Workshop on Business Analyses of Networks **Summer 2018**

Lecture 1: Conceptualize Social Networks

Author: Jason (Zhihang) Dong *Date: 06/20/2018*

Note: *LaTeX template courtesy of UC Berkeley EECS dept.*

For this lecture, we will be discussing some key concepts related to social network analysis. Instead of statistics, we will be focusing on some "key terms", which will be frequently referenced in organizational network literature.

1.1 Key Concepts

Definition 1.1 (Relationship) *Relationship is an irreducible property of two or more entities that:*

- *Contrast to properties of entities alone (attributes).*
- *Relations are possibly affected, but not determined, by entity attributes.*

Hence, relational data are collection of entities and a set of measured relations between them. Each, also known as **nodes, actors, egos, units** is connected with zero or more of others via relations, which is also known as **ties, links or edges**. Relations can be directed or undirected; it can also be valued or dichotomous.

Intuitively, we would like to focus on *nodal and dyadic attributes*, many networks exhibit the following features that are very important for organizational network analysis:

- *Reciprocity* of ties: e.g. whether you consider me as your friend if I consider you as my friend.
- *Degree heterogeneity* in the propensity to form or receive ties. This implies sociability and popularity.
- *Homophily by actor attributes*, in which higher propensity to form ties between actors with similar attributes, and attributes may be observed or unobserved
- *Transitivity of relationships*, the pattern that friends of friends have a higher propensity to be friends
- *Balance of relationships*, for example, liking those who dislike whom you dislike
- *Equivalence of nodes*, namely the fact that some nodes may have identical or similar patterns of relationships

1.1.1 Relational Data

As we have discussed, relational data are data that include a set of objects, and measurements between pairs of objects. Relations can be characterized as *undirected* or *directed*.

An **undirected (or symmetric)** relation has only one value per pair:

- $y_{i,j}$ measures the same thing as $y_{j,i}$;
- $y_{i,j}$ is equal to $y_{j,i}$ by design.

A **directed** relation has two values per pair: one value representing the perspective of each pair member.

- $y_{i,j}$ measures something different from $y_{j,i}$;
- $y_{i,j}$ may or may not be equal to $y_{j,i}$.

Relations can be characterized as **binary** or **valued**:

- A binary (or dichotomous) relation takes only two values.
- A valued relation takes more than two values.
  - > A valued relation whose possible values have an order is called ordinal.
  - > A valued relation whose possible values lack an order is called categorical.

Mathematically, these relational data can be represented as a sociomatrix.

Definition 1.2 (sociomatrix) *any relational variable measured on a nodeset can be represented by a square matrix with undefined diagonal entries. A sociomatrix can represent a wide variety of relational data:*

$$Y_1 = \begin{pmatrix} na & a_{1,2} & 3.0 & a_{1,4} \\ a_{2,1} & na & 2.0 & a_{2,4} \\ 1.0 & 2.4 & na & 0 \\ a_{4,1} & a_{4,2} & \cdots & na \end{pmatrix}$$

In a 4 by 4 matrix, we can represent the valued relationship by matrix such as Y_1. Now let's talk about two additional networks commonly seen in organizations. A mono-directional rating is often considered affiliation network, which is formally defined as below:

Definition 1.3 (affiliation network) *Affiliation network is a relational network between two sets of nodes for which there is no overlap between the two node sets and there are no relationships within a node set.*

We usually define **modes** (generally) as non-overlapping groups of objects and **bipartite relation** as a relationship defined on pairs consisting of one member from each of two modes.

Affiliation networks are sometimes referred to as two-mode bipartite networks. Keep in mind, though, They differ from relational data as previously defined:

- within-mode ties are structurally impossible;
- many features of one-mode networks are meaningless for affiliation networks (reciprocity, transitivity);
- they can be treated simply as matrix-valued or multivariate data, for which there is a large and separate statistical literature.

That said, most of the social network analyses were conducted under the assumption of a fully observed binary network.

1.1.2 Fully Observed Binary Network

Let's say binary network is the relational data consisting of a single dichotomous relation, typically taken indicate the presence or absence of a relationship. A fully observed network represents the relationship between each pair of individuals is observed to be either present or absent.

Definition 1.4 (graph) *Formally, a graph consists of a set of nodes* $\mathcal{N} = \{1, \cdots, n\}$ *and a set of edges or lines between nodes* $\mathcal{E} = e_1, \cdots, e_m$ *The graph is denoted* $\mathcal{G} = (\mathcal{N}, \mathcal{E})$.

Each edge $e \in \mathcal{E}$ is expressed in terms of the pair of nodes the line connects. For **undirected graph**,the edges have no direction, and the edge $\{i, j\}$ is the same as the edge $\{j, i\}$:$\{i, j\} = \{j, i\}$ i.e. each edge is an unordered pair of nodes. For **directed graph**,the edges have direction, and the edge (i, j) is not the same as the edge (j, i): $(i, j) \neq (j, i)$. i.e. each edge is an ordered pair of nodes.

Characteristics of Graphs

For an undirected graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$, here are some possible characteristics:

- adjacent : nodes i and j are adjacent if $\{i, j\} \in \mathcal{E}$.
- empty : the graph is empty if $\mathcal{E} = \emptyset$, i.e. there are no edges.
- complete : the graph is complete if $\mathcal{E} = \{\{i, j\} : i \in \mathcal{N}, j \in \mathcal{N}, i \neq j\}$, that is, all possible edges are present.
- incident : node i is incident with edge e if $e = \{i, j\}$ for some $j \in \mathcal{N}$.

Often, we care less about the entire network structure but rather some "interesting points". Here, we introduce the concept of subgraphs.

Definition 1.5 (Subgraph) *The subgraph generated by nodes* $\mathcal{N}_s$ *is the subgraph* $\mathcal{G}_s = (\mathcal{N}_s, \mathcal{E}_s)$ *where* $\mathcal{E}_s$ *includes all edges in* $\mathcal{E}$ *between nodes in* $\mathcal{N}_s$*, namely*

$$\mathcal{E}_s = \mathcal{E} \cap \{\{i, j\} : i \in \mathcal{N}_s, j \in \mathcal{N}_s\}$$

There are some scenarios where node-generated subgraphs are particularly helpful:

- when such a graph is of scientific interest but there is missing data for some nodes, and so we might focus on the subgraph generated by nodes with no missing data.
- when we want to identify cohesive subgroups of nodes, that is, subsets of nodes with a dense node-generated subgraphs.

Definition 1.6 (dyad) *A dyad is a subgraph generated by a single pair* $i, j \in \mathcal{N}$.

In an undirected graph, the possible states of the dyad are given by either $\mathcal{E}_s = \{i, j\}$ or $\mathcal{E}_s = \emptyset$, the empty or complete graphs.

Let $\mathcal{E}_s \subset \mathcal{E}$. The subgraph generated by Es is the subgraph $\mathcal{G}_s = (\mathcal{N}_s, \mathcal{E}_s)$ where $\mathcal{N}_s$ includes all nodes in $\mathcal{N}$ incident with an edge from $\mathcal{E}_s$. These subgraphs may arise in certain types of network sampling schemes, for example, network event data, such as international conflicts, where conflicts are recorded along with the

aggressor and target countries and transactional data, where transactional events are recorded, along with the participating parties.

Note that Edge-generated subgraphs may be misrepresentative of the underlying graph:

$$\{i,j\} \subset \mathcal{N}_s, (i,j) \in \mathcal{E} \implies (i,j) \in \mathcal{E}_s$$

These subgraphs are used less frequently than node-generated subgraphs.

Definition 1.7 (Isomorphic) *Two graphs $\mathcal{G}$ and $\mathcal{G}'$ are isomorphic if there is a 1-1 mapping from nodes of $\mathcal{G}$ to the nodes of $\mathcal{G}'$ that preserves the adjacency of nodes, or equivalently, $\mathcal{G}'$ can be obtained by relabeling the nodes of $\mathcal{G}$.*

Adjacency Matrix

Alternatively, sometimes we consider the structure of a dichotomous network using adjacency matrix. Suppose we have dichotomous (presence/absence) relationship measured between pairs of nodes in a node set $\mathcal{N} = \{1, ..., n\}$. As discussed, such relational data can be expressed as a graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$. The data can also be represented by an $n \times n$ matrix $\mathbf{Y} = \{y_{i,j} : i, j \in \mathcal{N}, i \neq j\}$, where

$$y_{i,j} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{if } (i,j) \notin \mathcal{E} \end{cases}$$

This matrix is called the adjacency matrix of the graph $\mathcal{G} = (\mathcal{N}, \mathcal{E})$.

- The adjacency matrix of every graph is a square, binary matrix with an undefined diagonal.
- Every square, binary matrix with an undefined diagonal corresponds to a graph.

For an undirected binary relation, $\{i,j\} = \{j,i\}$ and so $y_{i,j} = y_{j,i}$ by design.

- the representing graph is an undirected graph;
- the representing adjacency matrix is symmetric.

For a directed binary relation, $(i,j) \neq (j,i)$ and it is possible that $y_{i,j} \neq y_{j,i}$.

- the representing graph is a directed graph;
- the representing adjacency matrix is possibly asymmetric.

The last definition we want to cover is the isolated nodes. Graphs are often stored on a computer in terms of their edge set, or edge list. The **edge list** completely represents the graph unless there are isolated nodes.

Definition 1.8 (Isolated node) *An isolate is a node that is not adjacent to any other node.*

1.2 Centrality

A common goal in SNA is to identify the central nodes of a network. What does central mean?

- active?
- important?
- non-redundant?

Koschutzki et al. (2005) attempted a classification of centrality measures:

- Reach: ability of ego to reach other vertices
- Flow: quantity/weight of walks passing through ego
- Vitality: effect of removing ego from the network
- Feedback: a recursive function of alter centralities

In our notes, We will define and compare four centrality measures:

- degree centrality (based on degree)
- closeness centrality (based on average distances)
- betweeness centrality (based on geodesics)
- eigenvector centrality (recursive: similar to page rank methods)

Definition 1.9 (Node-level indices) *Let $c_1, ..., c_n$ be node-level centrality measures where c_i is centrality of node i by some metric. It is often useful to standardize the c_is by their maximum possible value:*

$$\tilde{c_i} = c_i / c_{\max}$$

Now, let's define a network-level centrality measurement, namely, how centralized is the network? To what extent is there a small number of highly central nodes?

Definition 1.10 (Network-level indices) *Let $c^* = \max\{c_1, \cdots, c_n\}$ Let $S = \sum_i [c^* - c_i]$ Then, we have $S = 0$ if all nodes are equally central; and S is large if one node is most central.*

Therefore, we have

$$C = \frac{\sum_i [c^* - c_i]}{\max \sum_i [c^* - c_i]}$$

Technically, we have $C = 0$ when all nodes have the same centrality; $C = 1$ if one actor has maximal centrality and all others have minimal.

In the next part, we will discuss the four centrality measures one-by-one.

We have the idea that a central actor is one with many connections. Degree centrality is a centrality measure motivated by this idea.

Definition 1.11 (degree centrality) *Given a centrality measure of person i as c_i^d. The undirected degree centrality is calculated by $c_i^d = \sum_{j:j\neq i} y_{i,j}$. The out-degree centrality is calculated $c_i^o = \sum_{j:j\neq i} y_{i,j}$ (namely, how many nodes does the given node "connects to". Then, in-degree centrality is measured by $c_i^i = \sum_{j:j\neq i} y_{j,i}$, namely how many nodes "connect to this node". Finally, we have our standardized degree centrality as:*

$$\tilde{c_i^d} = c_i^d / c_{\max}^d = c_i^d/(n-1)$$

Similarly, we have a graph-level measurement called *centralization*, which corresponds to centrality as a nodal measure. In the example of degree centralization. We define it as below:

Definition 1.12 (degree centralization) *Let actor centrality of a node as c_i^d and the maximum actor centrality observed in the network as c^{d*}. We calculate the centralization using the sum of differences between most central actor and others:*

$$C^d = \frac{\sum_i [c^{d*} - c_i^d]}{\max_Y \sum_i [c^{d*} - c_i^d]} = \frac{\sum_i [c^{d*} - c_i^d]}{(n-1)(n-2)}$$

It is not difficult to imagine that the C^d has its maximum occur when one node has the largest possible degree and the others have the smallest possible degree. This will deduce to a standard "star graph" (Figure 1.1). Here, as a 5-node star graph, all other nodes have a degree centrality of 1, while the central node has 4. So, the numerator should be $(4-1)*4+(4-4) = 3*4$. Imagine the star graph with n nodes, we can summarize that the numerator should be $(n-1)(n-2)$.

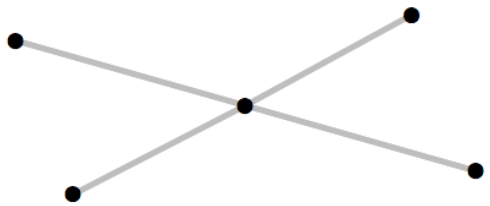

Figure 1.1: Star Graph

Next, inspired by the idea that a central node is one that is close, on average, to other nodes. We define the term closeness centrality.

Definition 1.13 (closeness centrality) *Given $d_{i,j}$ as the minimal path length from i to j, the closeness centrality is measured as:*

$$c_i^c = 1/\sum_{j:j\neq i} d_{i,j} = 1/[(n-1)\bar{d}_i]$$

With nodal level centrality measured, we can progress to the network-level centralization measurement. Here, we use the star graph again in Figure 1.1 as an example: the most central node has a closeness centrality of 1, as it is 1 step away from any other nodes. But all others have a 2-step distance from others, except a 1-step distance to the central node.

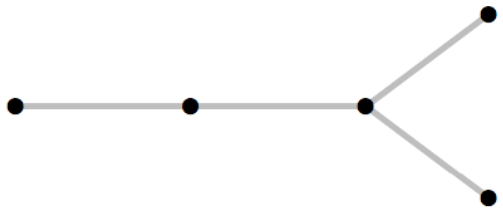

Figure 1.2: "Y" Graph

Definition 1.14 (closeness centralization) *Let actor centrality of a node as c_i^c and the maximum actor centrality observed in the network as c^{c*}. We calculate the centralization using the sum of differences between most central actor and others:*

$$C^c = \frac{\sum_i [c^{c*} - c_i^c]}{\max_Y \sum_i [c^{c*} - c_i^c]} = \frac{\sum_i [c^{c*} - c_i^c]}{(n-2)/(2n-3)}$$

The mathematics of this equation can be easily proven using a n-node star graph example. We will not elaborate here for the sake of conciseness. Note that, for closeness centralization measures, they are *not useful for disconnected graphs.*

Now, motivated by the idea a central actor is one that acts as a bridge, broker or gatekeeper. We want to introduce the idea of betweenness centrality. We know that interaction between unlinked nodes goes through the shortest path (geodesic) and a central node is one that lies on many geodesics.

Definition 1.15 (betweeness centrality) *Let $g_{j,k}$ as the number of geodesics between nodes j and k, and Let $g_{j,k}(i)$ as the number of geodesics between nodes j and k that goes through i, we have betweenness centrality measure as*

$$c_i^b = \sum_{j<k} g_{j,k}(i)/g_{j,k}$$

Here, $g_{j,k}(i)/g_{j,k}$ is the probability that a message from j to k goes through i, where j and k have $g_{j,k}$ routes of communication; i is on $g_{j,k}(i)$ of these routes; and a randomly selected route contains i with probability $g_{j,k}(i)/g_{j,k}$.

Example 1.1 What is the betweenness centrality for each node of the Y-graph as shown in Figure 1.2?

Answer: 0 3 5 0 0

Again, for the network-level measures, the maximum occurs when one node has the largest possible betweenness and the others have the smallest possible betweenness.

Definition 1.16 (betweenness centralization) *Let actor centrality of a node as c_i^b and the maximum actor centrality observed in the network as c^{b*}. We calculate the centralization using the sum of differences between most central actor and others:*

$$C^b(\mathbf{Y}) = \frac{\sum_i [c^{b*} - c_i^b]}{\max_Y \sum_i [c^{b*} - c_i^b]} = 2 * \frac{\sum_i [c^{b*} - c_i^b]}{(n-1)^2(n-2)}$$

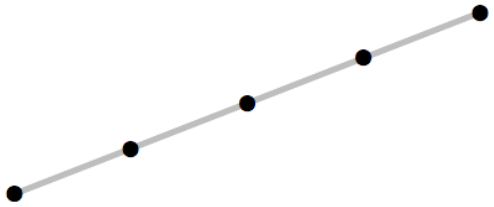

Figure 1.3: Line Graph

Example 1.2 What is the betweenness centralization for the line-graph as shown in Figure 1.3?

Answer: 0.417

The last centrality measure is called eigenvector centrality. Before introducing this concept, I shall introduce the working mechanism of Google PageRank.

Example 1.3 PageRank relies on the uniquely democratic nature of the web by using its vast link structure as an indicator of an individual pages value. In essence, Google interprets a link from page A to page B as a vote, by page A, for page B. But, Google looks at more than the sheer volume of votes, or links a page receives; it also analyzes the page that casts the vote. Votes cast by pages that are themselves important weigh more heavily and help to make other pages important.

This motivates the discussion of eigenvector centrality, in which we have a recursive definition that central actor is connected to other central actors.

Definition 1.17 (eigenvector centrality) *The centrality of each vertex is proportional to the sum of the centralities of its neighbors. Central vertices are those with many central neighbors. Given that the centrality measure is:*

$$c_i^e = \frac{1}{\lambda} \sum_{j:j \neq 1} y_{i,j} c_j^e$$

Using matrix algebra, such a vector of centralities satisfies

$$\mathbf{Y}\mathbf{c}^e = \lambda c^e$$

where the missing diagonal of $\mathbf{Y}$ has been replaced with zeros. A vector $\mathbf{c}^e$ satisfying the above equation is an eigenvector of $\mathbf{Y}$. There are generally multiple eigenvectors. The centrality is taken to be the one corresponding to the largest value of λ. This corresponds with the best rank-1 approximation to Y; Nodes with large c_i^e s have strong activity in the primary dimension of $\mathbf{Y}$.

Example 1.4 What is the eigenvector centrality for each node of the pentagon graph as shown in Figure 1.4?

Answer: All nodes are 0.447.

In the next example, I will show the codes calculating the eigenvector centralization in `R`. You are free to implement it in a simple network structure.

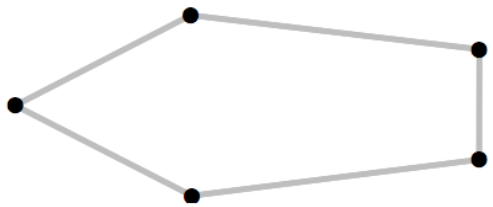

Figure 1.4: Pentagon Graph

Example 1.5 Given the following codes calculating eigenvector centralization for the network, describe the centralization measure mathematically.

```
Ce<-function(Y)
{
n<-nrow(Y)
e<-evecc(Y)
Y.sgn<-matrix(0,n,n) ; Y.sgn[1,-1]<-1 ; Y.sgn<-Y.sgn+t(Y.sgn)
e.sgn<-evecc(Y.sgn)
sum(max(e)-e)/ sum(max(e.sgn)-e.sgn)
}
```