.81

Appendix: User created functions

```
 #calculate DIC from LSM object
LSM_DIC = function(ergmm_object){
  L = ergmm_object$mcmc.mle$lpY
  llSum = 0
  n = length(ergmm_object$sample)
  for (s in 1:n){
    llSum = llSum + ergmm_object$sample$lpY[s]
  }
  P = 2 * (L - (llSum/n))
  DIC = -2*(L-P)
  DIC
}

# calculate DIC for bergm
# function to calculate likelihood at each posterior bergm sample
ll.bergm <- function(bergm.posterior.sample,bergm.offset.formula,MCMC.samplesize=2000,nsteps=50){
  # bergm.posterior is a matrix with posterior draws in rows and parameters in columns
  # bergm.offset.formula is the formula on which bergm.posterior was inferred, but with
  ## each term enclosed in offset()
  # MCMC.samplesize sets the sample size used to estimate the likelihood
  # nsteps sets the number of bridges used to estimate the likelihood
  require(ergm)
  thetas <- bergm.posterior.sample
  lls <- numeric(nrow(thetas))
  for(i in 1:length(lls)){
    estML <- ergm(bergm.offset.formula,offset.coef=thetas[i,])
    lli <- logLik(estML,add=T,control=control.logLik.ergm(MCMC.samplesize=MCMC.samplesize,nsteps=nsteps)
    lls[i] <- as.numeric(lli$mle.lik)
  }
  thetabar <- apply(thetas,2,mean)
  estML <- ergm(bergm.offset.formula,offset.coef=thetabar)
  llbar <- logLik(estML,add=T,control=control.logLik.ergm(MCMC.samplesize=MCMC.samplesize,nsteps=nsteps)
  list(lls=lls,llbar=llbar)
}

# function to calculate DIC for bergm
dic.bergm <- function(ll.bergm.result){
  Dtbar <- -2*as.numeric(ll.bergm.result$llbar$mle.lik)
  Dbar <- -2*mean(ll.bergm.result$lls)
  2*Dbar-Dtbar
}
```