SOCIAL NETWORK ANALYSIS USING STATA

5 Sept 2014, Aarhus
Nordic and Baltic Stata Group Meeting

Thomas Grund
Linköping University
thomas.u.grund@gmail.com
www.grund.co.uk

(with contributions from Peter Hedström, Yvonne Aberg, Lorien Jasny)
http://nwcommands.org

Email list

https://groups.google.com/forum/#!forum/nwcommands/join
http://nwcommands.org

Glossary

Below is a an alphabetical list of all nwcommands:

- `_extract_value_labels` extract value labels
- `_nwevalnetexp` evaluates a network expression (length if complex)
- `_nwsyntax` checks network syntax
- `_nwsyntax_other` checks other network syntax
- `_opt_oneof` small utility program for options
- `animate` produces animated-gifs
- `netexample` list of all example networks
- `netlist` concept similar to varlist
- `netname` concept similar to varname
- `nwaddnodes` adds nodes to a network
- `nważsortmix` produces a homophily network
- `nwclear` clears all networks; similar to clear
- `nwclose`ness calculates closeness centrality
- `nwcomponents` calculates number and component memberships
- `nwcompressobs` compresses observations
- `nwcontext` derives attribute values from network neighbors
- `nwcorrelate` correlates two networks or network and attribute
- `nwcurrent` gives information about the current network
GitHub

https://github.com/ThomasGrund/nwcommands
http://nwcommands.org
Getting started
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>[NW-1]</td>
<td>Introduction and concepts</td>
</tr>
<tr>
<td>[NW-2]</td>
<td>Topical list of nwcommands</td>
</tr>
<tr>
<td>[NW-3]</td>
<td>Alphabetical list of nwcommands</td>
</tr>
</tbody>
</table>

*! Date : 3sept2014  
*! Version : 1.0.1  
*! Contact : thomas.u.grund@gmail.com  
*! Web : http://nwcommands.org  
*! Bugs : mailto:bug@nwcommands.org
**Import/Export**

- **nwexport** exports network to pajek
- **nwimport** imports network from other file-formats
- **nwsave** saves network dataset
- **nwuse** uses network dataset
- **nwfromedge** generates network from edgelist
- **nwtoedge** generates edgelist

**Generators**

- **nwassortmix** produces a homophily network
- **nwdyadprob** generates network based on tie probabilities
- **nwexpand** expands attribute as a network
- **nwgenerate** generates network; similar to generate
- **nwgeodesic** calculates geodesic distances
- **nrwrandom** generates random network
- **nwpermute** makes network permutation
- **nwset** sets a network; similar to e.g. stset
- **nwtranspose** transposes a network
. nwuse glasgow

Loading successful
(3 networks)

1) Stored Network

   Network name: glasgow2
   Directed: true
   Nodes: 50

2) Stored Network

   Network name: glasgow1
   Directed: true
   Nodes: 50

3) Current Network

   Network name: glasgow3
   Directed: true
   Nodes: 50
. nwclear
.
. set obs 20
obs was 0, now 20
.
. gen gender = (_n > 10) + 2
.
. nwassortmix gender, density(0.05) homophily(0) name(no_hom)
. nwassortmix gender, density(0.05) homophily(5) name(pos_hom)
. nwassortmix gender, density(0.05) homophily(-5) name(neg_hom)
. nwpplot no_hom, color(gender) layout(circle) title("homophily = 0") saving(g1)
. nwpplot pos_hom, color(gender) layout(circle) title("homophily = 5") saving(g2)
. nwpplot neg_hom, color(gender) layout(circle) title("homophily = -5") saving(g3)
. graph combine g1.gph g2.gph g3.gph
Describe networks
Information

- **nwdyads**: calculates dyad census
- **nwcurrent**: gives information about the current network
- **nwind**: display some network information
- **nwname**: basic network information
- **nwsummary**: some summary information
- **nwtable**: two-way tabulate of two networks or network and attribute
- **nwtabulate**: one-way tabulates tie values of a network
- **nwtriads**: calculates triad census of network
. nwssummary glasgow3

Network name: glasgow3
Network id: 3
Directed: true
Nodes: 50
Arcs: 122
Minimum value: 0
Maximum value: 1
Density: .0497959183673469

. nwtabulate glasgow1

Network: glasgow1   Directed: true

<table>
<thead>
<tr>
<th>glasgow1</th>
<th>Freq.</th>
<th>Percent</th>
<th>Cum.</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>2,334</td>
<td>95.27</td>
<td>95.27</td>
</tr>
<tr>
<td>1</td>
<td>116</td>
<td>4.73</td>
<td>100.00</td>
</tr>
<tr>
<td>Total</td>
<td>2,450</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>
. nwdyads glasgow1

Dyad census: **glasgow1**

<table>
<thead>
<tr>
<th>Mutual</th>
<th>Asym</th>
<th>Null</th>
</tr>
</thead>
<tbody>
<tr>
<td>39</td>
<td>35</td>
<td>1151</td>
</tr>
</tbody>
</table>

. nwtriads glasgow1

Triad census: **glasgow1**

<table>
<thead>
<tr>
<th>003</th>
<th>012</th>
<th>021D</th>
<th>021U</th>
<th>021C</th>
<th>030T</th>
<th>030C</th>
</tr>
</thead>
<tbody>
<tr>
<td>16243</td>
<td>1470</td>
<td>5</td>
<td>18</td>
<td>21</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>120D</td>
<td>120U</td>
<td>120C</td>
<td>111D</td>
<td>111U</td>
<td>201</td>
<td>300</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>2</td>
<td>42</td>
<td>30</td>
<td>15</td>
<td>5</td>
</tr>
</tbody>
</table>
. ntable glasgow1 glasgow2, plot

Network 1:  glasgow1  Directed:  true
Network 2:  glasgow2  Directed:  true

<table>
<thead>
<tr>
<th></th>
<th>glasgow2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0 1 Total</td>
</tr>
<tr>
<td>glasgow1</td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>2,278 59 2,337</td>
</tr>
<tr>
<td>1</td>
<td>56 57 113</td>
</tr>
<tr>
<td>Total</td>
<td>2,334 116 2,450</td>
</tr>
</tbody>
</table>

Maximum: 2328

```
frequency
glasgow1
```

```
glasgow2
0 1
0 2,278 59 2,337 |
1 56 57 113 |
Total 2,334 116 2,450 |
```
```
from_smoke | 1 | 2 | 3 | Total
-----------|---|---|---|------
    1     | 71|  5| 11|  87
    2     |  5|  2|  4|  11
    3     |  7|  5|  3|  15
 Total    | 83| 12| 18| 113
```

```
from_smoke
1 2 3
to_smoke
maximum: 71
glasgow1
```
Manipulate networks
Manipulation

- `nwaddnodes`: adds nodes to a network
- `nwdrop`: drops a network; similar to drop
- `nwdropnodes`: drops nodes from a network
- `nwkeep`: keeps certain networks
- `nwkeepnodes`: keep certain nodes of a network
- `nwreplace`: replaces tie values of a network; similar to replace
- `nwreplacemat`: replaces tie values of a network with a Mata matrix
- `nwerecode`: recodes tie values; similar to recode
- `nwsym`: symmetrizes a network
. nwclear

. nwrandom 7, density(.2) name(first)
. nwrandom 7, density(.3) name(second)
. nwrandom 7, density(.3) name(third)
. gen attr= _n * 2

// replacing networks
. nwreplace first = 1
. nwreplace first = 2 in 3/5
. nwreplace first = exp(second) * attr if first == 1

// replacing subnetworks
. nwreplace first[(2::6),(1::5)] = 55
. nwreplace first[(1::4),(1::4)] = second * 7 if third != 1

// replacing with temporary networks
. nwreplace first =  99 * (_nwrandom 7, prob(.3))
Analyze networks
Analysis

- **nwcloseness**: calculates closeness centrality
- **nwcomponents**: calculates number and component memberships
- **nwcontext**: derives attribute values from network neighbors
- **nwcorsrelate**: correlates two networks or network and attribute
- **nwdegree**: calculates degree centrality
- **nwerpm**: runs exponential random graph model
- **nwevcent**: calculates eigenvector centrality
- **nwneighbor**: derives list of network neighbors
- **nwqap**: network quadratic assignment procedure
- **nwlreach**: calculates reach of a network
- **nwvalue**: returns single tie value
Network permutation

\[
\begin{array}{cccc}
- & 1 & 1 & 0 \\
1 & - & 0 & 1 \\
1 & 0 & - & 0 \\
0 & 0 & 0 & - \\
\end{array}
\]

\[
\begin{array}{cccc}
- & 0 & 0 & 0 \\
0 & - & 0 & 1 \\
1 & 0 & - & 1 \\
0 & 1 & 1 & - \\
\end{array}
\]


`. nwcorsrelate glasgow1 glasgow2, permutation(50)
   .4732457209617567`

`. return list`

**Scalars:**

- \( r(\text{lb}) = -.0490217059850693 \)
- \( r(\text{ub}) = .0426041558384895 \)
- \( r(\text{pvalue}) = 0 \)
- \( r(\text{corr}) = .4732457209617567 \)
- \( r(\text{id}_2) = 1 \)
- \( r(\text{id}_1) = 2 \)

**Macros:**

- \( r(\text{name}_1) : "\text{glasgow1}" \)
- \( r(\text{name}_2) : "\text{glasgow2}" \)
. nwcorsrect glasgow1, attribute(sport1) permutation(50)
   .025768556436961

. return list

scalars:

    r(lb)  =  -.0450700744986534
    r(ub)  =  .0454459525644779
    r(pvalue)  =  .12
    r(corr)  =  .025768556436961
    r(id_2)  =  4
    r(id_1)  =  2

macros:

    r(name_1)  :  "glasgow1"
    r(name_2)  :  "same_sport1"
. **nwqap glasgow3 glasgow2 smoke2 alcohol2, permutations(100) mode(same absdist)**

Permutation: 1 out of 100  
Permutation: 50 out of 100  
Permutation: 100 out of 100

**Multiple Regression Quadratic Assignment Procedure**

<table>
<thead>
<tr>
<th>Estimation</th>
<th>= QAP</th>
<th>Regression</th>
<th>= logit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Permutations</td>
<td>= 100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of vertices</td>
<td>= 50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Number of arcs</td>
<td>= 122</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Coef.</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>glasgow2</td>
<td>3.192734</td>
<td>0</td>
</tr>
<tr>
<td>same_smoke2</td>
<td>.353774</td>
<td>.17</td>
</tr>
<tr>
<td>absdist_alcohol2</td>
<td>-.208237</td>
<td>.13</td>
</tr>
<tr>
<td>_cons</td>
<td>-3.39164</td>
<td></td>
</tr>
</tbody>
</table>
Exponential random graph models

\[ Y_{ij}^c = \text{all dyads other than } Y_{ij} \]

\[
\text{logit}\left[ P(Y_{ij} = 1 \mid n \text{ actors}, Y_{ij}^c ) \right] = \sum_{k=1}^{K} \theta_k \delta s_k(y)
\]

Probability that there is a tie from \( i \) to \( j \).

Given, \( n \) actors AND the rest of the network, excluding the dyad in question!

Amount by which the feature \( s_k(y) \) changes when \( Y_{ij} \) is toggled from 0 to 1.
. nwergm glasgow1, formula(edges() + mutual() + nodematch("smoke1")) gof mcmc

Preparing analysis
(0 observations deleted)
Running ERGM...
C:\R\R-3.0.2\bin\R.exe --slave --silent <ergrcode.r

Exponential random graph analysis
Number of vertices = 50
Number of edges/arcs = 113
Directed = TRUE
Estimation = MLE
Iterations = 20
MCMC sample size = 10000
AIC = 718.74
BIC = 736.15

| network          | Observed | Coef.  | Std.Err. | MCMC% | P>|z| |
|------------------|----------|--------|----------|-------|-----|
| edges            | 113      | -4.309 | .199     | 1     | 0   |
| mutual           | 39       | 4.971  | .373     | 2     | 0   |
| nodematch.smoke1 | 76       | .197   | .157     | 0     | .209 |

Plotting goodness-of-fit statistics
Plotting MCMC-diagnostics
goodness-of-fit

- Percentage of nodes based on indegree and outdegree
- Percentage of nodes based on edge-wise shared partners and minimum geodesic distance

Based on 30 simulations
Visualize networks
Visualization

- animate: produces animated-gifs
- nwmovie: makes movie out of network sequence
- nwplot: plots a network
- scheme-s1network: network scheme1
- scheme-s2network: network scheme2
- scheme-s3network: network scheme3
. nwuse gomery, nwclear
. nwmovie _all, color(col_t*) scheme(s2network)
. `nwuse gang, nwclear`
. `nwplot, color(Birthplace) scheme(s2network)`
. nwplot, size(Arrests, forcekeys(5 15 30)) color(Birthplace) symbol(Prison)
. nwuse florentine, nwclear
. nwplot flobusiness, label(_label) edgecolor(flomarriage)
    edgecolorpalette(yellow red) title("Florentine Businesses", color(red) size(huge))
http://nwcommands.wordpress.com/demo_nwplot

. nwuse glasgow, nwclear
. nwmovie _all
http://nwcommands.org

Thomas Grund
Linköping University

thomas.u.grund@gmail.com
www.grund.co.uk