36-780: Social Network Modeling HW06

Hierarchical Network Models Due Tues 11 Feb 2014

- Reading, etc.:
 - This assignment is based on the following two papers, which you should read [both are available in the "hierarchical network models" area of the bilbiography on the class website]:
 - * Sweet, T. M., Thomas, A. C., and Junker, B. W. (2013) Hierarchical network models for education research: hierarchical latent space models. *Journal of Educational and Behavioral Research*, *33*, 295–318.
 - * Sweet, T. M., Thomas, A. C., and Junker, B. W. (2012). Hierarchical mixed membership stochastic block models for multiple networks and experimental interventions. Invited chapter in: Fienberg, S. E. & Bokalders, K. (eds.) *Handbook on Mixed Membership Models*. Boca Raton, FL: Chapman & Hall/CRC. (to appear, 2014).
 - The software for this assignment is in HNM-code.zip, a copy of which is in the HW06 area of the course website. Please unzip the file into an empty directory, set up an .Rdata file in that directory, and do the computing parts of the assignment in that directory.
- As usual there is nothing to turn in for this assignment. The goal is to create some discussion in class. However if you have some results that are worth sharing (because they are interesting, or confusing, or whatever, feel free to bring those to class to share, or email them to Brian so he can put them up on his laptop in class.
- 1. *Hierarchical latent space model (HLSM)*. Find the file Notes about the HLSM.txt in the directory that you extracted the zip file into, and follow the directions to fit an HLSM model to the Pitts & Spillane data in the Sweet et al. (2013, *JEBS*) paper. The main lines of code to execute are:

source("RunHLSM.r")

```
load("StartingValues.Rdata")
fixed.fit <- HLSM(0,0,advice.mat, edge.vars.mat, priors=0, tune=tune,T=0, 5000)
save(fixed.fit,file="fixed-fit.Rdata")</pre>
```

load("StartingValuesRandom.Rdata")
random.fit <- HLSM(0,1,advice.mat, edge.vars.mat, priors=0, tune=tune,T=0, 5000)
save(random.fit,file="random-fit.Rdata")</pre>

Notes:

- Each run of the HLSM() function will take around an hour to fit the model.
- You may need to load some additional packages from CRAN that this software depends on.
- Once saved, the "fixed.fit" and "random.fit" objects can be reloaded with, e.g. load("fixed-fit.Rdata").
- It may be necessary to run each model fit in a clean R workspace.
- "fixed.fit" treats the coefficients on the observable covariates as fixed effects; in a Bayesian model this means they have very flat priors with no hyperparameters.

• "random.fit" treats the coefficients on the observable covariates as fixed effects; the prior means and variances are estimated from the data, and there is some smoothing/borrowing of information from one network to the next in estimating these coefficients.

The difference between fixed.fit and random.fit is this:

random.fit is fitting exactly model (15) from Sweet et al. (2013, JEBS):

$$\begin{aligned} \text{logit } P[Y_{ijk} = 1] &= \beta_{0k} + X_{1ijk}\beta_{1k} + |Z_{ik} - Z_{jk}| \\ \begin{bmatrix} \beta_{0k} \\ \beta_{1k}^{(1)} \\ \vdots \\ \beta_{1k}^{(d-1)} \end{bmatrix} &\sim N_d \begin{pmatrix} \begin{bmatrix} \mu_0 \\ \mu_1 \\ \vdots \\ \mu_d \end{pmatrix}, \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{d-1}^2 \end{pmatrix} \\ \mu_0, \dots, \mu_{d-1} &\sim N(0, 1) \\ \sigma_0^2, \dots, \sigma_{d-1}^2 &\sim \text{Inv-Gamma (100, 150)} \\ Z_{ik} &\sim N_2 \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{bmatrix} 10 & 0 \\ 0 & 10 \end{pmatrix} \end{pmatrix} \end{aligned}$$

where X_{1iik} has three columns:

experience = 1 if teachers *i* and *j* in network *k* have same (or similar) number of years in school, 0 else **innovators** = 1 if teachers *i* and *j* in network *k* have same beliefs about how innovative they should be, 0 else **grade** = 1 if teachers *i* and *j* in school *k* teach in the same grade, 0 else

fixed.fit is fitting the same model, except that there is only one set of β 's for all 15 networks, and they are given extremely flat normal priors with no hyperparameters (they are treated as fixed effects); consequently in the fixed.fit object, the MCMC chains for μ and σ^2 contain only NA's (this is not a bug but rather reflects that μ and σ^2 are not estimated at all).

Similarly, since there is no network-level treatment in this example, the treatment effect parameter α should be identically zero in both models.

Please also do these things:

(a) The fixed fit and random fit objects contain MCMC posterior samples for each parameter in the respective models. Use these samples to compute posterior means and SEs for the four fixed effects β 's in fixed fit, and for the 15 × 4 β 's in random fit, and compare them. Is there anything interesting to say about the covariates, in either model?

You can use str(fixed.fit), str(random.fit), etc., to see the structure of the lists produced in fixed.fit and random.fit. To do these exercises you will need to access list elements, sublist elements, etc. of the two fitted model objects.

- (b) Both fixed.fit and random.fit provide posterior MCMC samples for the latent space locations of all teachers in all 15 schools in the data. Calculate posterior mean locations for each teacher in each school, under each model.
 - i. Are the locations comparable, under the two models?
 - ii. Are the locations interpretable? If so, give interpretations; if not, explain why not.

Since there are 10 or so of you in the class, it would also be interesting to compare these results across the 10 or so replications of the fits of these models...

2. *Hierarchical mixed membership stochastic block model (HMMSBM)*. Find the file HMMSBM for Intervention.txt in the directory that you extracted the zip file into, and follow the directions to simulate data with an intervention and fit a HMMSBM model to it, as in the Sweet et al (2012) book chapter on the HMMSBM. The main lines of code to execute are:

source("Simulating Intervention Data.r")

```
n <- 10
g <- c(5,5,5,5)
B <- 0
control.lambda <- rep(0.01, 4)
treatment.lambda <- rep(0.1, 4)
test <- Simulate.Int.Data(n, g, B, control.lambda, treatment.lambda)
source("Intervention HMMSBM Running Model.r")
fit <- Run.Int.HMMSBM(test$Y, priors=0, control.lambda=0, n.groups=4,
treatment=c(rep(0,5), rep(1,5)), n.iter=2000)
```

save(fit,file="fit-hmmsbm.Rdata")

Please note

- Run.Int.HMMSBM will take approximately one hour to run.
- You may need to load some additional packages from CRAN that this software depends on.
- The Simulate.Int.Data() function creates, in this case, 10 networks of 20 nodes each, following a MMSBM with four blocks each. The parametrization of the model is similar to equation (4.10) in the Sweet et al. (2012) book chapter.
- The fit object contains posterior MCMC samples for a model similar to equation (4.11) of the Sweet et al. (2012) book chapter.

Please also do the following:

(a) Use the posterior MCMC samples in the fit object to compute posterior means and SEs for the elements of B, θ , and α .

You can use str(fit), etc., to see the structure of the list produced in "fit". To do these exercises you will need to access list elements, sublist elements, etc. of the fitted model object.

- (b) Compare (posterior mean estimates of) S, and R with their simulated "true values" in the test object.
- (c) What can you conclude overall about (i) the recovery of a treatment effect; and (ii) recovery of block memberships?

Again, it would be interesting to compare your results with others' results in the class, as a kind of stability check.