

# Advanced Network Analysis

## Ego ERGMs

Olga Chyzh [[www.olgachyzh.com](http://www.olgachyzh.com)]

# Recommended Readings

- Salter-Townshend, Michael and Brendan Thomas Murphy. 2015. Role analysis in networks using mixtures of exponential random graph models." *Journal of Computational and Graphical Statistics* 24(2): 520--538.
- Box-Steffensmeier, Janet M., Benjamin W. Campbell, Dino P. Christenson, Zachary Navabi. (2018): Role analysis using the ego-ERGM: A Look at environmental interest group coalitions. *Social Networks* 52: 213--227.
- Campbell, Benjamin W. (2018): Inferring Latent Roles in Longitudinal Networks. *Political Analysis* 26(3): 292--311.

# The Core Idea

- Most current work on clustering nodes in a network focuses on community finding, based on the ratio of between to within linkage densities.
- An Ego-ERGM is a statistical model-based method of clustering nodes based on the role they play in the network.
- For example, identify the most central or high degree nodes, or nodes whose connections are distinct in some other way, i. e. discriminate on different patterns of connectivity.
- Model the set of all ego-networks using ERGMs, then perform a model-based clustering of these ego-networks and, thus nodes within them.
- Simultaneously estimate both the node clustering assignments and the ERGM parameters in the clusters.

# Ego-Networks

Define ego-networks for each node in the network:

1. Select a node; this is referred to as the ego.
2. Include the alters (nodes connected to the ego).
3. Include the connections between the alters.

Nodes performing differing local roles will have markedly different ego-networks and nodes performing similar roles will have similar ego-networks.

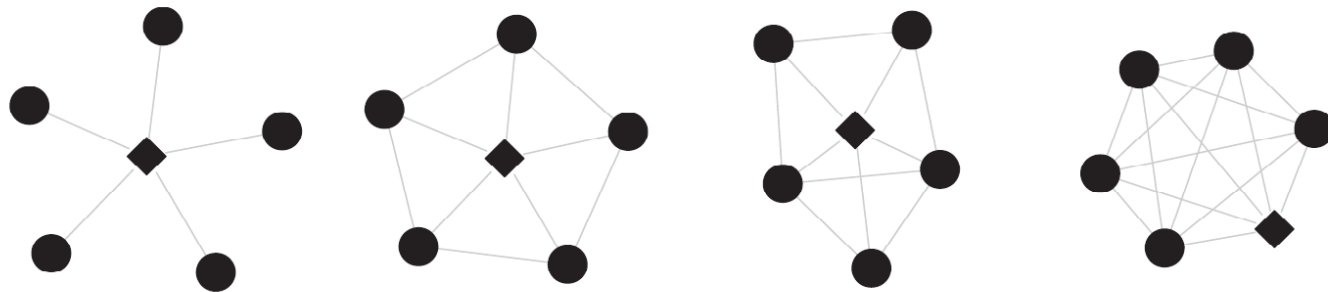
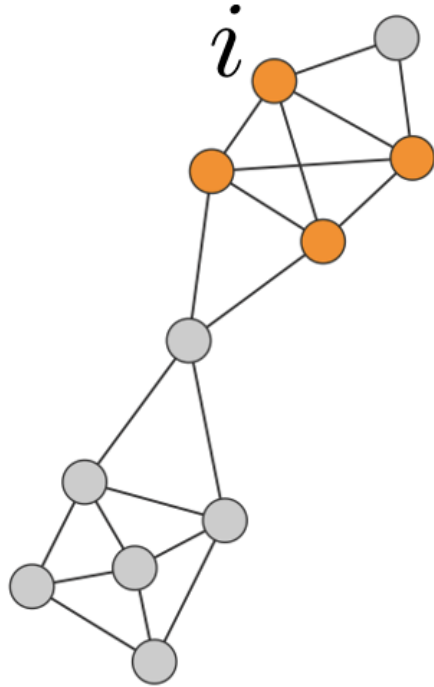


Figure 1. Differing alters' behavior: The ego is square and differing ego roles is illustrated via motifs. In all four cases, the ego has a degree of 5 but the pattern of connectivity between the alters varies.

# Can You Identify Ego Networks?



# Example: Lazega Lawyers

Table 1. Summary of the results for the Lazega Lawyers dataset

	ego-ERGM $_{G=3}$			ego-ERGM $_{G=2}$		Degree	
	$G_1$	$G_2$	$G_3$	$G_1$	$G_2$	Low	High
Partner	6	28	2	28	8	13	23
Associate	20	7	8	7	28	25	10

NOTE: Three models clusterings are compared to the observed status of the lawyers in the network; ego-ERGM with three groups, ego-ERGM with two groups and a descriptive model using degree only.

# Replicate Example from the Article

```
library(network)
library(sna)
library(ergm)
library(devtools)
install_github("ochyzh/networkdata")
library(networkdata)
data(lazega)
```

```
m1 <- ego_ergm(net = lazega,  
              form = c("edges", "mutual",  
                        'desp(.8, type="OSP")',  
                        'desp(.8, type="ISP")'),  
              core_size = 1,  
              min_size = 5,  
              roles = 3,  
              forking = FALSE,  
              ncpus = 1,  
              directed = TRUE,  
              edge_covariates = FALSE,  
              seed = 12345,  
              steps = 50,  
              tol = 1e-06)  
saveRDS(m1, file="data/m1_ego.rds")
```



```
m1<-readRDS("data/m1_ego.rds")
m1$group.theta
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,]  1.740699  1.447842 -0.3398682 -0.1047873
## [2,] -9.542963  1.514716  8.7070389  1.2591454
## [3,]  1.358836  0.928196 -0.4409083 -0.4892306
```

```
m1$role_assignments
```

```
##      Id Role
## 1    V1    1
## 2    V2    3
## 3    V3    3
## 4    V4    1
## 5    V5    1
## 6    V6    3
## 7    V7    1
## 8    V8    1
## 9    V9    1
## 10   V10   3
## 11   V11   1
## 12   V12   1
## 13   V13   1
## 14   V14   3
## 15   V15   1
```

# Replicate the Table

```
length(v.attrs$Status[-58]) #It kicks out node 58 (degree=0?)
```

```
## [1] 70
```

```
table(v.attrs$Status[-58],m1$role_assignments[,2])
```

```
##
```

```
##      1  2  3
```

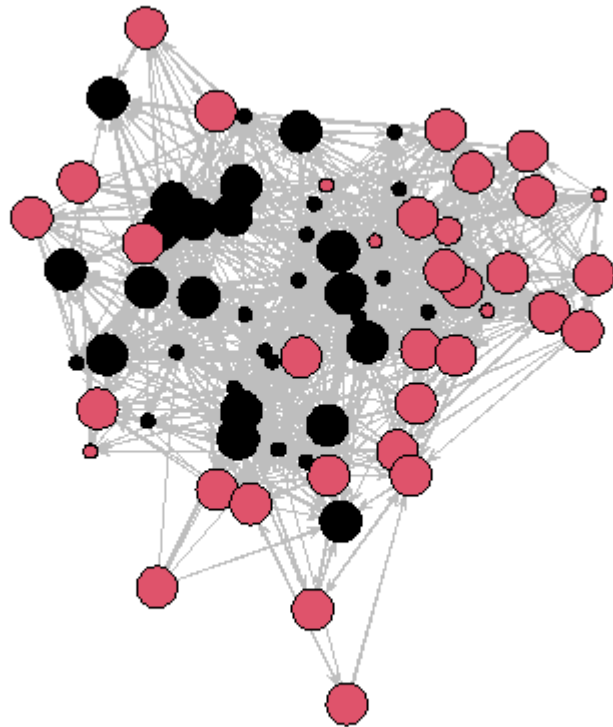
```
##    1 18  0 18
```

```
##    2  4  1 29
```

# Plot

```
plot(lazega,  
      displaylabels = FALSE,  
      # size of nodes based on vector vertexSize  
      vertex.cex = m1$role_assignments[,2],  
      # color of nodes based on vertex attribute: group  
      vertex.col = 'Status',  
      edge.col="gray"  
)
```

# Plot



# Your Turn

1. Load the `sampson` dataset from the `sna` library.
2. How many different roles do you think are in these data?
3. Estimate an ego ERGM on `samplike` data.
4. Cross-tabulate the roles you identified with the `group` vertex attribute. Does it seem like the model is picking up differences based on groups?
5. Cross-tabulate the roles you identified with the `cloisterville` vertex attribute. Does it seem like the model is picking up differences based on whether monks attended Cloisterville?
6. Make a network graph, coloring nodes by role and using node size to show group belonging.