Social Network Analysis – Spring 2022
Lab 2: Exponential Random Graph Models

Deadline: April 2nd at 11:59 pm

Deliverables: a single PDF file. Please include your report along with all the plots, needed tables and diagnostics in this file. If you write the code from scratch and do not use the provided R code, please also submit your code in a separate R file.

In this lab, we will be testing hypotheses about a network's structure using exponential random graph modeling (ERGM) techniques using **statnet** package in R. **statnet** provides a comprehensive framework for ERGM-based network modeling, including tools for model estimation, model evaluation, model-based network simulation, and network visualization. This functionality is powered by a central Markov chain Monte Carlo (MCMC) algorithm.

Statnet documentation / resources: http://csde.washington.edu/statnet/

* We highly recommend you use **the latest version of R**. To check whether your R is the latest, run 'version' so that you'll see a version of R in your console. The latest version of R is available at https://cran.r-project.org/.

Data:
This data is from a project performing people analytics, the use of HR data to understand behavior in an organization. We will analyze data about the social relations between employees, in a small Chinese company. We will be interested in understanding who employees go to for advice.

Our dependent variable is survey responses to the question:

"List up to 5 employees who you rely on the most for help or advice at work.**"**

These responses form a directed network where there is a tie directed from the survey respondent to the person they go to for advice. We provide this data to you in the format of an edgelist, *adviceEdgelist.csv*, in which all 66 employees have been assigned an id ranging from 1 to 66.

Our objective will be to determine what factors influence who someone goes to for advice. We will create a model of this Advice network based on data that is available to the company.

This includes typical HR information, on node attributes, available to them:

- *departmentNode.csv* contains text on which department each employee belongs to.
- *officeNode.csv* contains nodes office locations: 1 for the main office, 0 for the secondary.
- *tenureNode.csv* contains the tenure at the company of each node, in years.
- *leaderNode.csv* contains an indicator (binary) for whether each node is a department lead.
- *femaleNode.csv* contains an indicator variable for sex. 1 for female, 0 for male.

What is unique about this dataset is that we will also look at digital trace data available to the employer. They use an Enterprise Social Media (ESM) platform to allow communication between employees. Enterprise Social Media are social networks developed for work-related use, for example Microsoft Teams or Slack. We have collected data on the number of direct messages employees have sent to one another over roughly six weeks' time. This is shared in the form of an edgelist in *messageEdgelist.csv*.

Hypotheses:
We are going to test the following hypotheses in this lab.

Hypothesis 1: There will be indegree popularity effects – That is, a tendency for a small number of employees to be sought out for advice from many others (as opposed to advice seeking behaviors being spread evenly amongst all employees).

Hypothesis 2: Individuals will be more likely to report go to advice from people in their own department, as opposed to other departments.

Hypothesis 3: There will be homophily based upon the sex of individuals, in terms of who employees go to for advice.

Hypothesis 4*:* Employees who message someone more frequently on ESM will be more likely to report going to that person for advice.

Hypothesis 5*:* If an employee *i* goes to another employee *j* for advice, it will be more likely that *j* also goes to employee *i* for advice.

Hypothesis 6*:* Employees who work in the main office will be more likely to go to others for advice than employees from the secondary office.

Hypothesis 7: Employees who work in the main office will be more likely to be sought after for advice than employees from the secondary office.

Hypothesis 8: Advice seeking relationships tend to be transitive - That is, if individual *i* goes to

an individual *k* for advice, and *k* goes to an individual *j* for advice, then *i* is more likely to go to *j* for advice as well.

These are the hypotheses that we are interested in testing. However, you will notice that the code for ERGMs you have been provided contains more ERGM terms than just the ones corresponding to the above hypotheses. This is normal – in testing the hypotheses of interest, we may still want to include additional model terms to act as controls for other factors.

Your job is to run the code and, most critically, interpret the output. Test the above hypotheses and prepare the report on your work. Include a copy of the relevant output (model parameters, plots) in the pdf/word file for your report.

Part I: Building and Visualizing the Networks (15 pts)

The analysis will use 3 types of files: the "adviceEdgelist.csv" as the *base network* file (the ties the model is predicting), "messageEdgelist.csv" as the *covariate network* file, and the remaining csv files as data on *node attributes*. Download all these files into your working directory and load the network and attribute data.

1. Plot the base (Advice) network and include it in your report. Explain whether this plot *seems*, at a glance, to match what you would expect to see if hypothesis 1 were true. Think of this as just a basic descriptive check – we will perform a more rigorous statistical test in part II of the lab. **(5 pts)**
2. Plot the base network with the nodes now colored based on sex and include it in your report. Explain whether this plot seems to match what you would expect to see if hypothesis 3 were true. **(5 pts)**
3. Plot the covariate network and include it in your report. Comparing the network plots, explain whether this plot seems to match what you would expect to see if hypothesis 4 were true. **(5 pts)**

Part II: Model Estimation (55 pts)

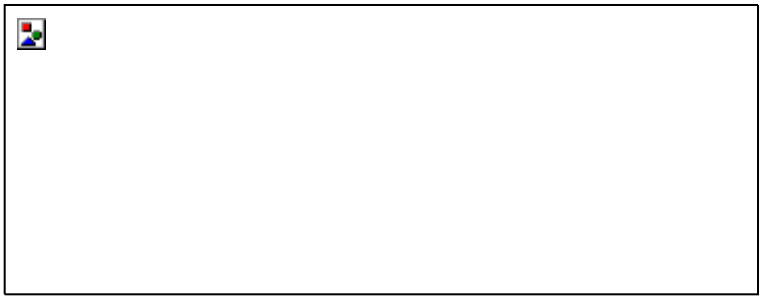
1. Build two ERGM models to test the hypotheses using the different network statistics described below and **include the** results (screenshot of the model output tables from the R console) in your report. Fit model 1 (simple model) and model 2 (complex model) using the terms already specified in the R script provided to you. **(15 pts)**

2. For each of the eight hypotheses, interpret the results from your models and state whether that hypothesis was supported. To determine whether a hypothesis is supported, look at whether there is a p-value < 0.05 and the directionality (positive/negative) of the effect. **When you interpret the results, convert the model coefficients, which are given by R as conditional log-odds, into odds ratios. (40 pts)**

Explanation of Network Statistics used in the Models

Endogenous Effects (Effects of the ties being predicted on other predicted ties)

- ***edges***: number of edges in the network
- ***mutual*****:** number of reciprocal edges in the network
- ***gwidegree****: Geometrically Weighted Indegree*. This term measures a tendency *against* indegree preferential attachment. (Negative coefficients show indegree preferential attachment – Incoming ties are more likely to be directed towards nodes that already have other incoming ties.)
- ***gwodegree****: Directed Geometrically Weighted Outdegree.* This term measures a tendency *against* outdegree preferential attachment. (Negative coefficients show outdegree preferential attachment – Outgoing ties are more likely to originate from nodes that already have other outgoing ties)



- ***dgwesp****, of type "OTP"*: *Directed Geometrically Weighted Edgewise Shared Partners* Number of edges that belong to certain types of triangles. "Edgewise" refers to the fact that we require a tie to exist between nodes i and j, and then measure the number of "shared partners" between them. Shared partners are nodes that have a certain relationship between i and j. In this case, we are looking at the Outgoing Two Path ("OTP") relationships. This is one way to operationalize transitivity. The "geometrically weighted" refers to the fact that we will use a weight parameter, λ, to add diminishing returns to the number of shared partners (i.e., the second shared partner between two nodes will have less effect on the likelihood of a network than the first shared partner, the third will have even less of an effect, and so on).

  Yes, geometrically weighted terms (gwidegree, gwodegree, dgwesp) are very complicated. Essentially, the "geometric weighted" part is saying that effects on network probability have diminishing returns for nodes as degree or the number of shared partners gets higher and higher. This helps avoid model fits where all the ties are directed towards one node. For the purposes of this class, you can ignore the technical details and just focus on interpreting them in terms of "preferential attachment" or "transitivity" effects.

Exogenous Effects (Effects of node attributes or variables outside the predicted ties)

- ***nodeicov***: covariance between in-degree of nodes and attributes of nodes

- ***nodeocov***: covariance between out-degree of nodes and attributes of nodes
- ***diff***: differences between nodes on some numeric attribute (ex. tenure, age). The way we have it specified in the code, diff scores are calculated as the attribute value of the sending node ($attb_i$) minus value of the receiving node ($attb_j$). (Heterophily/ anti-homophily on continuous variables).
- ***nodematch***: tendency of nodes to form ties with those of matching values (Homophily on categorical variables)
- ***nodemix***: mixing matrix of all different combinations of node attributes (ex. A -> A ties, A-> B ties, B -> A ties, B -> B ties). To avoid model overspecification, we need to leave one of these cells out of the model. The weights (effect sizes) estimated for all of the terms we leave in the model then represent the effect of a combination relative to the effect that we left out.
- ***edgecov***: covariance between edges of two networks (the presence/strength of a tie in an outside network on whether a tie exists in our dependent variable network – Advice)

Part III: Model Diagnostics (30 pts)

You can judge convergence of the MCMC process in the models using the `mcmc.diagnostics()` function. The function will plot the change of model statistics during the last iteration of the MCMC estimation procedure. For each model statistic, the left-hand side plot gives the change of the statistic with iterations, and the right-hand side plot is a histogram of the statistic values. Both are normalized, so the observed values are located at 0.

1. Attach the model diagnostics for model 1 and 2 in your report (you should submit a single PDF file) and interpret the plots. Has the MCMC process converged to a desired state? **(10 pts)**
2. Perform Goodness of Fit test to check how well the estimated model captures certain statistical features of the observed network for both model 1 and 2. **(10 pts)**
    a. To do so, simulate many networks from the estimated model and extract 100 samples from the simulation process. Please note, this may take 2 minutes or more to compute.
    b. Extract the number of triangles from each of the 100 samples.
    c. Compare the distribution of triangles in the sampled networks with the observed network by generating a histogram of the triangles. Interpret your result -- is the estimated model a good one in terms of triangle measure?
3. Repeat this goodness-of-fit evaluation process for a variety of other network statistics just for model 2 (for example, degree distribution, distribution of edgewise shared partners, and the distribution of geodesics). Simulate networks as we did above, compile statistics for these simulations as well as the observed network, and calculate p-values of all of the aforementioned values to evaluate the correspondence between the networks simulated by the model and the observed network. Report the p-values for the simulation and interpret them. **(10 pts)**

Note: Your `gof` graph should have 5 subplots corresponding to the 5 parameters of your model. The dark black line represents the data for the observed network. The boxplots represent the distribution of corresponding degrees across the simulated networks, and the soft lines are the 95% confidence intervals. In general, for configurations in the model, the fit is considered good if $|t| \leq 0.1$. For configurations not included in the model, the fit is considered good if $0.1 < |t| \leq 1$, and not extreme if $1 < |t| \leq 2$. If $|t| > 2$ the fit is bad.