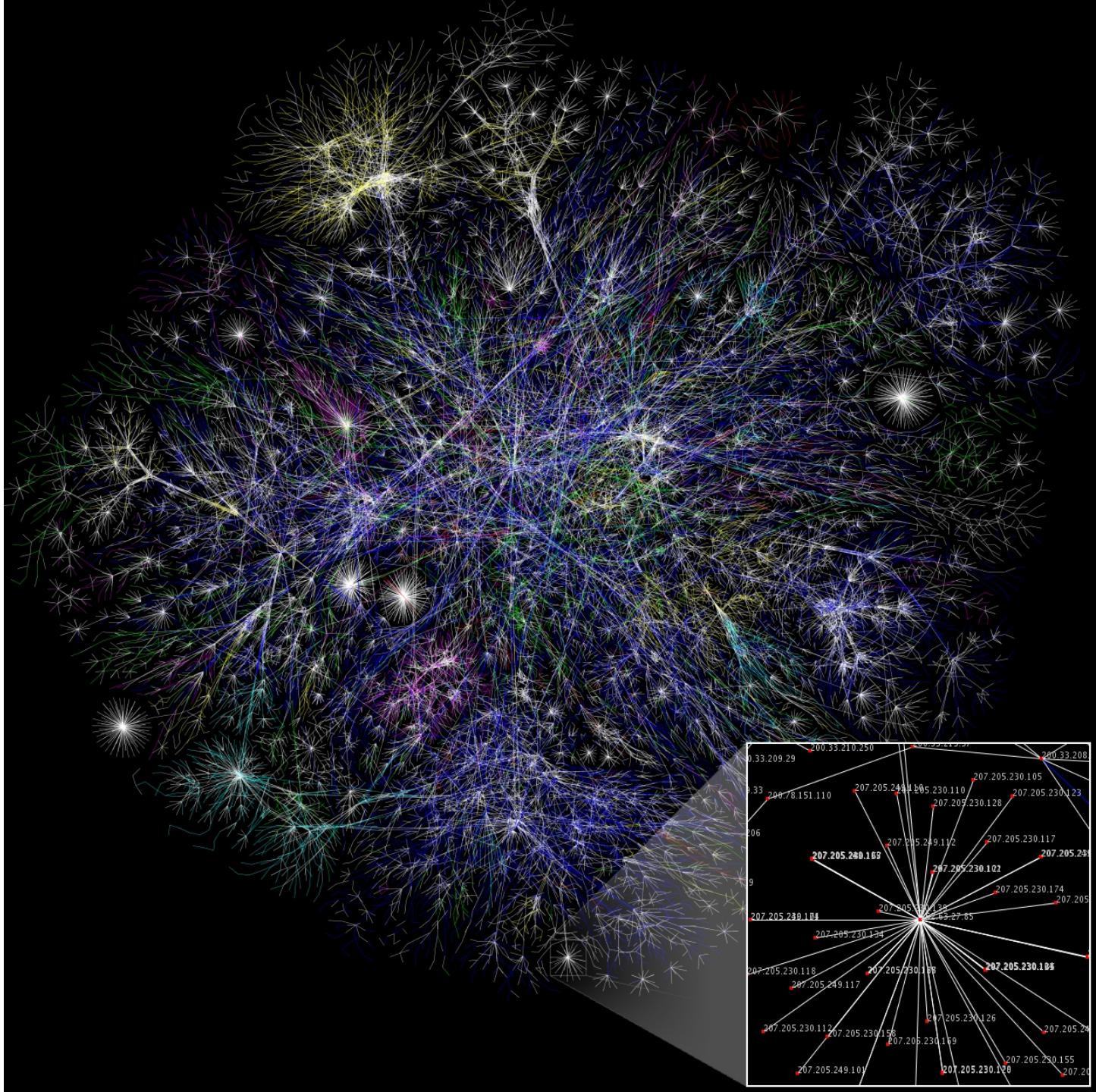
ERGM

Gaurav Bansal, Ph.D.



ERGM

Exponential-family random graph models (ERGMs) represent a general class of models based in exponential-family theory for specifying the probability distribution for a set of random graphs or networks. Within this framework, one can—among other tasks—obtain maximum-likehood estimates for the parameters of a specified model for a given data set; test individual models for goodness-of-fit, perform various types of model comparison; and simulate additional networks with the underlying probability distribution implied by that model.

We will be learning hands-on using two examples:

- one from Harris' *An Introduction to ERGM*
- one from Luke's *A User's Guide to Network Analysis in R*

Let's load the required package **statnet**.

We will be using **lhds** dataset for this analysis.

```
#Install or update statnet
install.packages( 'statnet' )

#Load statnet
library( 'statnet' )

#Load the local health department R data
library(ergmharris) #Data is part of ERGMHARRIS
data( lhds )

#Check that the data loaded properly
lhds

#See a summary of the network and attributes
summary( lhds )
```

LHDS Dataset

We can download the network file **lhds** using the following R commands in **two** different .csv files: one for edges and another one for nodes.

```
#download lhds dataset as two CSV files
library(igraph)
library(intergraph)
class(lhds) #shows it is a network object
lhdsi<-asIgraph(lhds) #converts into igraph object

#Extract data frames with node and edge info
lhdsi.v.df<-as_data_frame(lhdsi,what="vertices")
lhdsi.e.df<-as_data_frame(lhdsi,what="edges")

#Write CSV files
write.csv(lhdsi.v.df,file="lhds_V.csv") #Vertices
write.csv(lhdsi.e.df,file="lhds_e.csv") #Edges
```

LHDS Dataset

- Here we see some of the rows contained in the edges and vertices csv files.
- LHDS contains the United States Local Health Department (LHD) leadership network data.
- The data includes a network object entitled lhds consisting of 1283 Local Health Departments and the communication links between their leaders.
- The network is undirected and ties are present or absent (unweighted).
- Attributes of the network members included in the data include: the state they are located in, whether or not they conduct HIV screening programs or nutrition programs, how many people live in the department jurisdiction, and the number of years of experience the leader has.

Edges

	A	B	C	D	E	F	G	H
1		hivscreen	na	nutrition	popmil	state	vertex.na	years
2	1	Y	FALSE	Y	0.00818	AK	AK005	1
3	2	Y	FALSE	Y	0.17444	AL	AL002	3
4	3	Y	FALSE	Y	0.05744	AL	AL005	1
5	4	Y	FALSE	Y	0.03442	AL	AL009	0
6	5	Y	FALSE	Y	0.01406	AL	AL012	1
7	6	Y	FALSE	Y	0.0263	AL	AL013	1
8	7	Y	FALSE	Y	0.01381	AL	AL014	0
9	8	Y	FALSE	Y	0.0148	AL	AL015	0
10	9	Y	FALSE	N	0.05466	AL	AL017	3
11	10	Y	FALSE	Y	0.01307	AL	AL018	1

Vertices

	A	B	C	D
1		from	to	na
2	1	2	10	FALSE
3	2	2	11	FALSE
4	3	2	19	FALSE
5	4	2	20	FALSE
6	5	5	1003	FALSE
7	6	5	6	FALSE
8	7	6	11	FALSE
9	8	6	17	FALSE
10	9	10	11	FALSE
11	10	11	19	FALSE
12	11	11	26	FALSE
13	12	2	12	FALSE
14	13	6	12	FALSE
15	14	10	12	FALSE

LHDS Dataset

lhds

state: the state where the LHD is located.

tobacconutrition: binary variable indicating whether the LHD does tobacco use preventionnutrition programming (tobacconutrition=Y) or not (tobacconutrition = N).

hivscreen: binary variable indicating whether the LHD does HIV screening (hivscreen = Y) or not (hivscreen = N).

popmil: LHD jurisdiction population in millions.

years: number of years the current LHD leader has been in their position in categories of 1-2 years (years = 0), 3-5 years (years = 1), 6-10 years (years = 2), and 11+ years (years = 3).

The data was collected in 2010 by the National Association of County and City Health Officials (NACCHO). Visit the NACCHO website for additional information about the data source (http://www.naccho.org/).

```
> data(lhds)
> lhds

Network attributes:
 vertices = 1283
 directed = FALSE
 hyper = FALSE
 loops = FALSE
 multiple = FALSE
 bipartite = FALSE
 title = lhds
 total edges= 2708
   missing edges= 0
   non-missing edges= 2708
```

Source: https://cran.r-project.org/web/packages/ergmharris/ergmharris.pdf

Null Model

This is the simplest possible model. There is only one term in the model: edges.

The coefficient of edges is negative (-5.712). This shows that the density of the network is less than 50%.

Coefficient value 0 indicates 50% network density. Hence negative values (i.e. -5.712 indicatethat the network density is less than 50%. Most networks have density less than 50% - so this is not unusual.

```
#Estimating the LHD network
null model

null <- ergm( lhds ~ edges )
summary( null )
```

```
==========================
Summary of model fit
==========================
Formula:   lhds ~ edges
Iterations:  8 out of 20
Monte Carlo MLE Results:
      Estimate Std. Error MCMC % p-value
edges -5.71272    0.01925      0  <1e-04 ***
---
signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
     Null Deviance: 1140093  on 822403  degrees of freedom
 Residual Deviance:   36365  on 822402  degrees of freedom
AIC: 36367    BIC: 36379    (Smaller is better.)
>
```

Null Model

zDensity of the LHDS network is .0032 (as obtained by **`gden(lhds)`** command.

```
> gden(lhds) #.0032
[1] 0.00329279
```

Coefficient for “edges” also provides the same info.

The probability of creating an additional “edge” by adding one more node is:

$$\frac{1}{1+e^{-(Coefficient)}} = \frac{1}{1+e^{-(-5.712)}} = .0032$$

Which happens to be same as graph “density” (i.e. .003).

The probability can also be obtained by using `plogis` command in R.

```
> plogis(coef(model1)['edges'])
      edges
0.00329279
```

```
> plogis(-5.71272)
[1] 0.003292796
```

Visualizing how well null model captures overall network characteristics (such as triangles)

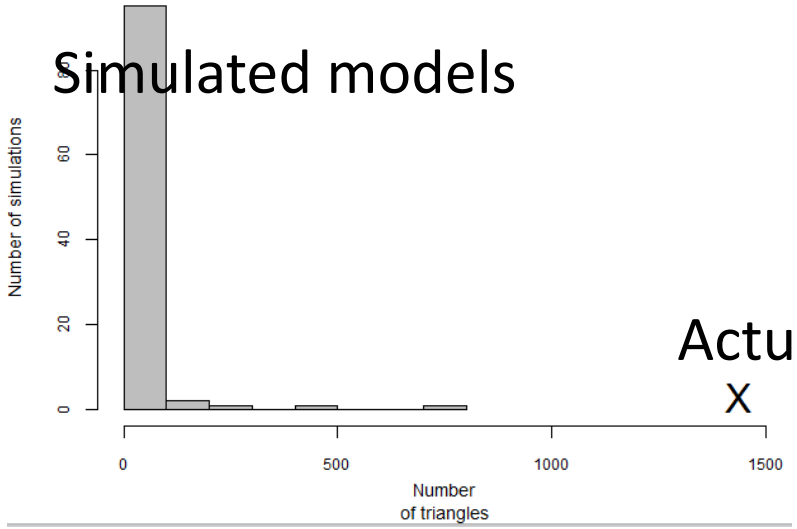
```
#Triangle distribution for simulated networks based on
null model

simtrinull<-simulate( null, nsim = 100,
monitor=~triangles, statsonly=TRUE,
control=control.simulate(MCMC.burnin=1000,
MCMC.interval=1000), seed=567)

lhds.tri <- summary( lhds~triangle )

dev.off()

par( mar = c( 4,4,1,1 ), cex.main = .9, cex.lab =
.9,cex.axis = .75 ) hist(simtrinull[,2], xlim=c(0,1500),
col='gray', main="", xlab="Number of triangles",
ylab="Number of simulations") points(lhds.tri,3, pch="X",
cex=2)
```



We simulate 100 networks based on the NULL model (analysis done with “edges” only). The histogram shows that simulated networks based on simple structure (edges alone) are not able to capture how triangles are formed in the network. There are close to 1500 triangles in the LHDS network, but the simulations based on the NULL model we developed, show very few triangles (as evident by the histogram).

Adding Node Attributes

We created a new model ‘popeffects’. In it we added ‘population size in millions’ to the model (along with ‘edges’).

Low p values for ‘popmil’ shows that population is a significant variable in predicting the connection between two offices “nodes”.

The AIC value of 36255 is smaller than the one obtained for the null model, indicating better fit.

Interpretation: If two LHD offices, have population size of 3 million and 2 million respectively, the likelihood of a having a “link” between the two is:

plogis(-5.787 +.192*3 +.192*2) = .0079

This shows that population of the jurisdiction indicates higher likelihood of the linkage– the density of the LHDS network is .003, so .the likelihood of connection based on “population” is higher than might be expected based on density alone.

```
popeffects <-ergm(lhds ~edges
                  +nodecov('popmil'))
summary(popeffects)
```

```
Summary of model fit
=========================

Formula:   lhds ~ edges + nodecov("popmil")

Iterations:  7 out of 20

Monte Carlo MLE Results:
              Estimate Std. Error MCMC % p-value
edges          -5.78697    0.02055      0  <1e-04 ***
nodecov.popmil  0.19196    0.01457      0  <1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 1140093  on 822403  degrees of freedom
 Residual Deviance:   36251  on 822401  degrees of freedom

AIC: 36255    BIC: 36278    (Smaller is better.)
>
```

```
> gden(lhds) #.0032
[1] 0.00329279
```

```
> plogis(-5.787 +.192*3 +.192*2)
[1] 0.007946858
```

Simulation Analysis and Goodness of Fit

Simulation analysis based on 100 simulations each for NULL model and Population effect model shows that Population Effects model captures more edges, and 1-, 2-, and 5-degree nodes, and also more triangles. But the population effect model is still very far from capturing the “true” number of triangles i.e. 1437.

```
null<-ergm(lhds~edges) #Null Model

nullsim <- simulate(null, verbose = TRUE,seed = 5)
#Simulations based on Null model

mainsim <- simulate(popeffects, verbose = TRUE, seed =
5) #Simulations based on population effect model

rowgof <- rbind(summary(lhds ~ edges + degree(0:5) +
triangle),
summary(nullsim ~ edges + degree(0:5) + triangle),
summary(mainsim ~ edges + degree(0:5) + triangle))

rownames(rowgof) <- c("lhds", "Null", "Population
effects")

rowgof
```

Original network

Simulated networks

	edges	degree0	degree1	degree2	degree3	degree4	degree5	triangle
lhds	2708	58	117	182	223	226	172	1437
Null	2628	21	86	163	253	281	200	9
Population effects	2686	12	92	174	224	251	227	19

Adding Homophily Effect

```
diffhomophily2 <- ergm( lhds ~ edges +
nodecov( 'popmil' ) +
nodefactor( 'years' ) +
nodematch('hivscreen', diff=T, keep=2) +
nodematch('nutrition', diff=T, keep=2) +
nodematch('state') )

summary( diffhomophily2 )
```

Note:
years: number of years the current LHD leader has been in their position in categories of 1-2 years (years=0), 3-5 years (years=1), 6-10 years (years=2), and 11+ years (years=3).

We created another model called **diffhomophily2**. Here we used `nodecov` and `nodematch` terms.

`nodecov` is used to add continuous variables such as population size (other example would be income, height, weight etc.).

`nodefactor` is used for categorical variables such as we have "years" here. Refer to the information on years pasted below here. We have 4 levels for years. ERGM provides coefficients of these nodefactors in relation to the base factor. The base by default is the smallest number, i.e. 0 in this case (experience 1~2 years).

`nodematch` is used for "homophily" where in we match if the two nodes have the same value for the underlying factor. For example here we are asking ERGM to compute probability for two nodes to connect when they both have same value for "HIVScreen" programming i.e. Yes-Yes, or No-No. (diff=TRUE allows for matching both No-No and also Yes-Yes, and keep=2 allows for only Yes-Yes computation).

Homophily Effect

AIC values are still lower, suggesting that it is a better fit so far.

All the terms have significant effect, since they all have low p values (less than .05).

`nodefactor` has three levels, since there are four levels for "years": 0,1,2,3. By default - 0 is not shown here, since it is used as a "base". The coefficient of Years=1~3 are interpreted in relation to year=0.

```
==========================
Summary of model fit
==========================

Formula:   lhds ~ edges + nodecov("popmil") + nodefactor("years") +
    nodematch("hivscreen", diff = T, keep = 2) + nodematch("nutrition",
    diff = T, keep = 2) + nodematch("state")

Iterations:  11 out of 20

Monte Carlo MLE Results:
                      Estimate Std. Error MCMC %  p-value
edges                 -9.55569    0.10996      0  < 1e-04 ***
nodecov.popmil         0.33097    0.02005      0  < 1e-04 ***
nodefactor.years.1     0.17563    0.04748      0 0.000216 ***
nodefactor.years.2     0.32382    0.04443      0  < 1e-04 ***
nodefactor.years.3     0.34631    0.04233      0  < 1e-04 ***
nodematch.hivscreen.Y  0.45866    0.04352      0  < 1e-04 ***
nodematch.nutrition.Y  0.24961    0.04504      0  < 1e-04 ***
nodematch.state        6.31030    0.08411      0  < 1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 1140093  on 822403  degrees of freedom
 Residual Deviance:   19457  on 822395  degrees of freedom

AIC: 19473    BIC: 19566    (Smaller is better.)
>
```

Homophily Effect: Probability Estimation

What is the probability of a connection between two LHD offices if the two nodes have (1) 2 million population, 11 years of experience, State of Wisconsin, Providing HIVscreening, Providing Nutrition programming, and (2) 1.5 million population, 12 years of experience, State of Wisconsin, Providing HIV Screening, Providing Nutrition Programming, respectively?

```
==========================
Summary of model fit
==========================

Formula:   lhds ~ edges + nodecov("popmil") + nodefactor("years") +
    nodematch("hivscreen", diff = T, keep = 2) + nodematch("nutrition",
    diff = T, keep = 2) + nodematch("state")

Iterations:  11 out of 20

Monte Carlo MLE Results:
                      Estimate Std. Error MCMC %  p-value
edges                 -9.55569    0.10996      0  < 1e-04 ***
nodecov.popmil         0.33097    0.02005      0  < 1e-04 ***
nodefactor.years.1     0.17563    0.04748      0 0.000216 ***
nodefactor.years.2     0.32382    0.04443      0  < 1e-04 ***
nodefactor.years.3     0.34631    0.04233      0  < 1e-04 ***
nodematch.hivscreen.Y  0.45866    0.04352      0  < 1e-04 ***
nodematch.nutrition.Y  0.24961    0.04504      0  < 1e-04 ***
nodematch.state        6.31030    0.08411      0  < 1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 1140093  on 822403  degrees of freedom
 Residual Deviance:   19457  on 822395  degrees of freedom

AIC: 19473    BIC: 19566    (Smaller is better.)
>
```

Answer on next slide…

Homophily Effect: Probability Estimation

What is the probability of a connection between two LHD offices if the two nodes have (1) 2 million population, 11 years of experience, State of Wisconsin, Providing HIVscreening, Providing Nutrition programming, and (2) 1.5 million population, 12 years of experience, State of Wisconsin, Providing HIV Screening, Providing Nutrition Programming, respectively?

Answer:

plogis(-9.556+.331*2+.331*1.5+.346

+.346+.459 +.249+.6.310 = .33

```
==========================
Summary of model fit
==========================

Formula:   lhds ~ edges + nodecov("popmil") + nodefactor("years") +
    nodematch("hivscreen", diff = T, keep = 2) + nodematch("nutrition",
    diff = T, keep = 2) + nodematch("state")

Iterations:  11 out of 20

Monte Carlo MLE Results:
                      Estimate Std. Error MCMC %  p-value
edges                 -9.55569    0.10996      0  < 1e-04 ***
nodecov.popmil         0.33097    0.02005      0  < 1e-04 ***
nodefactor.years.1     0.17563    0.04748      0 0.000216 ***
nodefactor.years.2     0.32382    0.04443      0  < 1e-04 ***
nodefactor.years.3     0.34631    0.04233      0  < 1e-04 ***
nodematch.hivscreen.Y  0.45866    0.04352      0  < 1e-04 ***
nodematch.nutrition.Y  0.24961    0.04504      0  < 1e-04 ***
nodematch.state        6.31030    0.08411      0  < 1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 1140093  on 822403  degrees of freedom
 Residual Deviance:   19457  on 822395  degrees of freedom

AIC: 19473    BIC: 19566    (Smaller is better.)
>
```

```
> plogis(-9.556+  .331*2+.331*1.5 +  .346  +  .459 +  .249 +  6.310)
[1] 0.3345894
```

Interpreting Coefficients Associated with Categorical Variables (nodefactor)

We computed Odds Ratios for the coefficients provided for the Nodefactor (Year) items. These odd ratios are interpreted with respect to the reference group for categorical variables.

Interpretation: LHDs offices with more than 11 years of experience are 1.41 times more likely than the ones with 1-2 years of experience to form a tie.

	Lower	OR	Upper
edges	0.0001	0.0001	0.0001
nodecov.popmil	1.3387	1.3923	1.4481
nodefactor.years.1	1.0861	1.1920	1.3083
nodefactor.years.2	1.2671	1.3824	1.5082
nodefactor.years.3	1.3013	1.4138	1.5362

```
#OddsRatio
or <- exp( diffhomophily2$coef )
or #odds ratio
ste <- sqrt( diag( diffhomophily2$covar ))
lci <- exp( diffhomophily2$coef-1.96*ste )
uci <- exp( diffhomophily2$coef+1.96*ste )
oddsratios <- rbind( round( lci,digits = 4 ),round(
or,digits = 4 ),round( uci, digits = 4 ))
oddsratios <- t( oddsratios )
colnames( oddsratios ) <- c( "Lower","OR","Upper" )
oddsratios
```

years: number of years the current LHD leader has been in their position in categories of 1-2 years (years=0), 3-5 years (years=1), 6-10 years (years=2), and 11+ years (years=3).

Goodness of Fit 1

Simulation statistics shows that our diff homophily2 model performed much better than other two models so far.

Overall edges, and number of nodes with one degree, two degree , three degree, four and also five degree were well matched with the original network. Number of triangles is also very similar to the original network.

	edges	degree0	degree1	degree2	degree3	degree4	degree5	triangle
lhds	2708	58	117	182	223	226	172	1437
Null	2628	21	86	163	253	281	200	9
Main effects	2686	12	92	174	224	251	227	19
Diff homophily 2	2679	42	119	195	222	234	171	1123

```
#Comparison of model fit for simulated networks from each model
nullsim <- simulate(null, verbose = TRUE,seed = 5)

mainsim <- simulate(maineffects, verbose = TRUE, seed = 5)

diff2sim <- simulate(diffhomophily2, verbose = TRUE, seed = 5)

rowgof <- rbind(summary(lhds ~ edges + degree(0:5) + triangle),
                summary(nullsim ~ edges + degree(0:5) + triangle),
                summary(mainsim ~ edges + degree(0:5) + triangle),
                summary(diff2sim ~ edges + degree(0:5) + triangle) )

rownames(rowgof) <- c("lhds", "Null","Main effects","Diff homophily 2")

rowgof
```

Goodness of Fit 2

Lower AIC values indicate better fit.

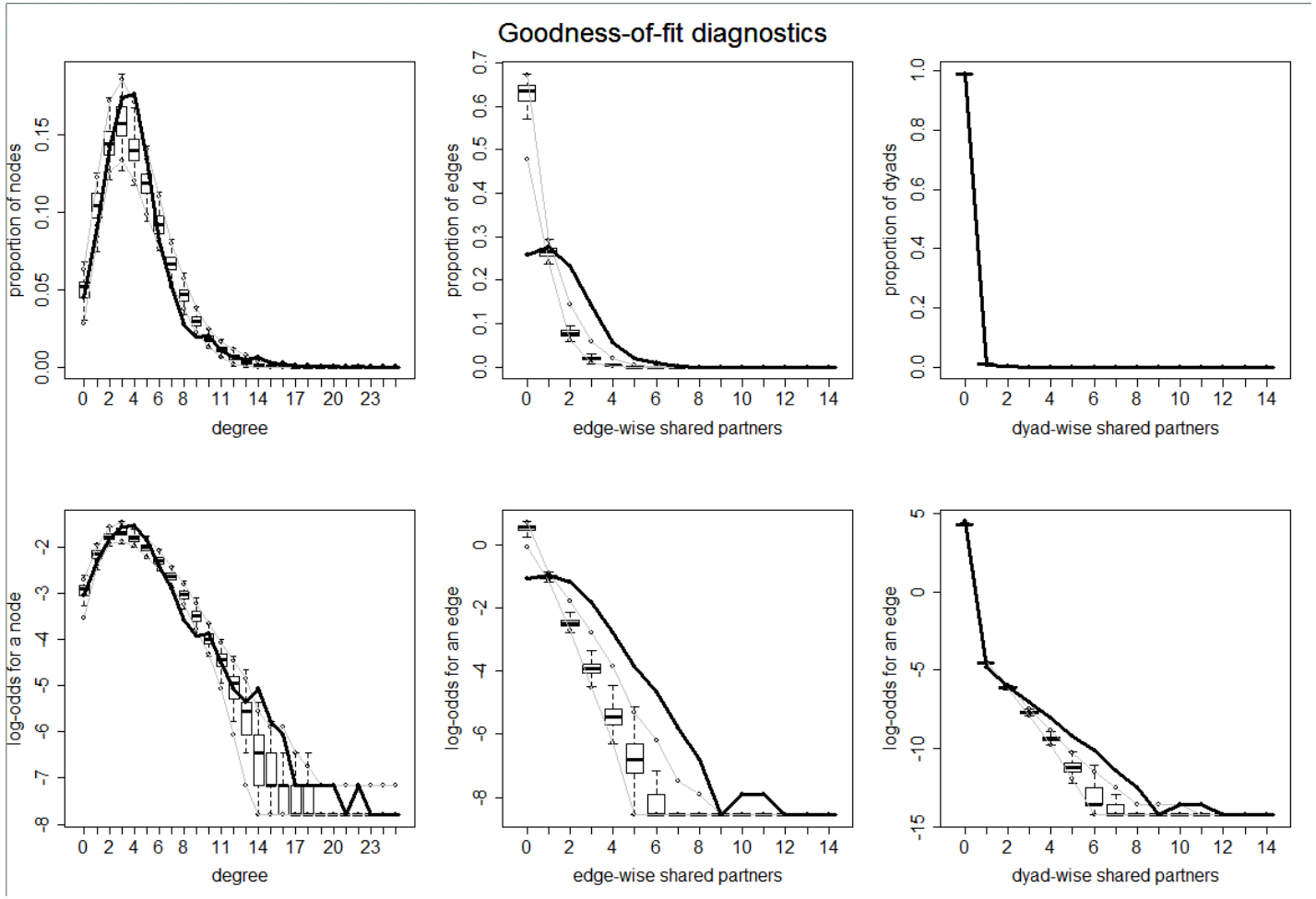
AIC is a relative term, and is useful in comparing two models. It does not reflect on how good the model actually is (by itself).

Goodness of Fit 3

Good of fit plots: the dark line needs to be within the gray lines to indicate goodness of fit.

Gray lines indicate the confidence interval.

The plot shows that degree terms have been modeled well, but there are some issues with edge-wise shared partners, and dyad-wise shared partners.



Goodness of Fit 4

This command would generate the plots, as well as the p value table as discussed later.

```
#Goodness-of-fit simulations for the diffhomophily2 model

diff2_gof <- gof( diffhomophily2, GOF =  ~degree + espartners +
                  dspartners, verbose = T, burnin = 10000,
                  interval = 10000, seed = 567 )
diff2_gof
```

```
#Graphic goodness-of-fit for the diffhomophily2 network
dev.off()
par( mfrow = c( 2,3 ) )
plot( diff2_gof, cex.lab = 1.5, cex.axis = 1.5 )
plot(diff2_gof, cex.lab = 1.5, cex.axis = 1.5, plotlogodds = T )
```

Goodness of Fit 5

High p value for “goodness-of-fit for degree” shows that the there are no significant differences when it comes to number of isolates (0 degree) between the simulated networks and the original network. And same can be said for all degree nodes from 0 to 34 (except for nodes with degrees 4,7,8, and 9, as they all p values less than .05).

The fact that majority of cases have high p value shows goodness of fit atleast from “degree” perspective.

Goodness-of-fit for degree

	obs	min	mean	max	MC p-value
0	58	30	63.67	87	0.52
1	117	96	133.56	161	0.32
2	182	155	186.63	234	0.88
3	223	151	203.45	243	0.34
4	226	151	182.05	235	0.04
5	172	121	153.00	188	0.10
6	104	80	118.75	153	0.28
7	67	66	85.86	106	0.02
8	35	44	59.72	78	0.00
9	25	25	38.06	59	0.02
10	26	11	23.61	37	0.54
11	14	6	14.65	23	0.96
12	8	1	8.60	16	0.98
13	6	0	4.60	12	0.62
14	8	0	2.40	6	0.00
15	4	0	1.39	4	0.06
16	3	0	0.97	4	0.10
17	1	0	0.55	3	0.82
18	1	0	0.33	2	0.60
19	1	0	0.23	2	0.44
20	1	0	0.13	1	0.26
21	0	0	0.09	1	1.00
22	1	0	0.12	1	0.24
23	0	0	0.09	2	1.00
24	0	0	0.10	1	1.00
25	0	0	0.07	1	1.00
26	0	0	0.06	1	1.00
27	0	0	0.08	1	1.00
28	0	0	0.08	1	1.00
29	0	0	0.03	1	1.00
30	0	0	0.03	1	1.00
31	0	0	0.03	1	1.00
34	0	0	0.01	1	1.00

Another Example

From *A User's Guide to Network Analysis in R*

(TCnetworks / TCdiss dataset)

```
==========================
Summary of model fit
==========================

Formula:   TCdiss ~ edges + nodecov("tob_yrs") +
    nodematch("agency_lvl", diff = TRUE) + gwesp(0.7, fixed =
    TRUE)

Iterations:  2 out of 20

Monte Carlo MLE Results:
                      Estimate Std. Error MCMC %  p-value
edges                 -5.21778    0.67546      0  < 1e-04 ***
nodecov.tob_yrs        0.07627    0.01910      0  < 1e-04 ***
nodematch.agency_lvl.1 1.24233    0.41789      0  0.00319 **
nodematch.agency_lvl.2 0.15928    0.32144      0  0.62060
nodematch.agency_lvl.3 1.01011    0.35413      0  0.00465 **
Gwesp.fixed.0.7        1.36997    0.32622      0  < 1e-04 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

     Null Deviance: 415.9  on 300  degrees of freedom
 Residual Deviance: 311.2  on 294  degrees of freedom

AIC: 323.4    BIC: 345.4    (Smaller is better.)
```

```
library(UserNetR)
library(statnet)
data(TCnetworks)

class(TCnetworks) #it is a list object, contains several (four) networks

summary(TCnetworks)

TCdiss<-TCnetworks$Tcdiss   #Tcdiss network is part of Tcnetworks list object

class(TCdiss)



############################

DSmod4g<-ergm(TCdiss~edges +
              nodecov('tob_yrs')+
              nodematch('agency_lvl',diff=TRUE)
+
              gwesp(0.7, fixed=TRUE), #gwesp term is added to account for transitivity effect
              control=control.ergm(seed=40)
              )
summary(DSmod4g)
```

Goodness of Fit Evaluation

High p values indicate good fit between the simulated and the original network

```
DSmod.fit<-gof(DSmod4g, GOF =
~ distance+ espartners +
degree + triadcensus)
summary(DSmod.fit)
```

```
#Four Plots
op<-par(mfrow=c(2,2))
plot(DSmod.fit,cex.axis=1.6,ce
x.label=1.6)
par(op)
```

Goodness-of-fit for degree

	obs	min	mean	max	MC p-value
0	0	0	1.15	4	0.54
1	1	0	0.40	3	0.60
2	2	0	0.59	4	0.26
3	3	0	0.71	3	0.04
4	2	0	1.65	6	1.00
5	1	0	1.79	5	0.96
6	2	0	2.25	7	1.00
7	2	0	2.72	6	1.00
8	0	0	2.49	5	0.06
9	1	0	2.36	7	0.52
10	3	0	2.02	6	0.72
11	2	0	1.88	5	1.00
12	1	0	1.41	5	1.00
13	1	0	0.97	4	1.00
14	2	0	0.83	4	0.44
15	1	0	0.67	3	0.96
16	0	0	0.42	2	1.00
17	0	0	0.30	2	1.00
18	0	0	0.23	2	1.00
19	0	0	0.08	1	1.00
20	0	0	0.04	1	1.00
21	0	0	0.02	1	1.00
22	0	0	0.02	1	1.00
24	1	0	0.00	1	0.00

Goodness-of-fit for triad census

	obs	min	mean	max	MC p-value
0	832	482	740.79	1061	0.50
1	759	829	922.95	1027	0.00
2	517	281	483.66	675	0.70
3	192	91	152.60	237	0.18

Goodness-of-fit for minimum geodesic distance

	obs	min	mean	max	MC p-value
1	103	74	102.09	129	1.00
2	197	111	155.69	188	0.00
3	0	0	14.28	56	0.02
4	0	0	0.85	18	1.00
5	0	0	0.05	3	1.00
6	0	0	0.01	1	1.00
Inf	0	0	27.03	90	0.54

Goodness-of-fit for edgewise shared partner

	obs	min	mean	max	MC p-value
esp0	1	0	0.61	4	0.76
esp1	8	0	3.55	9	0.10
esp2	10	4	11.82	23	0.84
esp3	7	9	20.00	31	0.00
esp4	15	10	20.67	32	0.32
esp5	11	6	17.18	31	0.20
esp6	9	3	12.15	25	0.74
esp7	14	0	7.43	21	0.18
esp8	14	0	4.44	10	0.00
esp9	5	0	2.24	8	0.26
esp10	4	0	1.09	4	0.06
esp11	1	0	0.59	2	0.86
esp12	1	0	0.20	2	0.36
esp13	2	0	0.06	1	0.00
esp14	1	0	0.04	1	0.08
esp15	0	0	0.01	1	1.00
esp15	0	0	0.01	1	1.00

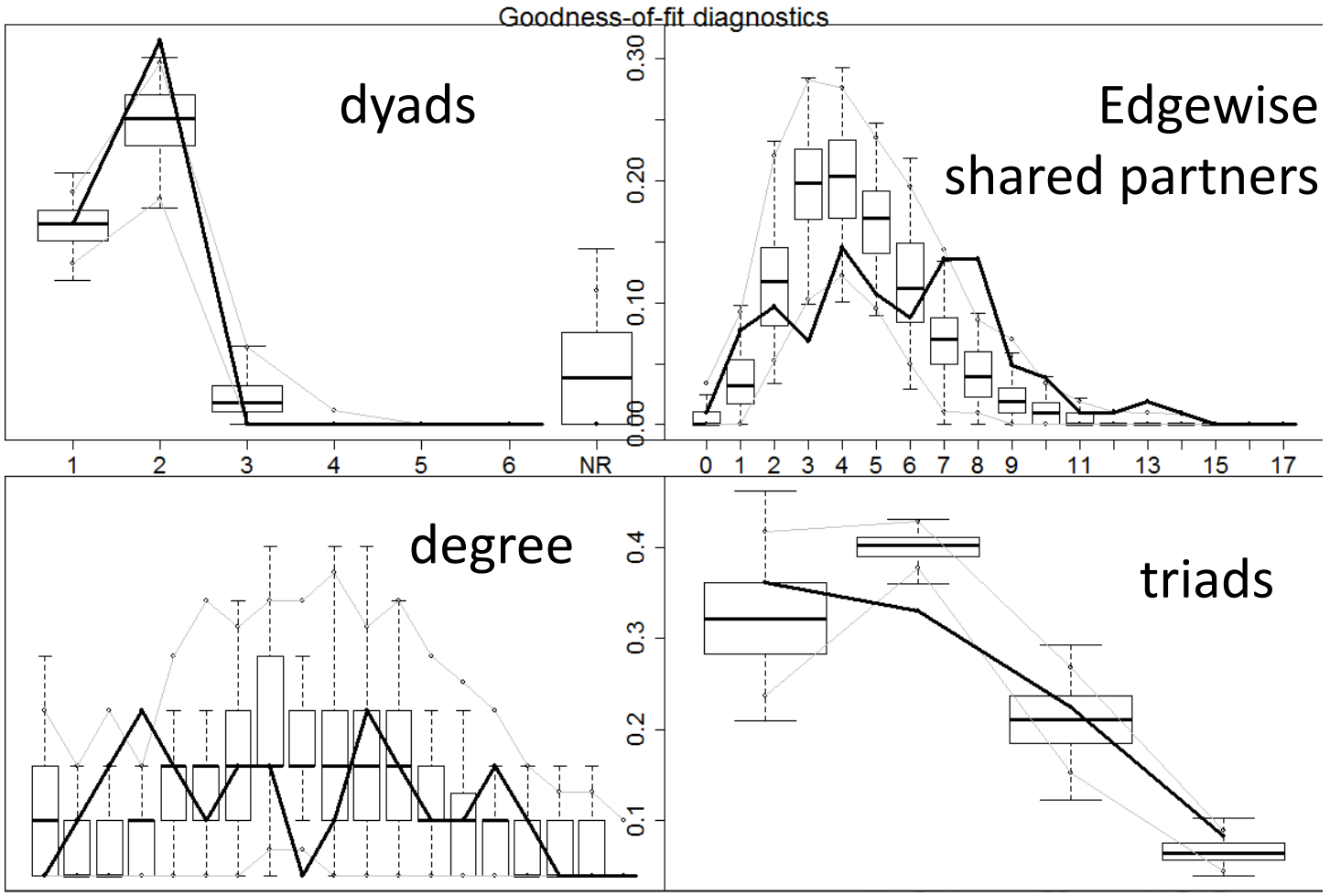
Goodness of Fit Evaluation

The fact that dark line
is within the confidence interval bands (grey lines) indicates good fit for all the four dimensions: dyads, triads, degree, and edgewise shared partners.

This is a reflection of high p values we just saw in the previous summary slide.

```
DSmod.fit<-gof(DSmod4g, GOF =
~ distance+ espartners +
degree + triadcensus)
summary(DSmod.fit)
```

```
#Four Plots
op<-par(mfrow=c(2,2))
plot(DSmod.fit,cex.axis=1.6,ce
x.label=1.6)
par(op)
```

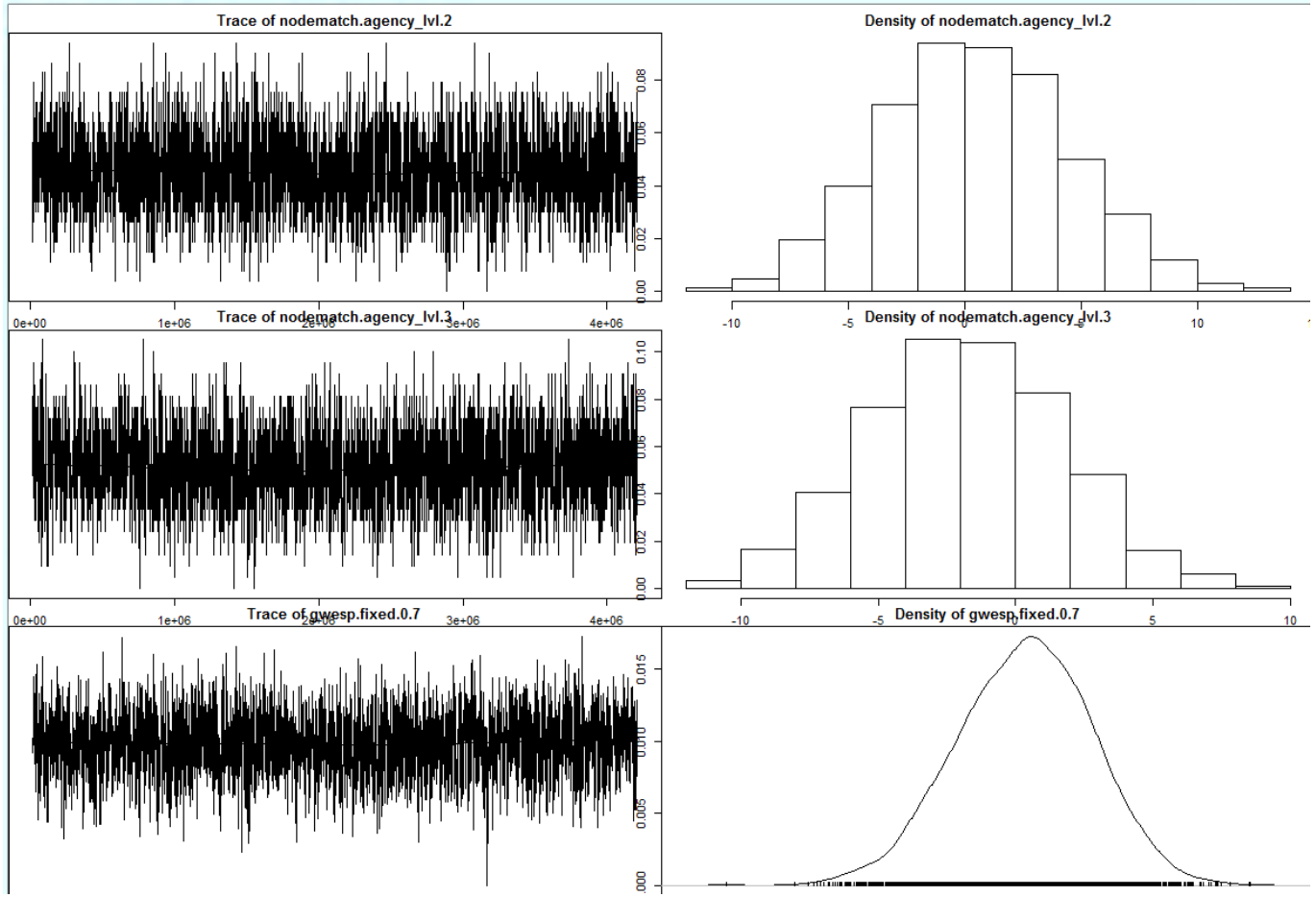


Goodness of Fit Evaluation

```
mcmc.diagnostics(DSmod4g)
```

Note:

- All the plots should be centered around 0 (or close to 0).
- MCMC diagnostics did not run unless I included the `gwesp` term in the formula.



Simulation of Edges, Degrees and Triangles

It seems adding `gwesp` term in DSmod4g does not the make the model differ too much from “main” which does not have `gwesp` term.

Note: One benefit of adding `gwesp` term was that it allowed MCMC processing to begin, and I was able to obtain MCMC diagnostics.

```
########## simulations
null<-ergm(TCdiss~edges)

#main is without gwesp term
main<-ergm(TCdiss~edges +
                nodecov('tob_yrs')+
                nodematch('agency_lvl',diff=TRUE),
                control=control.ergm(seed=40))
#simulations
nullsim <- simulate(null, verbose = TRUE,seed = 5)
mainsim <- simulate(main, verbose = TRUE, seed = 5)
DSmod4gsim <- simulate(DSmod4g, verbose = TRUE, seed = 5)

rowgof <- rbind(summary(TCdiss ~ edges + degree(0:5) + triangle),
                summary(nullsim ~ edges + degree(0:5) + triangle),
                summary(mainsim ~ edges + degree(0:5) + triangle),
                summary(DSmod4gsim ~ edges + degree(0:5) + triangle)
)
rownames(rowgof) <- c("TCDiss", "Null","Main","DSmod4g")
rowgof
```

	edges	degree0	degree1	degree2	degree3	degree4	degree5	triangle
TCDiss	103	0	1	2	3	2	1	192
Null	99	0	0	0	1	1	0	83
Main	100	0	0	1	3	0	2	124
DSmod4g	91	1	0	2	1	3	1	117

References

- Douglas A Luke, *A User's Guide to Network Analysis in R*
- J. K. Harris, *An Introduction to Exponential Random Graph Modeling*

Questions? Thoughts? Visit the course's online discussion forum.