November 25, 2019

Social Network Analytics, Empirical Exercise #6
Due on Wednesday, December 18, 2019 at 11:59pm

Network homophily and "failing fast" in Silicon Valley venture capital investing

In the last assignment, we analyzed the influence of investors' status at the firm level on the effectiveness of their diversification strategies. In this exercise, we will analyze the investor network not at the level of the firm, but at the level of the individual.

In an ideal world, investors would select entrepreneurs to invest in based on the merits of their quality and the business model of their startups. In reality, however, investors often choose to invest in for other reasons, such as past relationship history, connections to mutual investors or entrepreneurs, or homophily based on personal attributes like gender, ethnicity, and geographic location.

This homophily can contribute to a culture of "failing fast", in which startup founders can become serial founders as a result of being selected not on their prior performance, but on shared personal or network attributes with investors that might choose to invest in their startups. Failing fast has been described as a way for entrepreneurs to learn more about what makes a successful business model, but does this process actually make their companies more successful? We can analyze the individual-level investor-entrepreneur network an the performance of these companies to understand this feedback loop.

- The file "execs.csv" contains information on startup executives and the startup companies they represent. These are each given by the unique identifiers "PersonId" and "CompanyId."
- The file "individual_investors.csv" contains information on individual investors and the startup companies that they choose to fund.
  - The unique identifiers "PersonId" and "CompanyId" are consistent with the execs file
  - The firm an investor represents is given by the unique identifier "InvestorId"
  - The specific deal that generates the "invests in" relationship between investor and executive is given in "DealId"
- The file "deals_details.csv" contains information on each individual deal, such as its type and size—this is identical to the file with the same name from Exercise 5.
- The file "investor_details.csv" contains information about the investment firms that each individual investor represents, including their geographic location. This is identical to the file with the same name from Exercise #5.
- The file "company_details.csv" contains information about the startup companies each entrepreneur represents, including their industry category. This is identical to the file with the same name from Exercise 5.
- The file "people.csv" contains background information about each person, such as their education and gender.

Illustrating "failing fast" as a form of network homophily:

First, we want to know if homophily on personal or network attributes influences an investor choosing to invest in a particular startup executive. SIENA models allow us to estimate a regression in which the outcome variable is the network itself. We would like to estimate the relationship, "chooses to invest in," which can be represented as a bipartite network from funding PersonIds, investors, to funded PersonIds, executives.

Consider investments to be all deals in the Deal Class "Venture Capital". To make sure that the nodes in the network are likely to be drawn from a more established community, limit the analysis to only deals from the 2000s onward. In order for the network to be bipartite, exclude individuals that appear as both investors and entrepreneurs in the data.

We want to isolate the effects of investor-entrepreneur homophily, as opposed to broader effects of founding team diversity, e.g., having a larger set of initial entrepreneurs involved in a startup. In

order to accomplish this, focus the set of entrepreneurs just on those whose Full Title is indicated to be "Founder", "Chief Executive Officer", or "Managing Director".

Run a SIENA model on five industry categories of your choice, treating each industry as a separate network. Use the classifications given in the Primary Industry Group column for each company to select your industry categories, defining as the network the investors and entrepreneurs for each industry category. To capture the more modern period of investing, only include investment deals that occur within the last twenty years—for some industries that are newer, you can use more recent time periods. Consider each year in the network as a "wave", as in the SIENA example on the teenage networks from class, and consider the investor to be the "ego" sending the investment "tie" and the entrepreneur to be the "alter" receiving the investment tie.

Along with the default rate of network change parameters and parameter for venture capital outdegree that are automatically included in the model, also include in the model predictors for the following types of variables using the `includeEffects()` function.

- Structural predictors, based on network position. These are named parameters in `RSIENA` and can included in the model using the ShortName in typewritten text below—these parameters do not have an interaction in `includeEffects()`.
  - `cycle4`, a measure of closure in bipartite networks that captures network homophily
  - `outActSqrt`, a measure of the dispersion in investors' outdegrees that captures whether investors who fund many entrepreneurs' ventures tend to be more likely to do so going forward
  - `inPopSqrt`, a measure of the dispersion in entrepreneurs' indegrees that captures whether entrepreneurs who are funded by many investors tend to be more likely to receive funding opportunities by more investors going forward
  - `outInAss`, a measure of assortativity, or the correlation of indegrees and outdegrees, that captures whether investors who fund many entrepreneurs' ventures tend to associate with entrepreneurs who receive many funding opportunities from investors
- Dyadic predictors, based on individual attributes. These objects must be designated as constant, time-invariant dyadic covariates to the nodeSets "senders" and "receivers" using `coDyadCovar()`. These covariates should be input in the form of a matrix where the rows and columns match those of the adjacency matrix. These covariates will have an "X" for `interaction1` in `includeEffects()`. Most of the personal attributes such as gender and education can be found in the "people.csv" file.
  - Ethnic homophily, a constant dyadic covariate indicating whether the investor and entrepreneur share the same ethnicity. Because the sample is predominantly white and we want to capture dyads in which ethnic homophily is more salient, only consider as "matched" dyads who share a non-white ethnicity.

    This covariate can be computed as an outer product, e.g., with `outer()`, of the investor and entrepreneur race vectors.

    Background on estimating the ethnicities: the file "representative_names.csv" is drawn from data from the 2015 US Census that indicates, among the 50,000 most common last names, the proportion of each individuals from each distinct ethnic group defined by the census that have this last name. See the link to this data for an example.

    https://names.mongabay.com/data/1000.html

    The first row indicates that Smith is the most common last name in the United States, with 2,442,977 respondents sharing this last name. Of this group, 70.9% are White, 23.11% are Black, 0.5% are Asian or Pacific Islander, 0.89% are American Indian or Alaskan Native, 2.19% are two or more races, and 2.4% are Hispanic.

    From these data, we can define a tolerance threshold for classifying individuals' ethnicities based on their names. For example, a 100% threshold would only classify an individual into a name if all respondents to the Census shared this ethnicity. The data in "representative_names.csv" have scraped the data from the tables in the link and use a 66% threshold, so that if two out of every three respondents with last name identify as an ethnicity, we will classify a person as that ethnicity. You can match the names in "representative_names.csv" to the last names of the individual investors and entrepreneurs to classify their ethnicities.

  - Gender homophily, a constant dyadic covariate indicating whether the investor and entrepreneur share the same gender.

    This covariate can be computed as an outer product, e.g., with `outer()`, of the investor and entrepreneur gender vectors.

  - Top school homophily, a constant dyadic covariate indicating whether the investor and entrepreneur have both earned a degree from a top educational institution. You can include schools in the Ivy League and additional schools such as Caltech, Chicago, MIT, Stanford, and so on, as well as international schools that also represent top schools worldwide such as Cambridge or Oxford in the UK.

    This covariate can be computed as an outer product of a logical vector for investors and entrepreneurs indicating whether they earned a degree from one of these schools.

  - Geographic homophily, a constant dyadic covariate indicating the geographic distance between the headquarters of the investor's investment firm and the headquarters of the entrepreneur's startup. If individuals have multiple locations, you can use the first location ascribed to them in the data.

    This covariate can be computed using `distm()` from the `geosphere` library, applying `fun = distGeo`, where the inputs to `distm()` are the longitude and latitude of the investors' and entrepreneurs' headquarters.

    Background on estimating the distances: a member of last year's MSBA cohort was able estimate the geographic location of each headquarters by using airport data to match the location of each investor firm and startup company's city to the closest airport nearby. The RData file "edges_dist.RData" contains, for each deal in the data, the longitude and latitude of each investor firm, indicated by the column "InvestorId", and the longitude and latitude of each startup company, indicated by the column "CompanyId". These can be merged into the main data using these unique keys. This data can be loaded into the workspace using the `load()` function, which will import an R object called "edges_dist".

  - Experience homophily, a constant dyadic covariate indicating the similarity in years of experience of the investor and entrepreneur. Experience can be measured as the number of years since the investor's first investment or the entrepreneur's first venture received funding.

    This covariate can be computed as an outer product of the difference between the first year of investment or venture founding for each individual, and each individual difference can be subtracted from the maximum difference to turn this into a similarity.

  - Complementary skills, a constant dyadic covariate indicating whether one member of the dyad has a technical degree, such as an engineering degree or a PhD, and the other member of the dyad has a business degree, such as an MBA — part of the rationale for venture capital to entrepreneur matching is that investors can pair their business acumen with the technical skills of innovative entrepreneurs.

    This covariate can be computed as an outer product of a logical vector for investors and entrepreneurs indicating whether they earned one of these types of degrees and another logical vector indicating that they did not earn one of these types of degrees.

- Individual predictors for entrepreneurs, as baseline predictors to isolate the effects of the dyadic predictors. These objects must be designated as constant, time-invariant alter covariates to the nodeSets "receivers" using `coDyadCovar()`. These covariates should be input in the form of a vector where the entries match the ordering of the entrepreneurs in the adjacency matrix. These covariates will have an "altX" for `interaction1` in `includeEffects()`. Most of the personal attributes such as gender and education can be found in the "people.csv" file.

  - Entrepreneur ethnic minority, a constant alter covariate indicating that the entrepreneur has a race that is not White.

    This can be computed as a binary vector.

  - Entrepreneur gender, a constant alter covariate indicating an entrepreneurs' gender.

    This can be computed as a binary vector.

- Entrepreneur top school a constant alter covariate indicating whether an entrepreneur earned a degree from a top educational institution, as described in the previous section.

  This can be computed as a binary vector.
- Entrepreneur geographic hub, a constant alter covariate indicating that the entrepreneur's startup is located in one of the 10 cities that are most common for startups that are invested in to be located in. If individuals have multiple locations, you can use the first location ascribed to them in the data.

  This can be computed as a binary vector.
- Entrepreneur experience, a constant alter covariate indicating the year an entrepreneur's first venture received funding.

  This can be computed as a numeric vector.
- Entrepreneur business skills, a constant alter covariate indicating whether an entrepreneur received an MBA.

  This can be computed as a binary vector.
- Entrepreneur technical skills, a constant alter covariate indicating whether an entrepreneur received a technical degree like a PhD or an engineering degree.

  This can be computed as a binary vector.
- Entrepreneur venture round, a constant alter covariate indicating what round of funding a venture is on, as the cumulative count of unique funding deals it has received from investors.

  This can be computed as a numeric vector.

The SIENA model will want also want to take account of when people join and leave the network. To do this, incorporate using `sienaCompositionChange()` the period, e.g., 1, 2, or 3, corresponding to the first, second, and third year, each investor and entrepreneur enters the dataset. The function `sienaCompositionChange()` takes a list where each element is a vector of length two that takes in the first position the period it enters the data and in the second position the last period in the data.

As in the example from class, you can use `diagonalize = 0.2` to incorporate neighbor effects but also make computations efficient.

If you have access to multiple cores, it can be faster to process the model in parallel. This can also be done via AWS, which you are able to use for this exercise. An example setup for a model using 4 cores would be

```
siena_result = siena07(siena_algorithm, data = siena_data, effects = siena_effects,
nbrNodes = 4, useCluster = TRUE, initC = TRUE)
```

In order to make sure the SIENA model has converged properly, make sure all convergence t-ratios are below 0.1 and the maximum convergence t-ratio is below 0.25. If the convergence t-ratios are not below these thresholds on the first run, you can re-run the algorithm using the parameters estimated in the previous run as starting points with `prevAns = siena_result` in the call to `siena07`.

What do the coefficients estimate for each industry suggest about what factors influence investors' decisions about whether or not to invest in an entrepreneur's startup?

Extra credit: Business implications for homophily and failing fast. (3 points)

Who actually benefits from this practice of encouraging "failing fast"?

Next, we want to understand what failing fast implies for the performance of investors as well as the ventures of entrepreneurs. These data take a long time to clean and compile because we have to calculate, for each investor's and each entrepreneur's first- and second-degree ties, how many cycles of length four exist and what proportion of the ties for each neighborhood exhibit homophily. So, as an early Christmas present, the file "individual_investor_outcomes.csv" already contains all of the cleaned data with this information that can be put directly into a regression. The full code generating this data will be included in the solutions so you can get a sense of the process.

(A) For the first regression, run a regression similar to that of the last question of Exercise #2, in which we predicted whether a venture capital firm would go out of business based on its

network position. Now, we can predict whether the firm goes out of business based on the homophily of investment decisions reflected in its local network. Going out of business is represented by the variable "out_of_business" being equal to 1.

Include as predictors in the model the dyadic and entrepreneur attributes from the SIENA model:

- the scaled number of 4-cycles, given by "l4c_scaled"
- gender homophily, given by "gender"
- ethnic homophily, given by "ethnicity"
- age homophily, given by "age_diff"
- geographic homophily, given by "geo_dist"
- top school homophily, given by "ivyplus"
- complementary skills, given by "complementarity"
- entrepreneur gender, given by "male_exec"
- entrepreneur ethnic minority, given by "nonwhite_exec"
- entrepreneur top school, given by "ivyplus_exec"

Also include as controls the investor longitude and latitude and the year as a linear control.

Does investing based on homophily help investors avoid going out of business?

(B) For the second regression, run a regression similar to that of the last question of Exercise #2, in which we predicted whether a venture capital firm had more successful investments based on its network position. More successful investments are represented as a count in the variable "successful_investments".

Include the same predictors and controls as above, and, since this is a binary outcome, we can include time-averages for each predictor as in the models Exercise #5 to provide investor-level fixed effects.

Does investing based on homophily help investors achieve more successful investments?

(C) For the last regression, we will analyze whether startups benefit from investment based on homophily. So far, we have used regression to predict numerical outcomes. It is also possible to use regression to predict categorical outcomes as well. We will use a "multinomial logit" to predict the likelihood of a startup being in any particular state operation. We have information about six startup states, given in the cleaned data file "startup_states.csv" in the variable "company_state":

(a) Exit: the startup has IPO'd or been acquired for profit — this is the equivalent to the "successful_investments" outcome for the investors

(b) Generating revenue: generating revenue, breaking even

(c) Profitable: generating revenue, fully profitable

(d) Not profitable: losing money

(e) Startup: still in the ramp-up or product generating phase

(f) Out of business: startup has gone out of business or has declared bankruptcy

Set up the multinomial logit using the command `multinom` from the "nnet" package. Include in the model the same predictors and controls as above.

The coefficients in this model represent the likelihood of a startup being in any one of the categories indicated, versus the "reference" category, which acts as the baseline. R interprets the reference category as the first factor level as the reference category. Use as the reference category the company state "startup".

This command does not estimate p-values for statistical significance on its own, so instead these can be estimated by computing the z-scores of the coefficients:

```
z = summary(model)$coefficients/summary(model)$standard.errors
```

and then conducting the significance test:

```
(1 - pnorm(abs(z), 0, 1)) * 2
```

If the value returned is below 0.05, then we can conclude that the predictor was significant.

Does investing based on homophily help entrepreneurs' ventures achieve better outcomes such as Exit or Profitable? What does it seem to suggest about how failing fast through homophily influences the trajectory of a startup?