SGL: Stata graph library for network analysis

Hirotaka Miura
Federal Reserve Bank of San Francisco

Stata Conference
Chicago 2011

The views presented here are my own and do not necessarily represent the views of the Federal Reserve System or any person affiliated with it.

What is network analysis?

- Network analysis is an application of network theory, which is a subfield of graph theory, and is concerned with analyzing relational data.
- Some questions network analysis tries to address is how important, or how "central" are the actors in the network and how concentrated is the network.
- Example usages of network analysis include:
  - Determining the importance of a web page using Google's PageRank.
  - Examining communication networks in intelligence and computer security.
  - Solving transportation problems that involve flow of traffic or commodities.
  - Addressing the too-connected-to-fail problem in financial networks.
  - Analyzing social relationships between individuals in social network analysis.

Outline

- Modeling relational data
- Matrix representations
- Centrality measures
- Clustering coefficient
- Stata implementation
- Conclusion

Graph model

- A graph model representing a network $G = (V, E)$ consists of a set of vertices V and a set of edges E.
    - $|V|$ equals the number of vertices.
    - $|E|$ equals the number of edges.
- An edge is defined as a link between two vertices i and j, not necessarily distinct, that has vertex i on one end and vertex j on the other.
    - An edge may be directed or undirected and may also be weighted with differing edge values or have all equal edge values of one in which case the network is said to be unweighted.

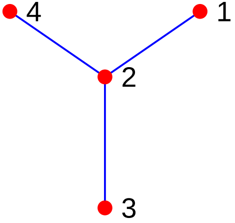
Special cases

- Special types of vertices and edges exist for which standard graph algorithms are not designed to handle or there simply does not exist routines to accommodate such types. Thus the following types of vertices and edges are currently excluded from analysis:
  - Isolated vertex - a vertex that is not attached to any edges.
  - Parallel edges - two or more edges that connect the same pair of vertices.
  - Self-loop - an edge connecting vertex *i* to itself.
  - Zero or negative weighted edge.

Storing relational data

- A variety of storage types are available for capturing relational data:
    - Adjacency matrix
    - Adjacency list
        - Core SGL algorithms use this structure.
    - Edge list
        - Most suited for storing relational data in Stata, as it allows the use of options such as `if` *exp* and `in` *range*.
    - Plus others such as the Compressed Sparse Row format for efficient storage and access.

Example storage types



Undirected unweighted network. Drawn using NETPLOT (Corten, 2011).

Adjacency matrix

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Adjacency list

Vertex	Neighbor(s)		
1	2		
2	1	3	4
3	2		
4	2		

Edge list

$$\begin{bmatrix} 1 & 2 \\ 2 & 3 \\ 4 & 2 \end{bmatrix}$$

Creating an edge list using **joinby** (1)

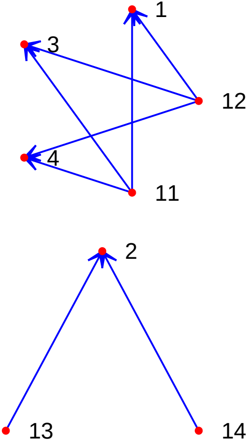
- We illustrate a method of creating an edge list using the **joinby** command and example datasets used in [D] *Data-Management Reference Manual*, **child.dta** and **parent.dta**.
  - Directed edges can represent parent vertices providing care to child vertices.

```
.   use child, clear
(Data on Children)

.   list

     +------------------------------------+
     | family~d   child_id   x1    x2 |
     |------------------------------------|
  1. |     1025          3   11   320 |
  2. |     1025          1   12   300 |
  3. |     1025          4   10   275 |
  4. |     1026          2   13   280 |
  5. |     1027          5   15   210 |
     +------------------------------------+
```

```
.   use parent, clear
(Data on Parents)

.   list

     +------------------------------------+
     | family~d   parent~d   x1    x3 |
     |------------------------------------|
  1. |     1030         10   39   600 |
  2. |     1025         11   20   643 |
  3. |     1025         12   27   721 |
  4. |     1026         13   30   760 |
  5. |     1026         14   26   668 |
     |------------------------------------|
  6. |     1030         15   32   684 |
     +------------------------------------+
```

Creating an edge list using **joinby** (2)

```
. joinby family_id using child

. list parent_id child_id

     +---------------------+
     | parent~d   child_id |
     |---------------------|
  1. |       12          4 |
  2. |       12          1 |
  3. |       12          3 |
  4. |       11          4 |
  5. |       11          3 |
     |---------------------|
  6. |       11          1 |
  7. |       14          2 |
  8. |       13          2 |
     +---------------------+
```



Drawn using NETPLOT (Corten, 2011).

Matrix representations

Adjacency matrix

- Adjacency matrix **A** for unweighted networks is defined as a $|V| \times |V|$ matrix with A_{ij} entries being equal to one if an edge connects vertices i and j and zero otherwise.
- A_{ii} entries are set to zero.
- Matrix **A** is symmetric if the network is undirected.
- For directed networks, rows of matrix **A** represent outgoing edges and columns represent incoming edges.
  - The convention of denoting X_{ij} entries as an edge from i to j is adopted for all matrices.
- For weighted networks, A_{ij} entries are equal to the weight of the edge connecting vertices i and j.

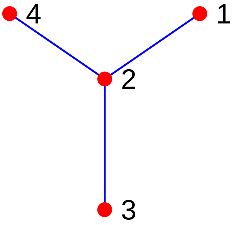
Distance matrix

- Distance matrix **D** is defined as a $|V| \times |V|$ matrix with D_{ij} entries being equal to the length of the shortest path between vertices i and j.
    - A path is defined as a way of reaching vertex j starting from vertex i using a combination of edges that do not go through a particular vertex more than once.
- If no path connects vertices i and j, D_{ij} is set to missing.
    - Signifies what is sometimes referred to as an infinite path.
- D_{ii} is set to zero.
- For undirected networks, matrix **D** is symmetric.

Path matrix

- Path matrix **P** is defined as a $|V| \times |V|$ matrix with P_{ij} entries being equal to the *number* of shortest paths between vertices i and j.
- If no paths exist between vertices i and j, P_{ij} is set to zero.
- P_{ii} is set to one.
- **P** matrix is symmetric for undirected networks.

Example matrices



Undirected unweighted network. Drawn using NETPLOT (Corten, 2011).

Adjacency matrix

$$\begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

Distance matrix

$$\begin{bmatrix} 0 & 1 & 2 & 2 \\ 1 & 0 & 1 & 1 \\ 2 & 1 & 0 & 2 \\ 2 & 1 & 2 & 0 \end{bmatrix}$$

Path matrix

$$\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

Centrality measures

Degree centrality (1)

Undirected network

- Degree centrality measures the importance of a vertex by the number of connection the vertex has if the network is unweighted, and by the aggregate of the weights of edges connected to the vertex if the network is weighted (Freeman, 1978).
- For an undirected network, degree centrality for vertex i is defined as

$$\frac{1}{|V|-1}\sum_{j(\neq i)} A_{ij} \tag{1}$$

where the leading divisor is adjusted for the exclusion of the $j = i$ term.

Degree centrality (2)

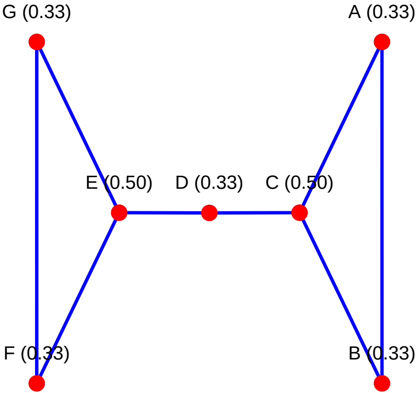
Directed network

- Directed networks may entail vertices having different number of incoming and outgoing edges, and thus we have out-degree and in-degree centrality.
- Out-degree centrality for vertex i is defined similarly to equation (1).
- For in-degree, we simply transpose the adjacency matrix:

$$\frac{1}{|V|-1}\sum_{j(\neq i)} A'_{ij}. \tag{2}$$

Example

Undirected unweighted network



Centrality comparison

Centrality	Vertex		
	A B F G	C E	D
Degree	0.33	0.50	0.33
Closeness			
Betweenness			
Eigenvector			
Katz-Bonacich			

Degree centrality in parentheses. Figure from Jackson (2008). Drawn using NETPLOT (Corten, 2011).

Closeness centrality

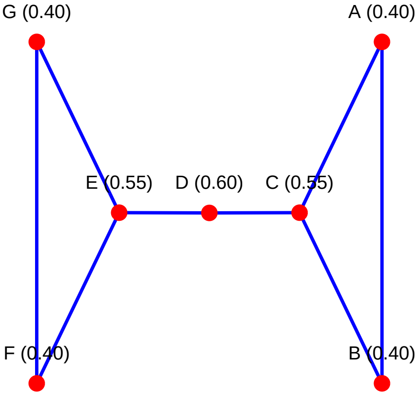
- Closeness centrality provides higher centrality scores to vertices that are situated closer to members of their component, or the set of reachable vertices, by taking the inverse of the average shortest paths as a measure of proximity (Freeman, 1978).
- That is, closeness centrality for vertex i is defined as

$$\frac{(|V|-1)}{\sum_{j(\neq i)} D_{ij}}, \tag{3}$$

which reflects how vertices with smaller average shortest path lengths receive higher centrality scores than those that are situated farther away from members of their component.

Example

Undirected unweighted network



Centrality comparison

Centrality	Vertex		
	A B F G	C E	D
Degree	0.33	0.50	0.33
Closeness	0.40	0.55	0.60
Betweenness			
Eigenvector			
Katz-Bonacich			

Closeness centrality in parentheses. Figure from Jackson (2008). Drawn using NETPLOT (Corten, 2011).

Betweenness centrality

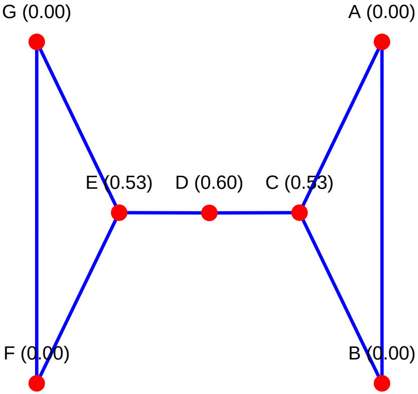
- Betweenness centrality bestows larger centrality scores on vertices that lie on a higher proportion of shortest paths linking vertices other than itself.
- Let P_{ij} denote the number of shortest paths from vertex i to j.
- Let $P_{ij}(k)$ denote the number of shortest paths from vertex i to j that vertex k lies on.
- Then following Anthonisse (1971) and Freeman (1977), betweenness centrality measure for vertex k is defined as

$$\sum_{ij:i\neq j,k\notin ij} \frac{P_{ij}(k)}{P_{ij}}. \tag{4}$$

- To normalize (4), divide by $(|V|-1)(|V|-2)$, the maximum number of paths a given vertex could lie on between pairs of other vertices.

Example

Undirected unweighted network



Centrality comparison

Centrality	Vertex		
	A B F G	C E	D
Degree	0.33	0.50	0.33
Closeness	0.40	0.55	0.60
Betweenness	0.00	0.53	0.60
Eigenvector			
Katz-Bonacich			

Normalized betweenness centrality in parentheses. Figure from Jackson (2008). Drawn using NETPLOT (Corten, 2011).

Eigenvector centrality (1)

- Eigenvector centrality can provide an indication on how important a vertex is by having the property of being large if a vertex has many neighbors, important neighbors, or both (Bonacich, 1972).
- For an undirected network with adjacency matrix **A**, centrality of vertex i, x_i, can be expressed as

  $$x_i = \lambda^{-1} \sum_j A_{ij} x_j \tag{5}$$

  which can be rewritten as

  $$\lambda \mathbf{x} = \mathbf{A}\mathbf{x}. \tag{6}$$

- The convention is to use the eigenvector corresponding to the dominant eigenvalue of **A**.

Eigenvector centrality (2)

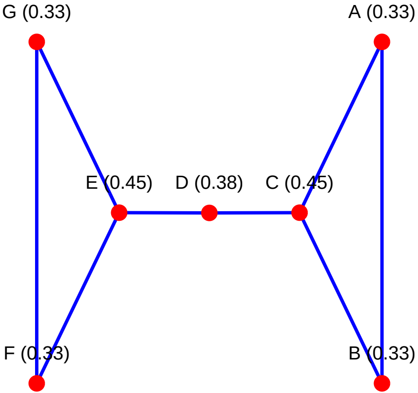
- For directed networks, the general concern is in obtaining a centrality measure based on how often a vertex is being pointed to and the importance of neighbors associated with the incoming edges.
- Thus with a slight modification to equation (6), eigenvector centrality is redefined as

$$\lambda \mathbf{x} = \mathbf{A}'\mathbf{x} \tag{7}$$

  where $\mathbf{A}'$ is the transposed adjacency matrix.
- There are several shortcomings to the eigenvector centrality:
  - A vertex with no incoming edges will always have centrality of zero.
  - Vertices with neighbors that all have zero incoming edges will also have zero centrality since the sum in equation (5), $\sum_j A_{ij} x_j$, will not have any terms.
- The Katz-Bonacich centrality, a variation of the eigenvector centrality, seeks to address these issues.

Example

Undirected unweighted network



Centrality comparison

Centrality	Vertex		
	A		
	B		
	F	C	
	G	E	D
Degree	0.33	0.50	0.33
Closeness	0.40	0.55	0.60
Betweenness	0.00	0.53	0.60
Eigenvector	0.33	0.45	0.38
Katz-Bonacich			

Eigenvector centrality in parentheses. Figure from Jackson (2008). Drawn using NETPLOT (Corten, 2011).

Katz-Bonacich centrality

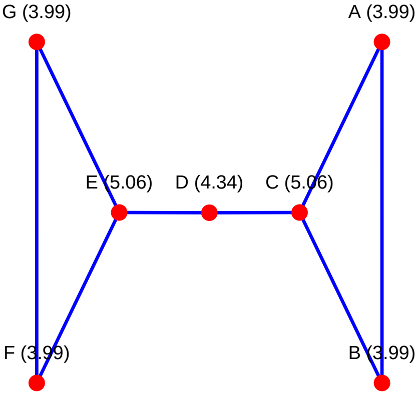
- The additional inclusion of a free parameter (also referred to as a decay factor) and a vector of exogenous factors into equation (7):
  - Avoids the exclusion of vertices with zero incoming edges.
  - Allows connection values to decay over distance.
- Attributed to Katz (1953), Bonacich (1987), and Bonacich and Lloyd (2001).
- Centrality measure is defined as a solution to the equation

  $$\mathbf{x} = \alpha \mathbf{A}' \mathbf{x} + \boldsymbol{\beta} \tag{8}$$

  where α is the free parameter and $\boldsymbol{\beta}$ is the vector of exogenous factors which can vary or be constant across vertices.
- For the centrality measure to converge properly, absolute value of α must be less than the absolute value of the inverse of the dominant eigenvalue of $\mathbf{A}$.
  - A positive α allows vertices with important neighbors to have higher status while a negative α value reduces the status.

Example

Undirected unweighted network



Centrality comparison

Centrality	Vertex		
	A B F G	C E	D
Degree	0.33	0.50	0.33
Closeness	0.40	0.55	0.60
Betweenness	0.00	0.53	0.60
Eigenvector	0.33	0.45	0.38
Katz-Bonacich*	3.99	5.06	4.34

* Maximum $\alpha = 0.43$ (0.33 used). Exogenous factors set to one for all vertices.

Katz-Bonacich centrality in parentheses. Figure from Jackson (2008). Drawn using NETPLOT (Corten, 2011).

Clustering coefficient

Clustering coefficient

- Clustering coefficient is one way of gauging how tightly connected a network is.
- The general idea is to consider transitive relations:
  - If vertex j is connected to vertex i, and i is connected to k, then j is also connected to k.
- Global clustering coefficients provide indication on the degree of concentration of the entire network and consists of overall and average clustering coefficients.
  - Overall clustering coefficient is equal to all observed transitive relations divided by all possible transitive relations in the network.
  - Average clustering coefficient involves applying the definition of overall clustering coefficient at the vertex level, then averaging across all the vertices.

Overall clustering coefficient

- For an undirected unweighted adjacency matrix **A**, overall clustering coefficient is defined as

$$c^o(\mathbf{A}) = \frac{\displaystyle\sum_{i;j\neq i;k\neq j;k\neq i} A_{ji}A_{ik}A_{jk}}{\displaystyle\sum_{i;j\neq i;k\neq j;k\neq i} A_{ji}A_{ik}} \tag{9}$$

  where the numerator represents the sum over *i* of all closed triplets in which transitivity holds, and the denominator represents the sum over *i* of all possible triplets.

Local and average clustering coefficient

- With a slight modification in notation, local clustering coefficient for vertex i is defined as

$$c_i(\mathbf{A}) = \frac{\sum\limits_{j \neq i; k \neq j; k \neq i} A_{ji} A_{ik} A_{jk}}{\sum\limits_{j \neq i; k \neq j; k \neq i} A_{ji} A_{ik}} \tag{10}$$

  which leads to the average clustering coefficient:

$$c^a(\mathbf{A}) = \frac{1}{|V|} \sum_i c_i(\mathbf{A}). \tag{11}$$

- By convention, $c_i(\mathbf{A}) = 0$ if vertex i has zero or only one link.

Generalized clustering coefficient

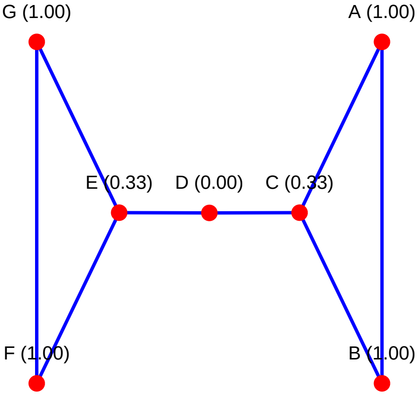
- Building upon the works of Barrat et al. (2004), Opsahl and Panzarasa (2009) propose generalized methods.
- Clustering coefficients for vertex i based on weighted adjacency matrix $\mathbf{W}$ and corresponding unweighted adjacency matrix $\mathbf{A}$ are calculated as

$$c_i(\mathbf{W}) = \frac{\sum\limits_{j \neq i; k \neq j; k \neq i} \omega A_{jk}}{\sum\limits_{j \neq i; k \neq j; k \neq i} \omega} \tag{12}$$

  where ω equals $(W_{ji} + W_{ik})/2$ for arithmetic mean, $\sqrt{W_{ji} \times W_{ik}}$ for geometric mean, $\max(W_{ji}, W_{ik})$ for maximum, and $\min(W_{ji}, W_{ik})$ for minimum.
- For unweighted networks, $\mathbf{W} = \mathbf{A}$ and the four types of clustering coefficients are all equal.

Example

Undirected unweighted network



Average clustering coefficient: 0.67
Overall clustering coefficient: 0.55

Local clustering coefficients in parentheses. Figure from Jackson (2008). Drawn using NETPLOT (Corten, 2011).

Stata implementation

network and **netsummarize** commands

- We demonstrate the use of **network** and **netsummarize** commands on a dataset of 15th-century Florentine marriages from Padgett and Ansell (1993) to compute betweenness and eigenvector centrality measures.
    - **network** generates vectors of betweenness and eigenvector centralities.
    - **netsummarize** merges vectors to Stata dataset.

15th-century Florentine marriages
Edge list stored as Stata dataset

```
     +-----------------------+
     |        v1          v2 |
     |-----------------------|
  1. |   Peruzzi   Castellan |
  2. |   Peruzzi     Strozzi |
  3. |   Peruzzi    Bischeri |
  4. | Castellan     Strozzi |
  5. | Castellan   Barbadori |
     |-----------------------|
  6. |   Strozzi     Ridolfi |
  7. |   Strozzi    Bischeri |
  8. |  Bischeri    Guadagni |
  9. | Barbadori      Medici |
 10. |   Ridolfi      Medici |
     |-----------------------|
 11. |   Ridolfi   Tornabuon |
 12. |    Medici   Tornabuon |
 13. |    Medici     Albizzi |
 14. |    Medici    Salviati |
 15. |    Medici   Acciaiuol |
     |-----------------------|
 16. | Tornabuon    Guadagni |
 17. |  Guadagni   Lambertes |
 18. |  Guadagni     Albizzi |
 19. |   Albizzi      Ginori |
 20. |  Salviati       Pazzi |
     +-----------------------+
```

Computing network centrality measures

```
.   // Generate betweenness centrality.
.   network v1 v2, measure(betweenness) name(b,replace)

Breadth-first search algorithm (15 vertices)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
...............
Breadth-first search algorithm completed

Betweenness centrality calculation (15 vertices)
----+--- 1 ---+--- 2 ---+--- 3 ---+--- 4 ---+--- 5
...............
Betweenness centrality calculation completed
matrix b saved in Mata

.   netsummarize b/((rows(b)-1)*(rows(b)-2)),
> generate(betweenness) statistic(rowsum)


.   // Generate eigenvector centrality.
.   network v1 v2, measure(eigenvector) name(e,replace)
matrix e saved in Mata

.   netsummarize e, generate(eigenvector) statistic(rowsum)
```

Data description (1)

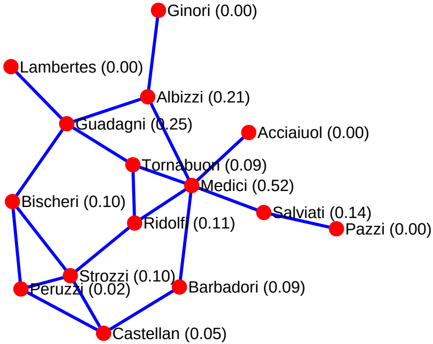
```
Contains data from florentine_marriages.dta
  obs:            20                          15th century Florentine marriages
>  (Padgett and Ansell 1993)
 vars:            10                          10 Dec 2010 09:45
 size:         1,160 (99.9% of memory free)
-------------------------------------------------------------------------------
              storage  display     value
variable name   type   format      label      variable label
-------------------------------------------------------------------------------
v1              str9   %9s
v2              str9   %9s
betweenness_source
                float  %9.0g                  rowsum of Mata matrix
                                                b/((rows(b)-1)*(rows(b)-2))
betweenness_target
                float  %9.0g                  rowsum of Mata matrix
                                                b/((rows(b)-1)*(rows(b)-2))
eigenvector_source
                float  %9.0g                  rowsum of Mata matrix e
eigenvector_target
                float  %9.0g                  rowsum of Mata matrix e
-------------------------------------------------------------------------------
Sorted by:
     Note:  dataset has changed since last saved
```

Data description (2)

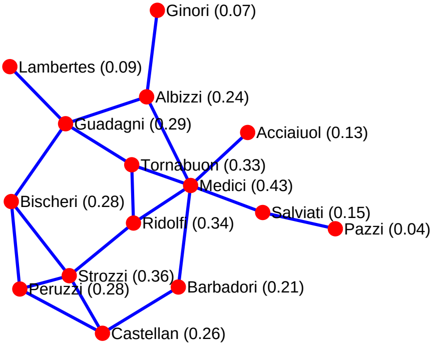
```
.   list v1 v2 betweenness_source betweenness_target eigenvector_source eigenve
> ctor_target

     +-------------------------------------------------------------------------+
     |        v1          v2   betwee~e   betwee~t   eigenv~e   eigenv~t |
     |-------------------------------------------------------------------------|
  1. |   Peruzzi   Castellan    .021978   .0549451   .2757304   .2590262 |
  2. |   Peruzzi     Strozzi    .021978   .1025641   .2757304   .3559805 |
  3. |   Peruzzi    Bischeri    .021978   .1043956   .2757304   .2828001 |
  4. | Castellan     Strozzi   .0549451   .1025641   .2590262   .3559805 |
  5. | Castellan   Barbadori   .0549451   .0934066   .2590262   .2117053 |
     |-------------------------------------------------------------------------|
  6. |   Strozzi     Ridolfi   .1025641   .1135531   .3559805   .3415526 |
  7. |   Strozzi    Bischeri   .1025641   .1043956   .3559805   .2828001 |
  8. |  Bischeri    Guadagni   .1043956   .2545788   .2828001   .2891156 |
  9. | Barbadori      Medici   .0934066    .521978   .2117053   .4303081 |
 10. |   Ridolfi      Medici   .1135531    .521978   .3415526   .4303081 |
     |-------------------------------------------------------------------------|
 11. |   Ridolfi   Tornabuon   .1135531   .0915751   .3415526   .3258423 |
 12. |    Medici   Tornabuon    .521978   .0915751   .4303081   .3258423 |
 13. |    Medici     Albizzi    .521978   .2124542   .4303081   .2439561 |
 14. |    Medici    Salviati    .521978   .1428571   .4303081   .1459172 |
 15. |    Medici   Acciaiuol    .521978          0   .4303081   .1321543 |
     |-------------------------------------------------------------------------|
 16. | Tornabuon    Guadagni   .0915751   .2545788   .3258423   .2891156 |
 17. |  Guadagni   Lambertes   .2545788          0   .2891156   .0887919 |
 18. |  Guadagni     Albizzi   .2545788   .2124542   .2891156   .2439561 |
 19. |   Albizzi      Ginori   .2124542          0   .2439561   .0749227 |
 20. |  Salviati       Pazzi   .1428571          0   .1459172   .0448134 |
     +-------------------------------------------------------------------------+
```

Network visualization

Betweenness centrality



Eigenvector centrality



Centrality scores in parentheses. Drawn using NETPLOT (Corten, 2011).

Conclusion

- Different types of matrices, centrality measures, and clustering coefficients can be generated from information retrieved from relational data.
- The command **network** provides access to SGL functions that generate network measures based on edge list stored in Stata.
- The postcomputation command **netsummarize** allows the user to generate standard and customized network measures which are merged into Stata dataset.
- Future developments include:
  - More efficient algorithms.
  - Designing functions for additional network measures.
  - Optimizing SGL in Mata.

References I

Anthonisse, J. M. (1971): "The rush in a directed graph," Discussion paper, Stichting Mathematisch Centrum, Amsterdam, Technical Report BN 9/71.

Barrat, A., M. Barthélemy, R. Pastor-Satorras, AND A. Vespignani (2004): "The architecture of complex weighted networks," *Proceedings of the National Academy of Sciences*, 101(11), 3747–3752.

Bonacich, P. (1972): "Factoring and weighting approaches to status scores and clique identification," *Journal of Mathematical Sociology 2*, pp. 113–120.

——— (1987): "Power and Centrality: A Family of Measures," *The American Journal of Sociology*, 92(5), 1170–1182.

Bonacich, P., AND P. Lloyd (2001): "Eigenvector-like measures of centrality for asymmetric relations," *Social Networks*, 23(3), 191–201.

Corten, R. (2011): "Visualization of social networks in Stata using multidimensional scaling," *Stata Journal*, 11(1), 52–63.

Freeman, L. C. (1977): "A set of measures of centrality based on betweenness," *Sociometry*, 40(1), 35–41.

——— (1978): "Centrality in social networks: Conceptual clarification," *Social Networks*, (1), 215–239.

Jackson, M. O. (2008): *Social and Economic Networks*. Princeton University Press.

Katz, L. (1953): "A New Status Index Derived from Sociometric Analysis," *Psychometrika*, 18, 39–43.

Opsahl, T., AND P. Panzarasa (2009): "Clustering in weighted networks," *Social Networks*, 31(2), 155–163.

Padgett, J. F., AND C. K. Ansell (1993): "Robust action and the rise of the Medici, 1400-1434," *The American Journal of Sociology*, 98(6), 1259–1319.