36-780: Social Network Modeling

CID Networks
Brian Junker
132E Baker Hall
brian@stat.cmu.edu

Announcements

- Special presentations (so far):
  - Feb 4: David Krackhardt, on (I think) transitive correlation for measuring transitivity in a graph
  - Feb 13: David Choi, topic TBA (computation and inference for social network models)
  - One or two others still to be announced
- HW05:
  - Propose
    - Two papers to present; or
    - One small project
  - See hw05 online for details

Outline

- Conditionally Independent Dyad (CID) Networks
  - Basic ideas
  - Glm; probit model for edges
  - Directed vs undirected graphs
- Intercept-only model (Erdos-Renyi-Gilbert)
- Other modeling components
  - COV(): edge covariates
  - SR(): p_1-like sender/receiver components
  - LSM(): latent space (distance) model
  - LVM(): latent vector (bilinear form) model
  - SBM(): stochastic block model
- Estimation and Post-processing

Conditionally Independent Dyad (CID) models

- Common form of LSM, SBM models:

$$g(E[Y_{ij}]) = \beta_0 + X_{ij}\beta + U_{ij}$$

  - $Y_{ij} \sim p(y \mid E[Y_{ij}]$, other parameters), independent over ij-pairs
  - g() is the "link" function for a glm
- β_o is an overall intercept
- $X_{ij}\beta$ are edge covariates (that preserve independence of dyads)
- U_{ij} is a random effect, i.e. latent/unobserved structure
  - Allows for some structured dependence across dyads
  - ***Still have $Y_{ij} \amalg Y_{lm} \mid U_{ij}, U_{lm}$ whenever (ij)≠(lm)***

CID Network Models (Thomas et al.)

- ***Valued ties*** with normal distributions,

$$Y_{ij} = \beta_0 + X_{ij}\beta + U_{ij} + \epsilon_{ij},\ \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$$

- ***Dichotomous ties*** with probit distributions,

$$\begin{aligned} Y_{ij} &= \begin{cases} 1\ ,\ Z_{ij} > 0 \\ 0\ ,\ \text{else} \end{cases} \\ Z_{ij} &= \beta_0 + X_{ij}\beta + U_{ij} + \epsilon_{ij},\ \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2) \\ \text{i.e., } \Phi^{-1}(E[Y_{ij}]/\sigma) &= \beta_0 + X_{ij}\beta + U_{ij},\ \Phi() = \text{std normal CDF} \end{aligned}$$

- ***Ordinal ties*** with ordinal probit distributions,

$$Y_{ij} = k,\ \text{iff } c_k \le Z_{ij} < c_{k+1},\ k = 0, \ldots, K-1$$

CIDnetworks is an R package that implements these forms

- Treats all edges as independent
  - No reciprocity effects except through the latent variables U_{ij}, for directed networks
  - Since the U_{ij} are usually symmetric in i,j (e.g. LSM and SBM), probably best-suited to undirected networks
- Designed to handle many kinds of CID model
  - Erdos-Renyi-Gilbert
  - Beta-model (sender-receiver model)
  - Probit regression, LSM, SBM, other latent structures
  - Combinations of these

CIDnetworks main workhorse

```
CID.Gibbs(edgelist=…, sociomatrix=…,
class.outcome=…, outcome=…,
components=list(…))
```

- Specify `edgelist` or `sociomatrix`, ***(not both)***
- `class.outcome:`
  - `“binary”` or missing: binary probit model
  - `“ordinal”:` ordinal probit model
  - `“gaussian”:` normal model
- `outcome:` tie values for “ordinal” or “gaussian”
- `ordinal.count:` how many categories for “ordinal” data
- `components` specifies the kind of model ***(more on following slides)***

CIDnetworks components

```
components = list()
```

- **Empty or missing:** Erdos-Renyi-Gilbert model

$$Z_{ij} = \beta_0 + \epsilon_{ij},\ \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$$

- **components=list(COV(covariates=X))**
  - Each column of X is a covariate; listed in edge-list order 11, 12, …, 1n, 21, 22, …, 2n, …, …, nn

$$Z_{ij} = \beta_0 + X_{ij}\beta + \epsilon_{ij},\ \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$$

CIDnetworks components, cont'd

- **components=list(SR())**
  - Sender-receiver model, aka "beta" model

$$Z_{ij} = \beta_0 + \beta_i + \beta_j + \epsilon_{ij},\ \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2)$$

- **components=list(LSM(dimension=d))**
  - Latent space model of dimension d

$$\begin{aligned} Z_{ij} &= \beta_0 + |u_i - u_j| + \epsilon_{ij},\ \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2) \\ u_i &\overset{iid}{\sim} N_d(0, \Sigma_{d \times d}) \end{aligned}$$

CIDnetworks components, cont'd

- **components=list(LVM(dimension=d))**
  - Latent vector model (latent space model using inner product instead of distance)

$$\begin{aligned} Z_{ij} &= \beta_0 + u_i^{\mathsf{T}} u_j + \epsilon_{ij},\ \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2) \\ u_i &\overset{iid}{\sim} N_d(0, \Sigma_{d \times d}) \end{aligned}$$

- **components=list(SBM(n.groups=k))**
  - Stochastic block model

$$\begin{aligned} Z_{ij} &= \beta_0 + S_i^{\mathsf{T}} B S_j + \epsilon_{ij},\ \epsilon_{ij} \overset{iid}{\sim} N(0, \sigma^2) \\ S_i &\sim \text{multinom}_k(1/k, 1) \\ b_{ij} & \quad \text{estimated from the data},\ i = 1, \ldots, k,\ j = 1, \ldots, k \end{aligned}$$

CIDnetworks components, cont'd

- There are a few other forms that we may consider later
  - MMSBM – mixed membership stochastic block model
  - HBM – Hierarchical block model

- The components can also be mixed, e.g.
  - components=list(COV(covariates=X),LSM(dimension=3))
  - components=list(LSM(dimension=3),SBM(n.groups=5))

Estimation

- Markov Chain Monte Carlo (MCMC)
  - For each parameter β^* or latent variable u^*, randomly sample from the "complete conditional" distribution

$$p(\beta^* | \; \textit{all other parameters, latent variables, \& the data})$$

  - The probit form leads to a particularly straightforward form of MCMC called "Gibbs sampling"
  - Still requires a certain amount of "tuning" and fussing, to fit complex models
  - Generating one set of "draws" like this, one for each parameter and latent variable, is one "Gibbs step".

Post-processing: identifiability heuristics

- **LSM and LVM:** After every Gibbs step,
  - Recenter the u_i so that $\sum_i u_i = 0$
  - Rotate so that node 1 lies on the positive x axis
  - Rotate so that node 2 lies in the x-y plane with positive y-value
  - Repeat the procedure for each of the first d nodes, so that node i lies in the i-dimensional hyperplane with all positive coordinates.
- **SBM:** After every Gibbs step:
  - Permute the block numbers so that block 1 maximizes the probability of membership for node 1
  - Continue permuting so that P[node i is in block i] is maximized, unless node i is already in a lower-numbered block

Other tools in CIDnetworks

- There is a ***plot*** function for CIDnetwork objects
  - Generally plots something appropriate for each modeling component
  - Often includes traceplots of MCMC steps, for visual check of MCMC convergence
- There is a ***CID.generate*** function that randomly generates models of each type (as well as models with more than one component)
- There are several data sets for playing with
  - Lazega (lawyers), c.elegans (worm neurons), dolphins

Summary

- Conditionally Independent Dyad (CID) Networks
  - Basic ideas
  - Glm; probit model for edges
  - Directed vs undirected graphs
- Intercept-only model (Erdos-Renyi-Gilbert)
- Other modeling components
  - COV(): edge covariates
  - SR(): p_1-like sender/receiver components
  - LSM(): latent space (distance) model
  - LVM(): latent vector (bilinear form) model
  - SBM(): stochastic block model
- Estimation and Post-processing

Announcements

- Special presentations (so far):
  - Feb 4: David Krackhardt, on (I think) transitive correlation for measuring transitivity in a graph
  - Feb 13: David Choi, topic TBA (computation and inference for social network models)
  - One or two others still to be announced
- HW05:
  - Propose
    - Two papers to present; or
    - One small project
  - See hw05 online for details