Advanced Network Analysis

Latent Space Models

Olga Chyzh [www.olgachyzh.com]

Latent Space Models

- An alternative to ERGMs
- Allow to account for unobserved network dependencies (without measuring them)
- A highly parameterized, lightly supervised approach to fit out the dependencies
- Do not exhibit degeneracy issues

Trade Data

```
##              ARG         AUL         BEL         BNG         BRA
## ARG           NA 0.058268908 0.24686008 0.039220713 1.76473080
## AUL 0.086177696          NA 0.37843644 0.104360015 0.21511138
## BEL 0.270027137 0.350656872         NA 0.019802627 0.39877612
## BNG 0.000000000 0.019802627 0.12221763          NA 0.01980263
## BRA 1.693779061 0.239016900 0.62057649 0.039220713         NA
## CAN 0.190620360 0.641853886 0.61518564 0.076961041 0.66782937
## CHN 0.470003629 1.415853163 1.01160091 0.457424847 0.59883650
## COL 0.058268908 0.009950331 0.13976194 0.000000000 0.11332869
## EGY 0.000000000 0.009950331 0.08617770 0.009950331 0.01980263
## FRN 0.667829373 0.815364813 3.06011453 0.067658648 0.81536481
## IND 0.067658648 0.307484700 0.70309751 0.518793793 0.14842001
## INS 0.058268908 0.788457360 0.42526774 0.095310180 0.13102826
## IRN 0.000000000 0.009950331 0.30010459 0.009950331 0.37156356
## ITA 0.746687947 0.904218151 1.90657514 0.048790164 1.12817109
## JPN 0.587786665 2.215937286 1.48387469 0.418710335 1.23547147
## MEX 0.343589704 0.122217633 0.24686008 0.000000000 0.48242615
## NTH 0.190620360 0.418710335 3.21486780 0.086177696 0.41871033
## PAK 0.019802627 0.095310180 0.12221763 0.086177696 0.02955880
## PHI 0.009950331 0.173953307 0.09531018 0.000000000 0.03922071
## POL 0.029558802 0.019802627 0.39204209 0.009950331 0.13102826
```

Sender heterogeneity

An actor can induce dependence across its "recievers." Thus values across a row, say $\{y_{ij}, y_{ik}, y_{il}\}$, may be more similar to each other than other values in the adjacency matrix because each of these values has a common sender i.

	i	j	k	l
i	NA	y_{ij}	y_{ik}	y_{il}
j	y_{ji}	NA	y_{jk}	y_{jl}
k	y_{ki}	y_{kj}	NA	y_{kl}
l	y_{li}	y_{lj}	y_{lk}	NA

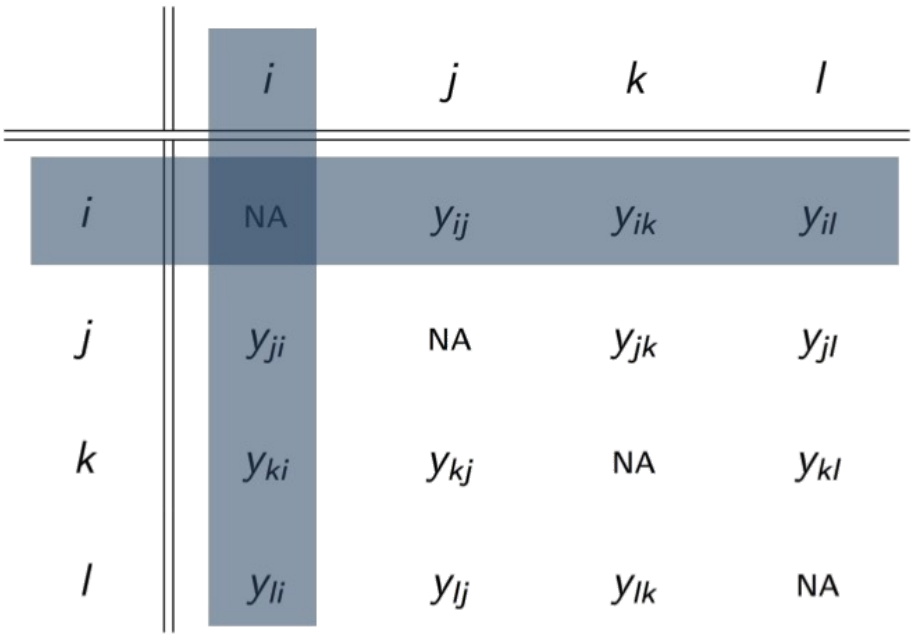
Receiver heterogeneity

Additionally, values across a column, say $\{y_{ji}, y_{ki}, y_{li}\}$, may be more similar to each other than other values in the adjacency matrix because each of these values has a common receiver i.

	i	j	k	l
i	NA	y_{ij}	y_{ik}	y_{il}
j	y_{ji}	NA	y_{jk}	y_{jl}
k	y_{ki}	y_{kj}	NA	y_{kl}
l	y_{li}	y_{lj}	y_{lk}	NA

Sender-Receiver Covariance

Actors who are more likely to send ties in a network may also be more likely to receive them.



	i	j	k	l
i	NA	y_{ij}	y_{ik}	y_{il}
j	y_{ji}	NA	y_{jk}	y_{jl}
k	y_{ki}	y_{kj}	NA	y_{kl}
l	y_{li}	y_{lj}	y_{lk}	NA

Reciprocity

Values of y_{ij} and y_{ji} may be statistically dependent. Dyads might exhibit high reciprocity because there is a tendency for actors to treat each other similarity, i.e., "respond in kind" to these behaviors.

	i	j	k	l
i	NA	y_{ij}	y_{ik}	y_{il}
j	y_{ji}	NA	y_{jk}	y_{jl}
k	y_{ki}	y_{kj}	NA	y_{kl}
l	y_{li}	y_{lj}	y_{lk}	NA

Latent distance model

(Hoff, Raftery, and Handcock 2002; Krivitsky et al. 2009; Sewell & Chen 2015)

Each node i has an unknown latent position

$$\mathbf{u}_i \in \mathbb{R}^k$$

The probability of a tie from i to j depends on the distance between them

$$Pr(Y_{ij} = 1|\mathbf{u}_i, \mathbf{u}_j, \theta) = \theta - |\mathbf{u}_i - \mathbf{u}_j|,$$

where $\theta = \alpha + \beta' x_{i,j}$, and α, β, $\mathbf{u}_i$ and $\mathbf{u}_j$ are estimated parameters.

- Nodes nearby one another are more likely to have a tie, and will likely have similar ties to others

Software packages:

- CRAN: `latentnet` (Krivitsky et al. 2015)
- CRAN: `VBLPCM` (Salter-Townshend 2015)

Example: Kirkland (2012)

Argument: Legislators from multimember districts collaborate more than those from single-member districts.

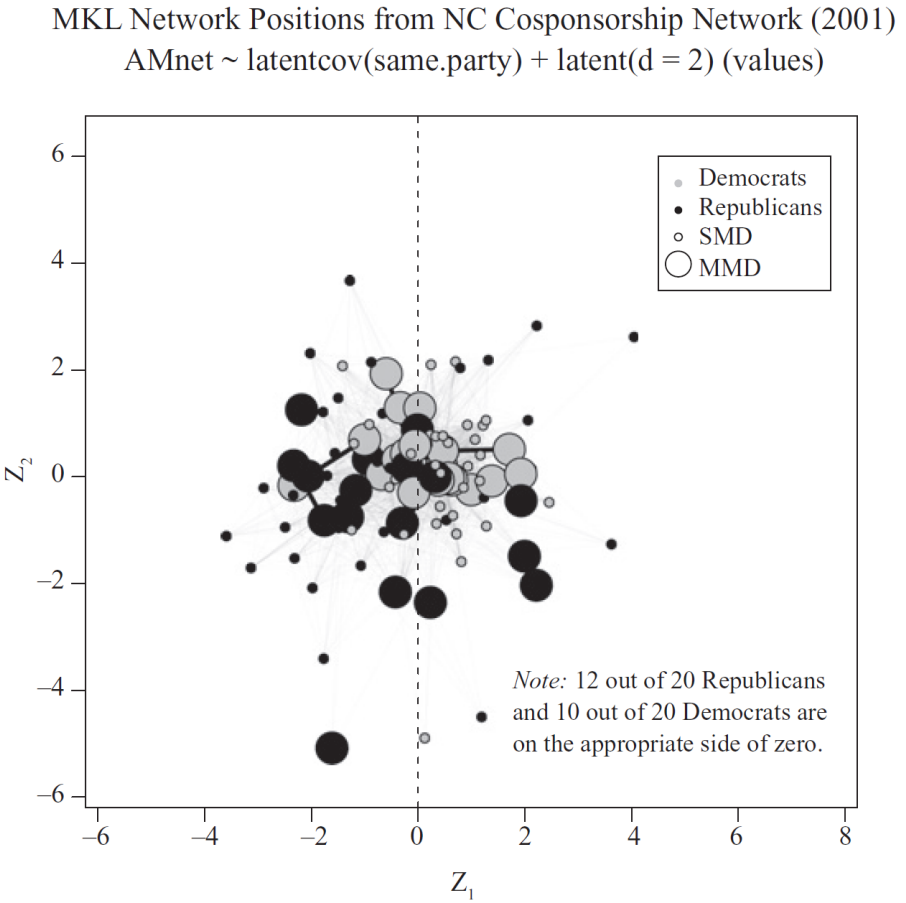
- Legislative policy preferences are multidimensional and are driven by a desire to satisfy constituents' preferences.
- Some issues (abortion) have clear partisan definitions, while other issues are less easy to define along a partisan continuum.
- Legislators from multimember districts or similar geographic regions may have very similar preferences over distributive legislation even if they are of different parties.
- So long as the legislature faces some real constraint on the amount of legislation it can pass, legislators from multimember districts will always have greater incentive to coordinate their efforts than those from single-member districts.

Kirkland (2012)

- Measures collaboration as co-sponsorships (network ties).
- During the 2000–2002 redistricting cycle, North Carolina's lower legislative chamber switched from a system that had 17 multimember districts electing 30 of the 120 legislators to a system that exclusively used single-member districts.
- The latent social space represents unmodeled similarities between actors that are implied by their relationships with one another.
- The latent space model allows for the assessment of distance between two unconnected actors based on the patterns of connections between other actors in the network.
- Because the model places actors in a multidimensional social space based on their connections with one another across many bills, it allows us to observe whether multimember legislators are closer on nonpartisan dimensions of behavior than single-member legislators.

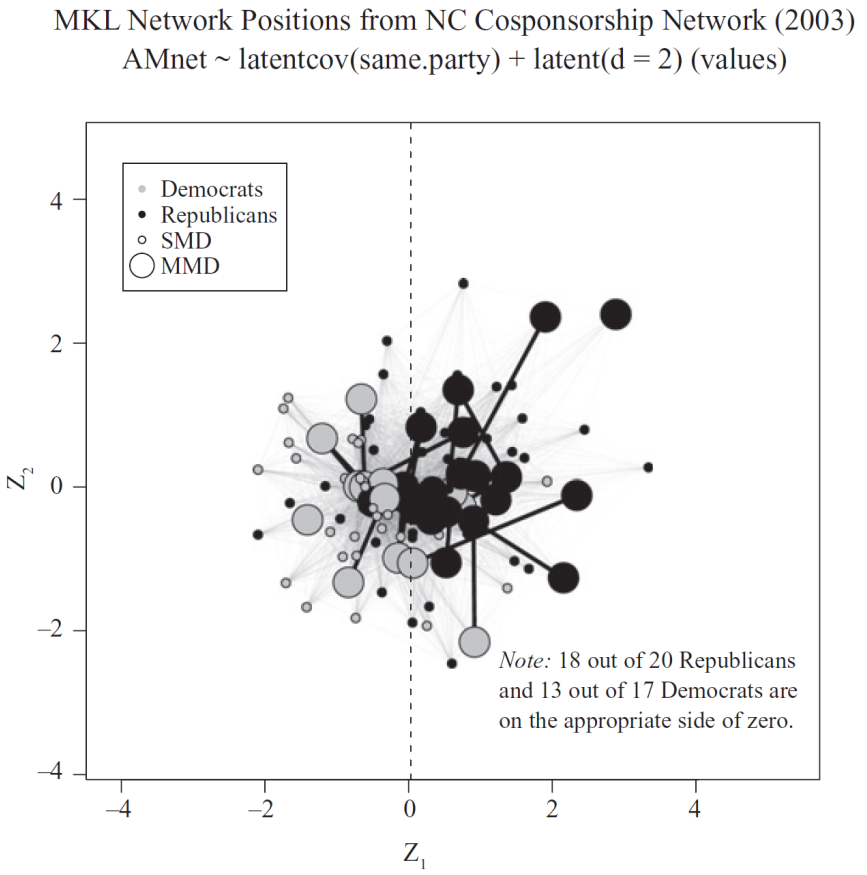
Results: Pre-Redistricting

(a) NC Network Positions in 2001



Results: Post-Redistricting

(b) NC Network Positions in 2003



Let's Replicate Kirkland (2012)

```
library(statnet)
library(latentnet)
library(networkdata)
data("Kirkland2012")
```

Data

- The DV is the cosponsorship network in each year, `AMnet01` and `AMnet03`

```
AMnet01
```

```
##  Network attributes:
##   vertices = 123
##   directed = TRUE
##   hyper = FALSE
##   loops = FALSE
##   multiple = FALSE
##   bipartite = FALSE
##   total edges= 7036
##     missing edges= 0
##     non-missing edges= 7036
##
##  Vertex attribute names:
##     vertex.names
##
##  Edge attribute names not shown
```

```
as.matrix(AMnet01)[1:5,1:5]
```

```
##   1 2 3 4 5
## 1 0 0 0 0 0
## 2 0 0 1 0 1
## 3 0 1 0 1 0
## 4 0 1 1 0 1
## 5 0 0 0 0 0
```

Data

- Three IVs: same district, same party, and their interaction

```
class(same.dist01)
```

```
## [1] "matrix" "array"
```

```
same.dist01[is.na(same.dist01)
same.dist01[1:5,1:5]
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
```

```
set.network.attribute(AMnet01,
```

```
class(same.party01)
```

```
## [1] "matrix" "array"
```

```
same.party01[1:5,1:5]
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    1    0    0    0    1
## [2,]    0    1    1    1    0
## [3,]    0    1    1    1    0
## [4,]    0    1    1    1    0
## [5,]    1    0    0    0    1
```

```
set.network.attribute(AMnet01,
```

Data

```
class(interact01)
```

```
## [1] "matrix" "array"
```

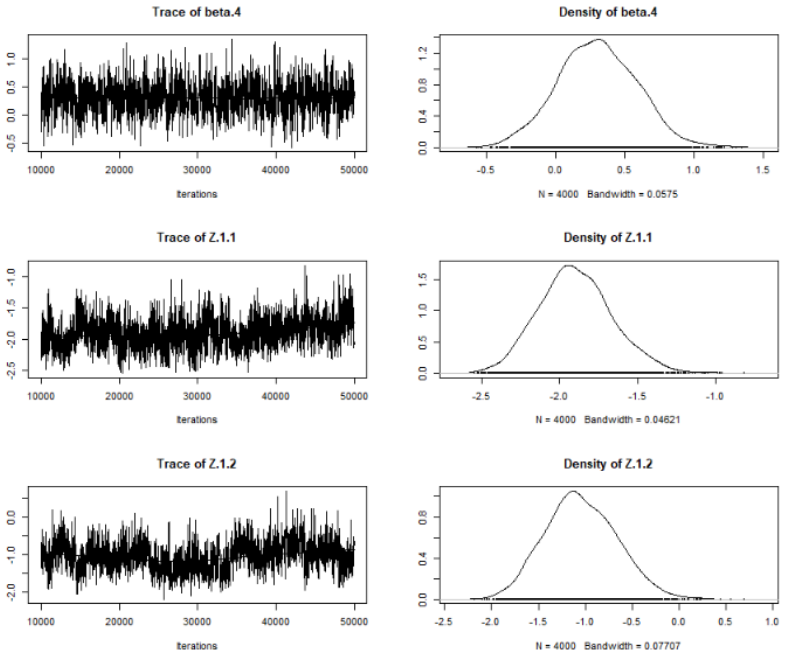
```
interact01[1:5,1:5]
```

```
##      [,1] [,2] [,3] [,4] [,5]
## [1,]    0    0    0    0    0
## [2,]    0    0    0    0    0
## [3,]    0    0    0    0    0
## [4,]    0    0    0    0    0
## [5,]    0    0    0    0    0
```

```
interact01[is.na(interact01)]<-0 #Make sure no missing data
set.network.attribute(AMnet01,"interact01",interact01)
```

Specify the model:

```
mod01<-ergmm(AMnet01~edgecov("sameparty")+edgecov("samedist")+edgecov
mcmc.diagnostics(mod01)
```



```
summary(mod01)
```

```
##
## ============================
## Summary of model fit
## ============================
##
## Formula:   AMnet01 ~ edgecov(same.party01) + edgecov(same.dist01) + edgecc
##     euclidean(d = 2)
## Attribute: values
## Model:     Poisson
## MCMC sample of size 4000, draws are 10 iterations apart, after burnin of 1
## Covariate coefficients posterior means:
##                       Estimate     2.5%  97.5% 2*min(Pr(>0),Pr(<0))
## (Intercept)            0.35025  0.27075 0.4299               <2e-16 ***
## edgecov.same.party01   0.82140  0.73717 0.9019               <2e-16 ***
## edgecov.same.dist01    1.76097  1.42983 2.0989               <2e-16 ***
## edgecov.interact01    -0.23959 -0.64644 0.1637               0.2445
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Overall BIC:        22493.03
## Likelihood BIC:     21550.57
## Latent space/clustering BIC:     942.455
##
## Covariate coefficients MKL:
##                        Estimate
```

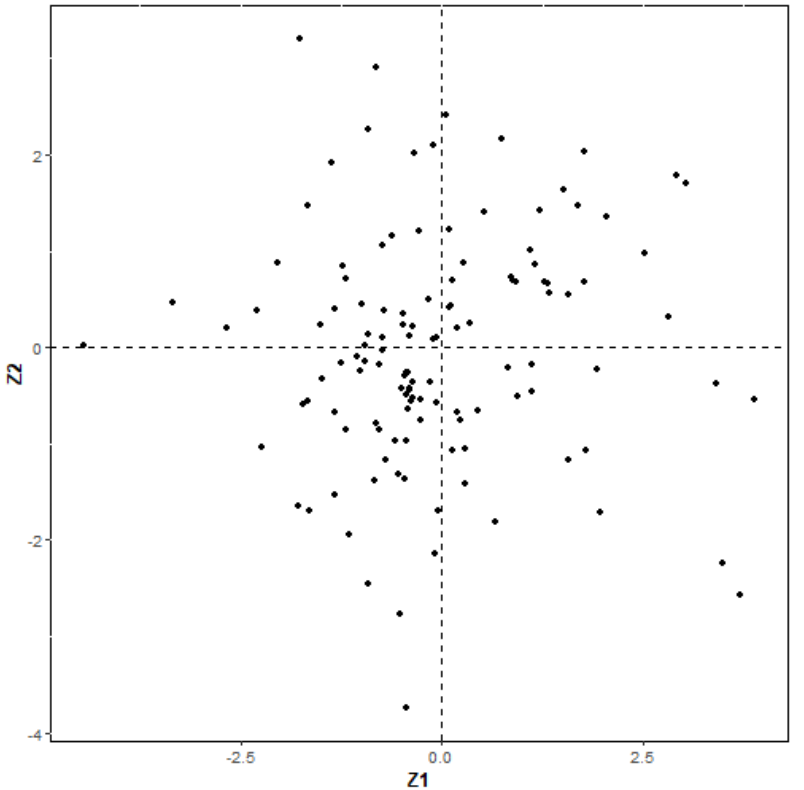
Pulling out Nodal Positions

Can pull out the positions of actor in a euclidean latent space!

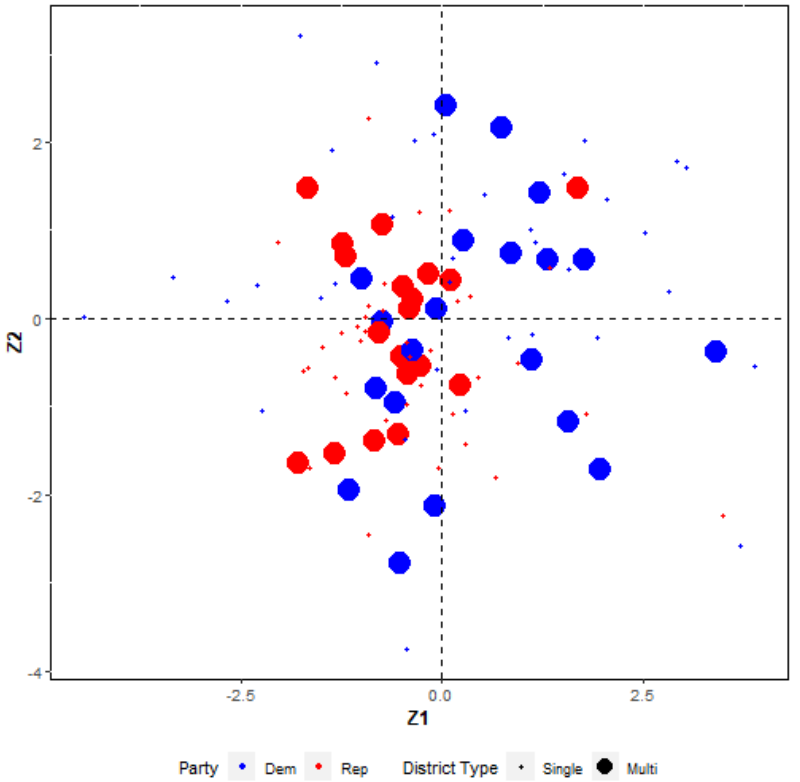
```
zPos = summary(mod01)$'pmean'$Z
head(zPos)
```

```
##             [,1]       [,2]
## [1,]  3.6992586 -2.5726430
## [2,] -0.7859004 -0.8594968
## [3,] -0.3606575 -0.5181687
## [4,] -0.3537175  0.2149023
## [5,]  0.2619265  0.8745140
## [6,] -0.6138960  1.1505274
```

Plot Nodal Positions



Show Party and District Type:



Your Turn:

1. Estimate legislators' latent positions for 2003.
2. Check the model's diagnostics.
3. Make a graph showing positions by party and district type.
4. Does your graph look the same as Figure 4b of Kirkland (2012)?

Apply LDM to International Trade

```
data("Dyadic_COW_4.0")
class(TRADE)
```

```
## [1] "data.frame"
```

```
TRADE[1:5,1:5]
```

```
##               2            20           31          40            41
## 2            NA 346062.59000 542.0477300   0.000000 9.082188e+02
## 20 276158.2800            NA  91.8779140 558.916440 4.011414e+01
## 31   3690.3623     246.83345           NA   1.170046 6.561742e-01
## 40    329.0571     445.61664   0.0000000          NA 6.568751e-04
## 41   1404.2172      47.87239   0.1397724  25.439621           NA
```

Set Up Trade as A Valued Network

```
tradenet<-network(as.matrix(TRADE),directed=TRUE,matrix.type='adjacer
    ignore.eval = FALSE,
  names.eval = "volumes")
as.matrix(tradenet[1:10,1:10])
```

```
##    2 20 31 40 41 42 51 52 53 54
## 2  0  1  1  0  1  1  1  1  1  1
## 20 1  0  1  1  1  1  1  1  1  1
## 31 1  1  0  1  1  1  1  1  1  1
## 40 1  1  0  0  1  1  1  1  1  0
## 41 1  1  1  1  0  1  1  1  1  1
## 42 1  1  1  1  1  0  1  1  1  1
## 51 1  1  1  1  1  1  0  1  1  1
## 52 1  1  1  1  1  1  1  0  1  1
## 53 1  1  1  1  1  1  1  1  0  1
## 54 1  1  1  0  1  1  1  1  1  0
```

```
network::list.edge.attributes(tradenet)
```

```
## [1] "na"      "volumes"
```

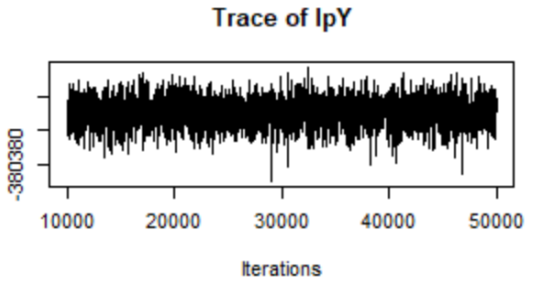
```
as.matrix(tradenet, attrname = "volumes")[1:10,1:10]
```

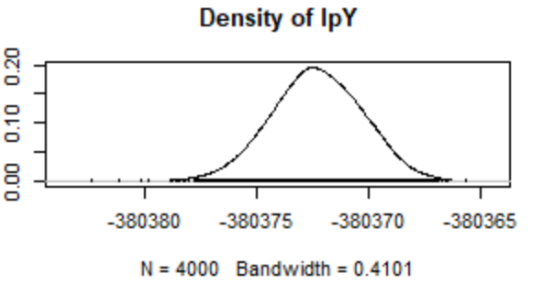
```
##                2           20          31          40           41
## 2        0.00000 3.460626e+05 542.0477300   0.0000000 9.082188e+02 4518.897
## 20 276158.28000 0.000000e+00  91.8779140 558.9164400 4.011414e+01 1637.674
## 31   3690.36230 2.468334e+02   0.0000000   1.1700455 6.561742e-01    4.349
## 40    329.05710 4.456166e+02   0.0000000   0.0000000 6.568751e-04   47.120
## 41   1404.21720 4.787239e+01   0.1397724  25.4396210 0.000000e+00 1727.940
## 42   8750.12010 1.946108e+02 390.3346300  79.5329970 3.781249e+00    0.000
## 51   2391.60080 1.199765e+02  53.1063580   1.4854457 1.112114e-02   53.548
## 52   2655.14530 3.144949e+02  12.5382950   1.0545393 8.435080e-01   32.665
## 53    627.33594 1.954806e+02   0.3627099   0.5254627 1.780585e-01    9.354
## 54     70.63872 9.088092e+00   0.2890149   0.0000000 3.253810e-02    4.304
##              51         52           53          54
## 2  285.127560 5992.21240  51.10862000  1.51433000
## 20 263.126430  554.48572  12.77633400  0.35298631
## 31  26.933874  149.35327   7.09561160  0.28212446
## 40   8.074952   66.20779   0.01613329  0.00000000
## 41   2.079861   27.44184   2.78890540  0.02921979
## 42   4.533408  791.60394   4.27885200  0.11252818
## 51   0.000000  726.87836  29.00094600 22.37743000
## 52  21.496393    0.00000 129.24066000 12.94273800
## 53  28.558105  761.58569   0.00000000  5.27564380
## 54   3.333000   98.98096  11.50654800  0.00000000
```

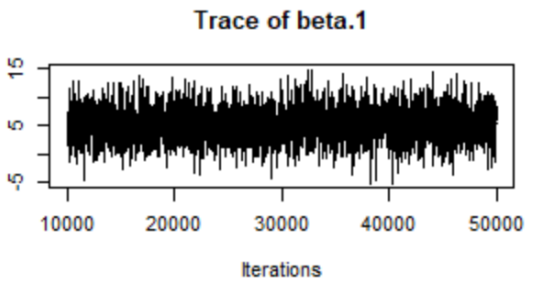
Estimate the Model

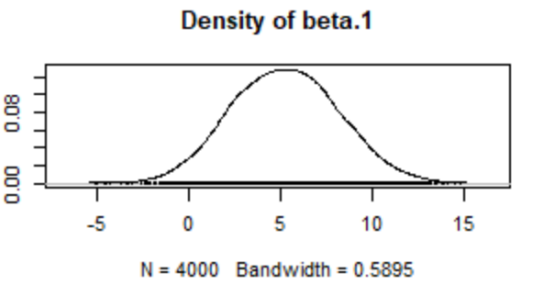
```
y.var<-4*sd(as.matrix(TRADE), na.rm=TRUE) #Need to give a variance pr
m1<-ergmm(tradenet~euclidean(d=2),
          family="normal",
          response="volumes",
          verbose=1,
          fam.par=list(
          prior.var=y.var,
          prior.var.df=1 # certainty of the prior, higher more certa
          ))
mcmc.diagnostics(m1)
```

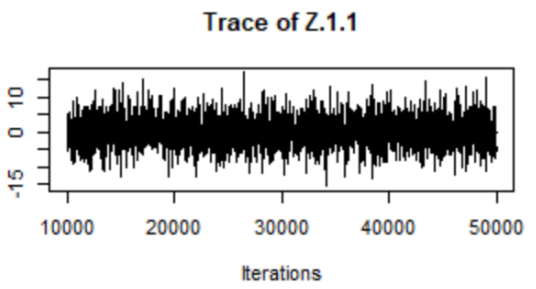
Trace Plots

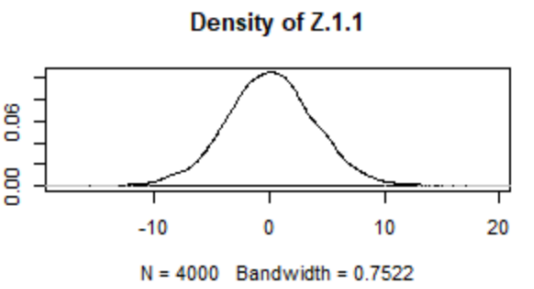

Trace of lpY
-380380
10000
20000
30000
40000
50000
Iterations


Density of lpY
0.00
0.10
0.20
-380380
-380375
-380370
-380365
N = 4000 Bandwidth = 0.4101


Trace of beta.1
-5
5
15
10000
20000
30000
40000
50000
Iterations


Density of beta.1
0.00
0.08
-5
0
5
10
15
N = 4000 Bandwidth = 0.5895

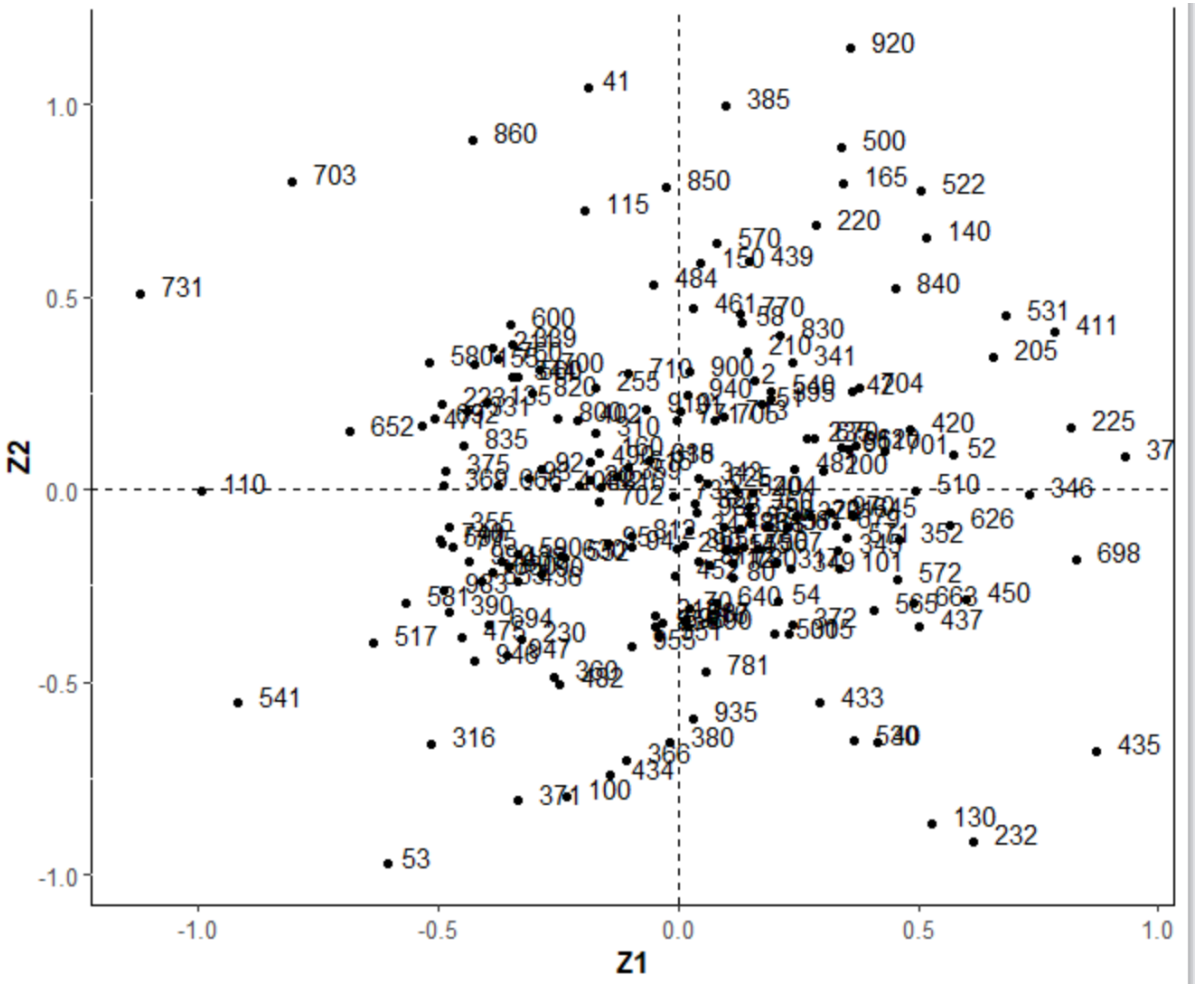

Trace of Z.1.1
-15
0
10
10000
20000
30000
40000
50000
Iterations


Density of Z.1.1
0.00
0.06
-10
0
10
20
N = 4000 Bandwidth = 0.7522

Your Turn

1. Pull out nodal coordinates that place each country in a latent space.
2. Plot countries in latent space.
3. Do you see any patterns? What variables should we control for, in order for the latent space to represent policy similarity among countries.

Plot Countries in Latent Space



Latent factor model

(Hoff 2003; Hoff 2007; Minhas et al. 2018; Hoff 2018)

Each node i has an unknown latent factor

$$\mathbf{u}_i \in \mathbb{R}^k$$

The probability of a tie from i to j depends on their latent factors

$$\begin{aligned} Pr(Y_{ij} = 1 | \mathbf{u}_i, \mathbf{u}_j) =& \theta + \mathbf{u}_i^T \Lambda \mathbf{u}_j \text{ , where} \\ & \Lambda \text{ is a } K \times K \text{ diagonal matrix} \end{aligned}$$

- Each node i has a vector of unobserved characteristics $u_i = \{u_{i1}, \ldots, u_{iK}\}$
- Similar values of u_{ik} and u_{jk} will contribute positively or negatively to the relationship between i and j, i.e. positive or negative homophily between i and j.
- This is a generalization of the latent distance model

- Can account for both stochastic equivalence and homophily
- Comes at the cost of harder to interpret multiplicative factors ... let's see what I mean

Software packages:

- CRAN: `amen` (Hoff et al. 2015)

Running LFM through AME

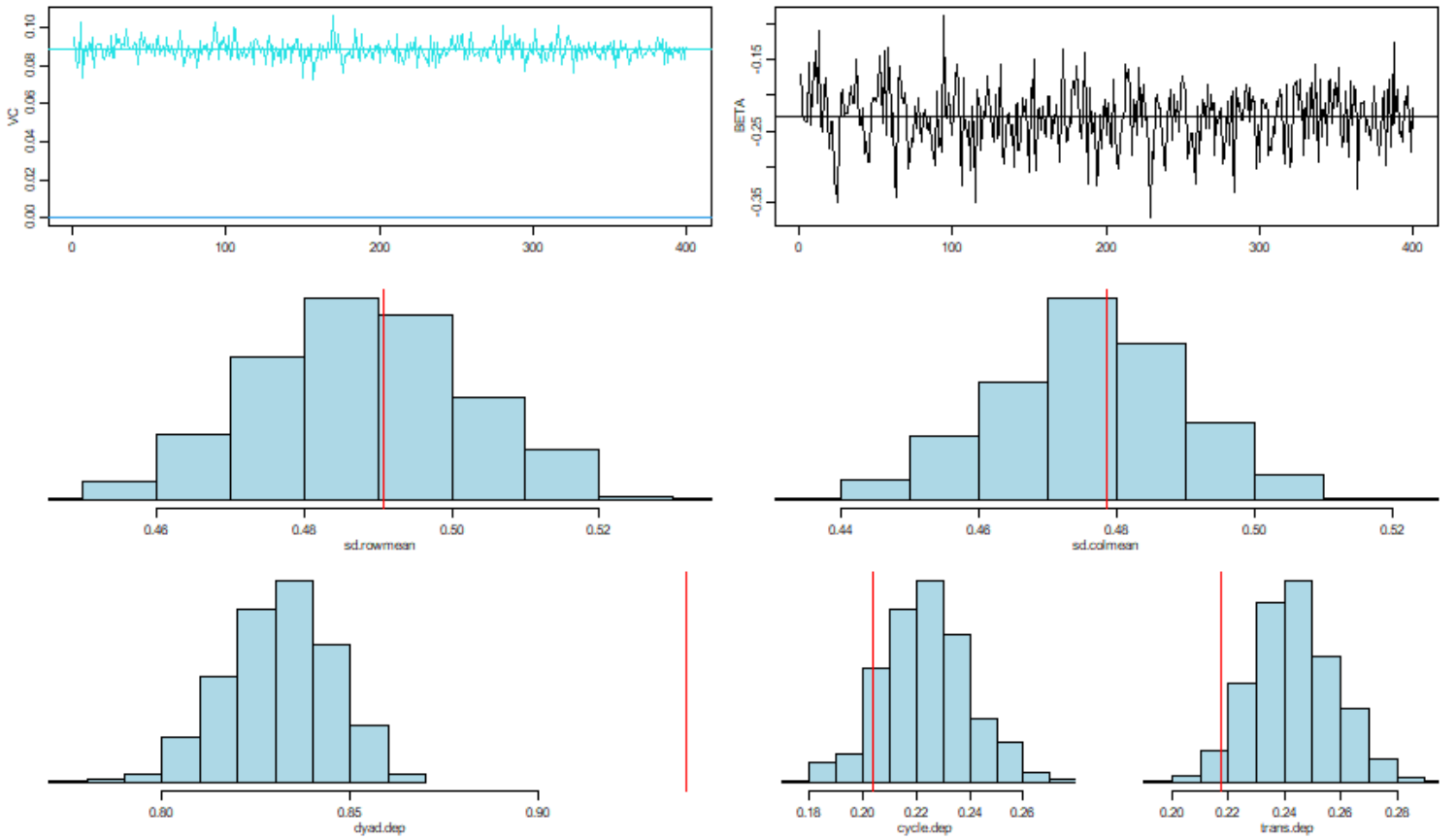
To run a latent factor model, we can use the amen package:

```
library(amen) # Load additive and multiplicative effects pkg
data(IR90s) # Load trade data
gdp = IR90s$nodevars[,2]
topgdp = which(gdp>=sort(gdp,decreasing=TRUE)[30] )
Y<-log( IR90s$dyadvars[topgdp,topgdp,2] + 1 )
#Our DV is logged exports, our sample is
#the top 30 countries in terms of gdp
```

```
lfmFit = ame(Y,
  family='nrm', symmetric=FALSE,
  seed=6886,
  # restrict SRM parameters
  cvar=FALSE, rvar=FALSE, dcor=FALSE,
  R=2,
  plot=FALSE, print=FALSE
  )
```

How well do we capture network effects?

```
plot(lfmFit)
```



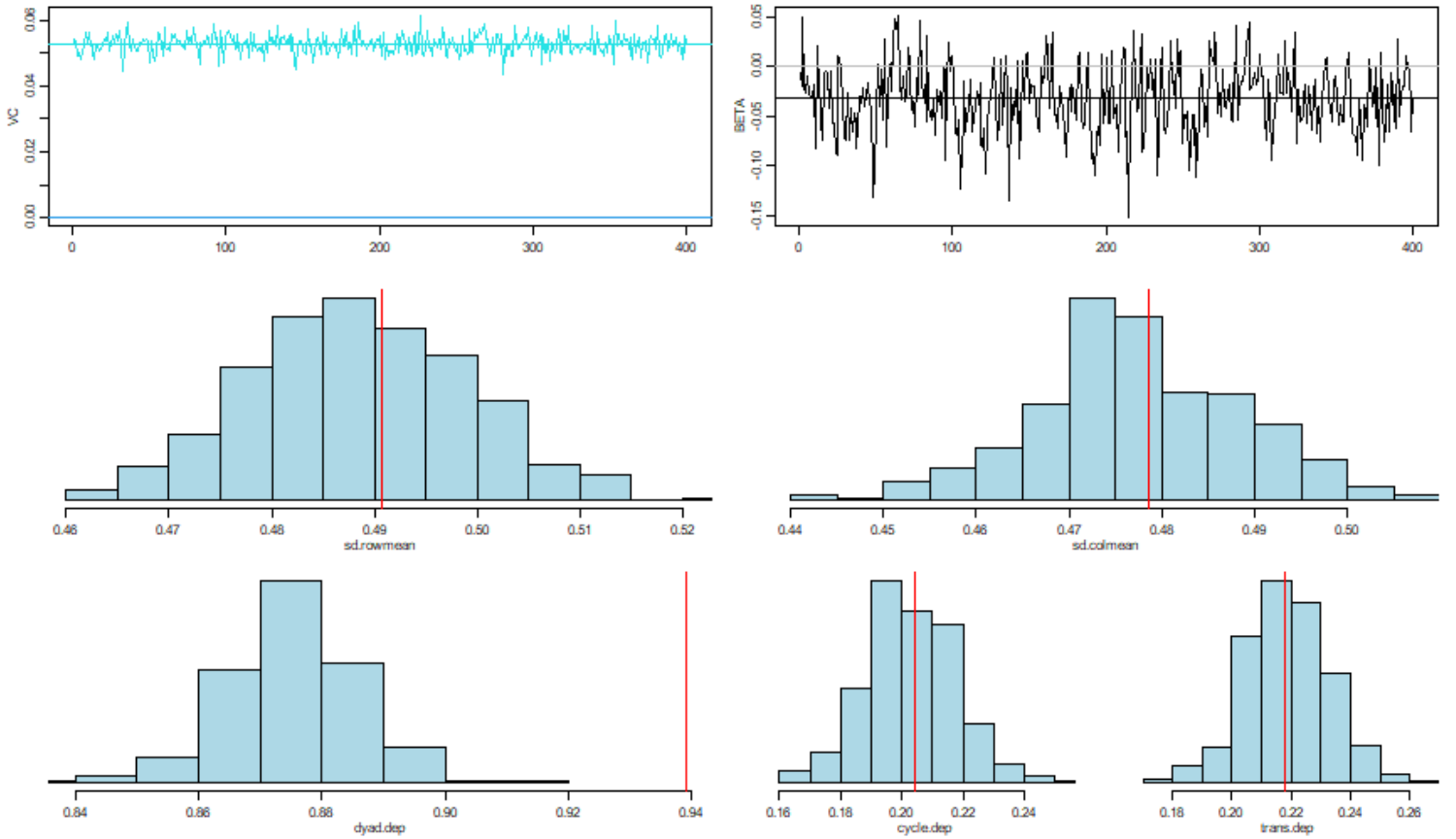
When to increase R

If our model is not adequately accounting for network effects, we can adjust the dimension of the multiplicative effects, R, in the LFM framework:

```
lfmFitk3 = ame(Y,
  family='nrm', symmetric=FALSE,
  seed=6886,
  cvar=FALSE, rvar=FALSE, dcor=FALSE,
  R=3,
  plot=FALSE, print=FALSE
  )
```

Check GOF again

```
plot(lfmFitk3)
```



Interpreting the UV term

So what is this UV term?

```
lfmFitk3$U[1:3,]
```

```
##       [,1]   [,2]   [,3]
## ARG -0.343 -0.123 -0.047
## AUL -0.799 -0.119  0.631
## BEL -1.145 -0.084 -1.006
```

```
lfmFitk3$V[1:3,]
```

```
##       [,1]  [,2]   [,3]
## ARG -0.408 0.246 -0.039
## AUL -0.813 0.331  0.370
## BEL -1.165 0.229 -0.982
```

```
lfmFitk3$UVPM[1:3,1:3]
```

```
##       ARG   AUL   BEL
## ARG 0.112 0.221 0.417
## AUL 0.272 0.844 0.284
## BEL 0.485 0.531 2.304
```

How can we use it?

We can interpret the cross-sections of the UV term as a measure of how likely a pair of actors are to form an edge with one another:

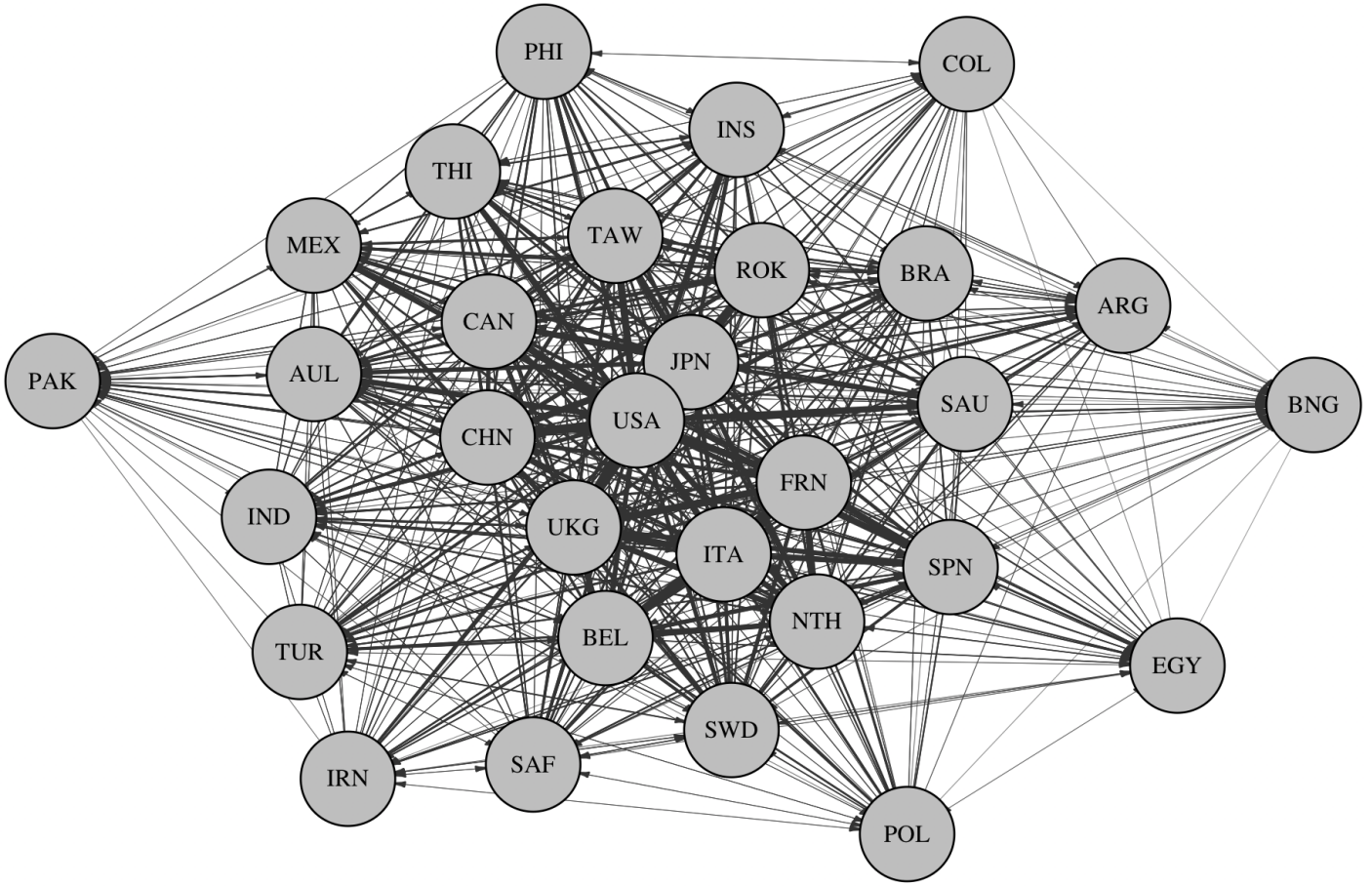
```
uv = lfmFit$UVPM ; diag(uv) = NA
uvNet = igraph::graph.adjacency(uv,
  mode='directed',
  weighted=TRUE,
  diag=FALSE)

# create graph object
diag(Y) = NA
yGraph = igraph::graph.adjacency(Y,
  mode='directed',
  weighted=TRUE,
  diag=FALSE
  )
```

```r
# add node attributes
library(scales)
V(yGraph)$size = rescale(
  apply(Y, 2, sum, na.rm=TRUE), c(10, 16) )

set.seed(6886)
plot(uvNet,
  vertex.color='grey',
  vertex.label.color='black',
  vertex.size=V(yGraph)$size,
  vertex.label.cex =.75,
  edge.color='grey20',
  edge.width=E(uvNet)$weight,
  edge.arrow.size=.2,
  asp=FALSE
  )
```

Visualizing UV



Putting it all together: AME

$$
\begin{aligned}
y_{ij} &= g(\theta_{ij}) \\
\theta_{ij} &= \beta^T \mathbf{X}_{ij} + e_{ij} \\
e_{ij} &= a_i + b_j + \epsilon_{ij} + \mathbf{u}_i^T \mathbf{D} \mathbf{v}_j
\end{aligned}
$$

- $a_i + b_j + \epsilon_{ij}$, are additive random effects and account for sender, receiver, and within-dyad dependence
- multiplicative effects, $\mathbf{u}_i^T \mathbf{D} \mathbf{v}_j$, capture higher-order dependence patterns that are left over in θ after accounting for any known covariate information

AME Gibbs Sampler

- Probit regression framework: $y_{ij,t} = g(\theta_{ij,t})$, where $\theta_{ij,t} = \beta^\top \mathbf{X}_{ij,t} + a_i + b_j + \mathbf{u}_i^\top \mathbf{D} \mathbf{v}_j + \epsilon_{ij}$
- Prior distributions for the parameters are specified as follows:
  - β drawn from multivariate normals with mean zero and a (0,10) covariance matrix
  - $\Sigma_{a,b} \sim$ inverse Wishart$(I_{2\times 2}, 4)$
  - σ_u^2, and σ_v^2 are each drawn from an i.i.d. inverse gamma(1,1)

AME Gibbs Sampler

- Given initial values of $\{\beta, \mathbf{a}, \mathbf{b}, \mathbf{U}, \mathbf{V}, \Sigma_{ab}, \rho, \text{ and } \sigma_\epsilon^2\}$, the algorithm proceeds as follows:
  - sample $\theta \mid \beta, \mathbf{X}, \theta, \mathbf{a}, \mathbf{b}, \mathbf{U}, \mathbf{V}, \Sigma_{ab}, \rho, \text{ and } \sigma_\epsilon^2$ (Normal)
  - sample $\beta \mid \mathbf{X}, \theta, \mathbf{a}, \mathbf{b}, \mathbf{U}, \mathbf{V}, \Sigma_{ab}, \rho, \text{ and } \sigma_\epsilon^2$ (Normal)
  - sample $\mathbf{a}, \mathbf{b} \mid \beta, \mathbf{X}, \theta, \mathbf{U}, \mathbf{V}, \Sigma_{ab}, \rho, \text{ and } \sigma_\epsilon^2$ (Normal)
  - sample $\Sigma_{ab} \mid \beta, \mathbf{X}, \theta, \mathbf{a}, \mathbf{b}, \mathbf{U}, \mathbf{V}, \rho, \text{ and } \sigma_\epsilon^2$ (Inverse-Wishart)
  - update ρ using a Metropolis-Hastings step with proposal $p^* \big| p \sim$ truncated normal${[-1,1]}(\rho, \sigma^{2})$
  - sample $\sigma_\epsilon^2 \mid \beta, \mathbf{X}, \theta, \mathbf{a}, \mathbf{b}, \mathbf{U}, \mathbf{V}, \Sigma_{ab}, \text{ and } \rho$ (Inverse-Gamma)
  - For each $k \in K$:
  - Sample $\mathbf{U}_{[,k]} \mid \beta, \mathbf{X}, \theta, \mathbf{a}, \mathbf{b}, \mathbf{U}_{[,-k]}, \mathbf{V}, \Sigma_{ab}, \rho, \text{ and } \sigma_\epsilon^2$ (Normal)
  - Sample $\mathbf{V}_{[,k]} \mid \beta, \mathbf{X}, \theta, \mathbf{a}, \mathbf{b}, \mathbf{U}, \mathbf{V}_{[,-k]}, \Sigma_{ab}, \rho, \text{ and } \sigma_\epsilon^2$ (Normal)
  - Sample $\mathbf{D}_{[k,k]} \mid \beta, \mathbf{X}, \theta, \mathbf{a}, \mathbf{b}, \mathbf{U}, \mathbf{V}, \Sigma_{ab}, \rho, \text{ and } \sigma_\epsilon^2$ (Normal)

Set up Trade Data

```
#### ---- nodal covariates
dimnames(IR90s$nodevars)[[2]]
```

```
## [1] "pop"    "gdp"    "polity"
```

```
Xn<-IR90s$nodevars[topgdp,]
Xn[,1:2]<-log(Xn[,1:2])
#### ---- dyadic covariates
dimnames(IR90s$dyadvars)[[3]]
```

```
## [1] "conflicts"   "exports"     "distance"    "shared_igos" "polity_int"
```

```
Xd<-IR90s$dyadvars[topgdp,topgdp,c(1,3,4,5)]
Xd[,,3]<-log(Xd[,,3])
```

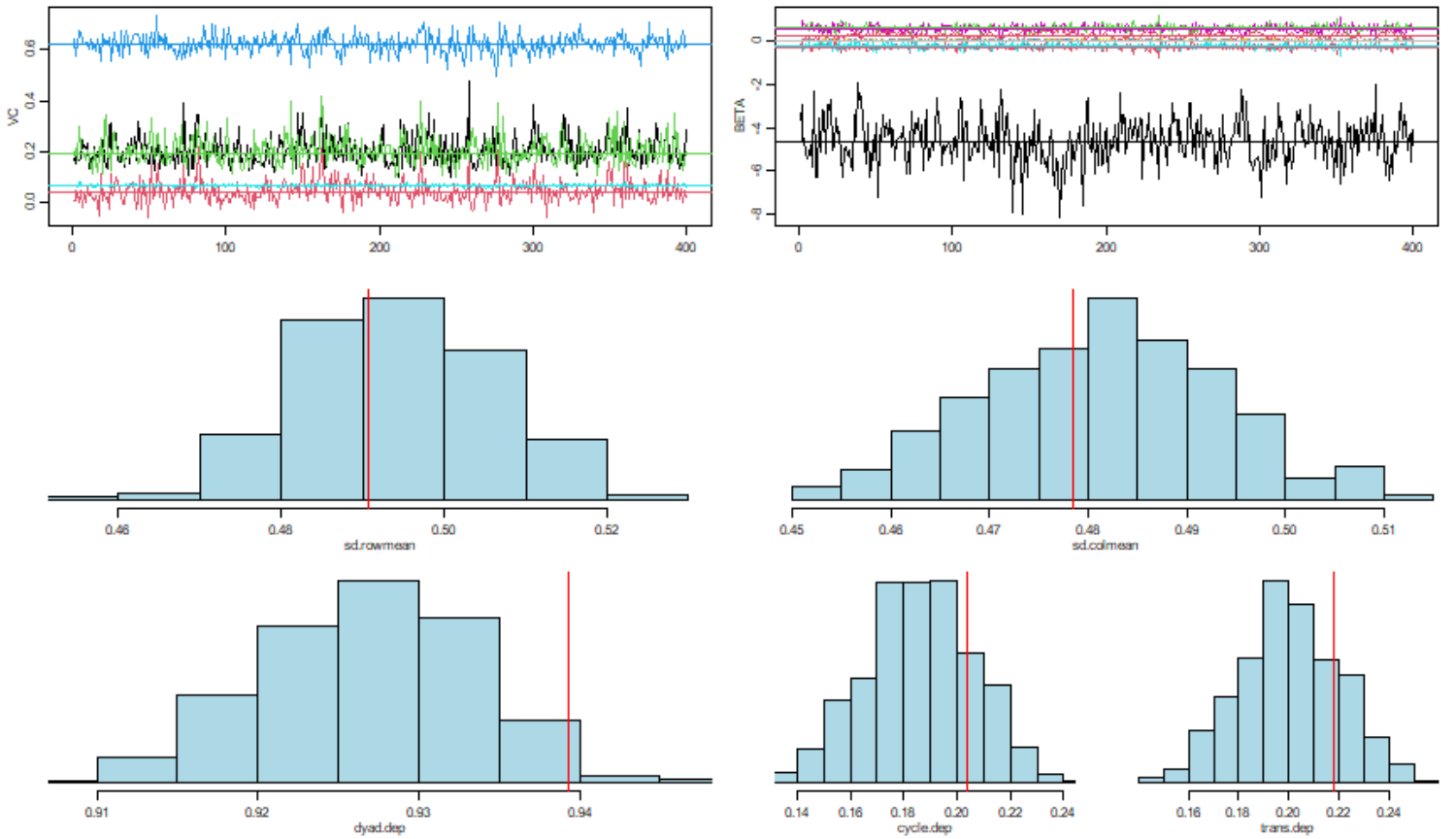
Estimating with multiplicative effects

Multiplicative effects can be added by toggling the `R` input parameter

```
fitAME = ame(Y=Y,
  Xdyad=Xd, # incorp dyadic covariates
  Xrow=Xn, # incorp sender covariates
  Xcol=Xn, # incorp receiver covariates
  symmetric=FALSE, # tell AME trade is directed
  intercept=TRUE, # add an intercept
  family='nrm', # model type
  rvar=TRUE, # sender random effects (a)
  cvar=TRUE, # receiver random effects (b)
  dcor=TRUE, # dyadic correlation
  R=2, # 2 dimensional multiplicative effects
  nscan=10000, burn=25, odens=25,
  plot=FALSE, print=FALSE, gof=TRUE
  )
```

Capturing network features part 2

```
plot(fitAME)
```



Summary method

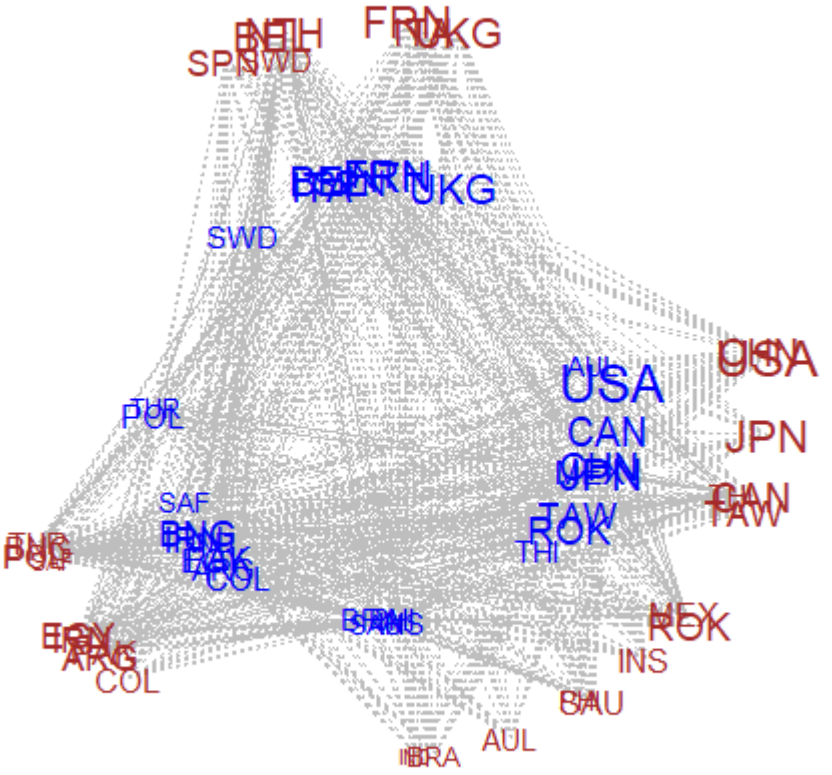
```
summary(fitAME)
```

```
##
## Regression coefficients:
##                    pmean   psd z-stat p-val
## intercept         -4.627 1.010 -4.581 0.000
## pop.row           -0.301 0.116 -2.598 0.009
## gdp.row            0.583 0.136  4.280 0.000
## polity.row        -0.002 0.017 -0.121 0.903
## pop.col           -0.259 0.119 -2.168 0.030
## gdp.col            0.539 0.147  3.670 0.000
## polity.col         0.008 0.017  0.464 0.643
## conflicts.dyad     0.017 0.038  0.441 0.659
## distance.dyad     -0.035 0.005 -7.053 0.000
## shared_igos.dyad   0.234 0.124  1.892 0.058
## polity_int.dyad   -0.001 0.000 -2.549 0.011
##
## Variance parameters:
##     pmean   psd
## va  0.202 0.058
## cab 0.042 0.044
## vb  0.201 0.054
```

Visualizing the multiplicative effects

```
x=circplot(
  Y=Y, U=fitAME$U, V=fitAME$V,
  vscale=.6
  )
```

Visualizing the multiplicative effects



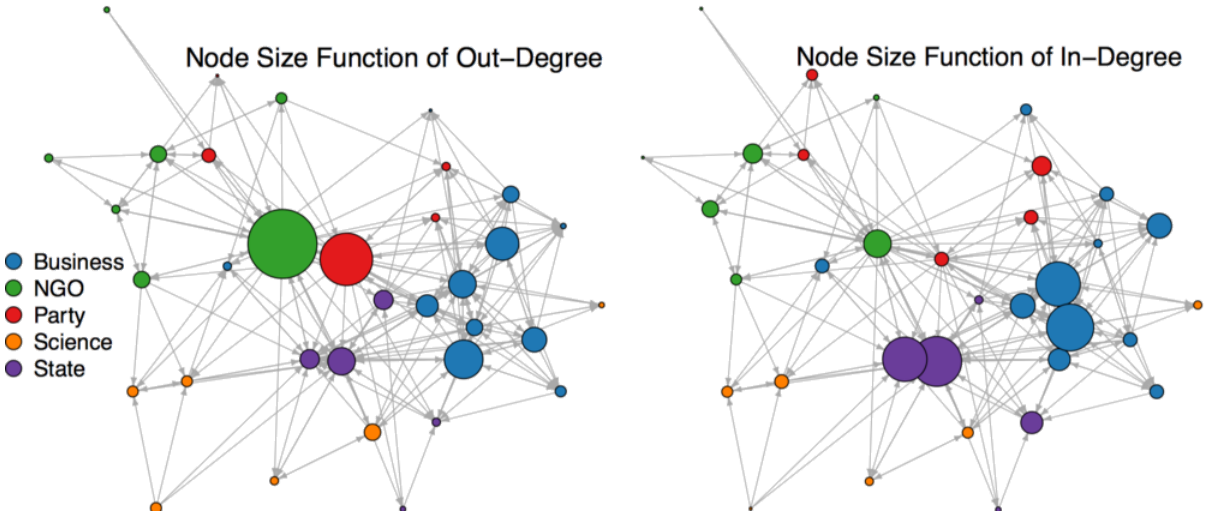
Benefits of this approach

- At its core, AME is just a GLM with random effects used to ensure that we can treat dyadic observations as conditionally independent
- AME can be used:
  - on both undirected and directed data,
  - on longitudinal and static networks,
  - and on a variety of distribution types we commonly encounter in political science (binomial, gaussian, and ordinal).

Real world comparison

Cranmer et al. (2017)

- Compares a few inferential network approaches
- Utilized Swiss climate change policy collaboration network as application (Ingold, 2008)



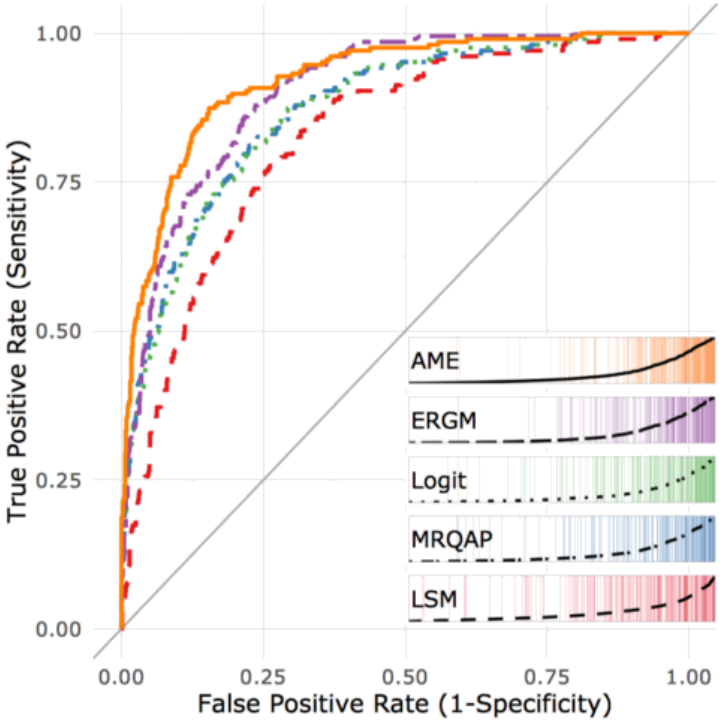
β Estimates

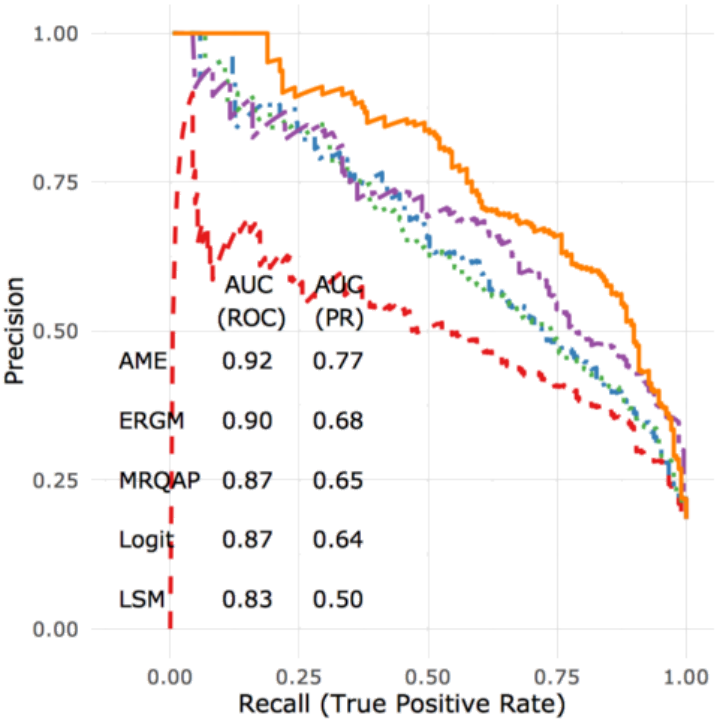
	Logit	MRQAP	LSM	ERGM	AME
Intercept/Edges	-4.44*	-4.24*	0.94*	-12.17*	-3.39*
	(0.34)		[0.09; 1.82]	(1.40)	[-4.38; -2.50]
Conflicting policy preferences					
Business vs. NGO	-0.86	-0.87*	-1.37*	-1.11*	-1.37*
	(0.46)		[-2.42; -0.41]	(0.51)	[-2.44; -0.47]
Opposition/alliance	1.21*	1.14*	0.00	1.22*	1.08*
	(0.20)		[-0.40; 0.39]	(0.20)	[0.72; 1.47]
Preference dissimilarity	-0.07	-0.60	-1.76*	-0.44	-0.79*
	(0.37)		[-2.62; -0.90]	(0.39)	[-1.55; -0.08]
Transaction costs					
Joint forum participation	0.88*	0.75*	1.51*	0.90*	0.92*
	(0.27)		[0.86; 2.17]	(0.28)	[0.40; 1.47]
Influence					
Influence attribution	1.20*	1.29*	0.08	1.00*	1.09*
	(0.22)		[-0.40; 0.55]	(0.21)	[0.69; 1.53]
Alter's influence indegree	0.10*	0.11*	0.01	0.21*	0.11*
	(0.02)		[-0.03; 0.04]	(0.04)	[0.07; 0.15]
Influence absolute diff.	-0.03*	-0.06*	0.04	-0.05*	-0.07*
	(0.02)		[-0.01; 0.09]	(0.01)	[-0.11; -0.03]
Alter = Government actor	0.63*	0.68	-0.46	1.04*	0.55
	(0.25)		[-1.08; 0.14]	(0.34)	[-0.07; 1.15]

	Logit	MRQAP	LSM	ERGM	AME
Functional requirements					
Ego = Environmental NGO	0.88*	0.99	-0.60	0.79*	0.67
	(0.26)		[-1.32; 0.09]	(0.17)	[-0.38; 1.71]
Same actor type	0.74*	1.12*	1.17*	0.99*	1.04*
	(0.22)		[0.63; 1.71]	(0.23)	[0.63; 1.50]
Endogenous dependencies					
Mutuality	1.22*	1.00*		0.81*	0.39
	(0.21)			(0.25)	[-0.12; 0.96]
Outdegree popularity				0.95*	
				(0.09)	
Twopaths				-0.04*	
				(0.02)	
GWIdegree (2.0)				3.42*	
				(1.47)	
GWESP (1.0)				0.58*	
				(0.16)	
GWOdegree (0.5)				8.42*	
				(2.11)	

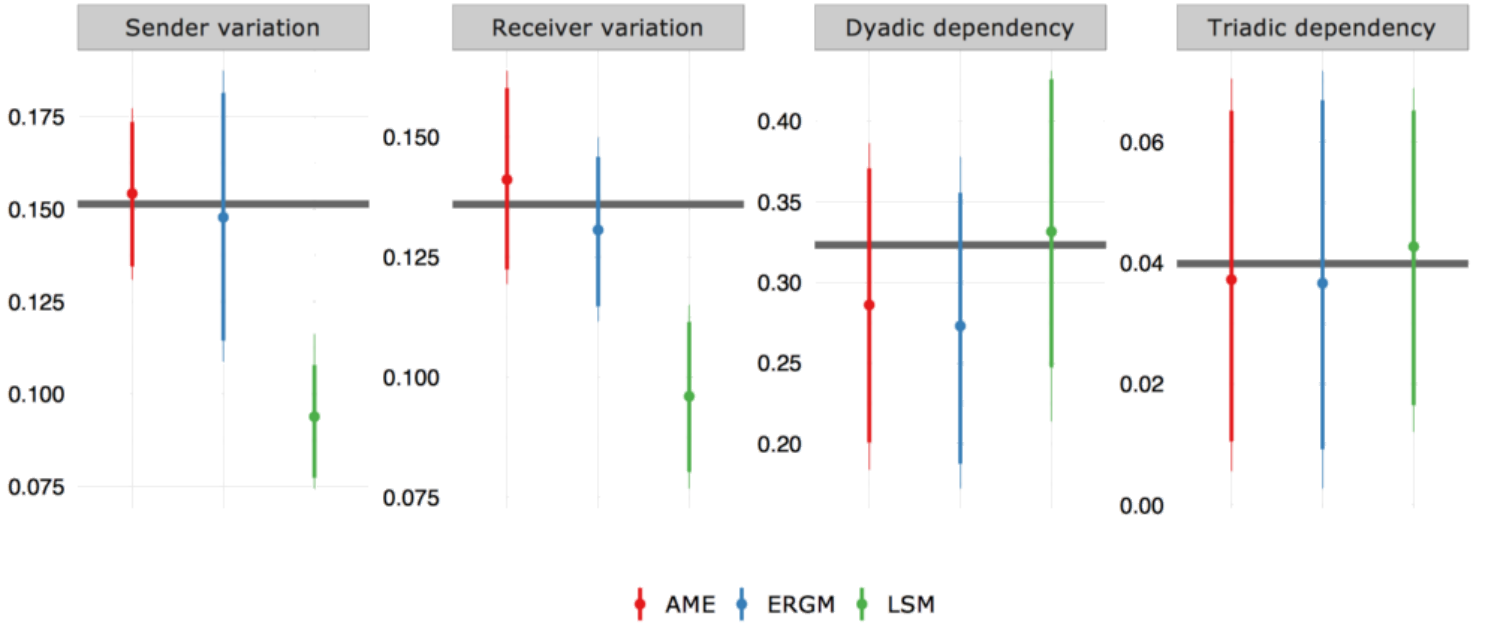
Which approach fits Y best?

Out-of-sample Network Cross-Validation





Which approach fits network dependencies best?



Summary

- LFM is a powerful framework that has proven useful
- A lot of other things going on:
    - Community structure in longitudinal, multidimensional arrays (Mucha et al. 2010)
    - Multilinear tensor regression (Hoff 2015, Schein et al. 2015, Minhas et al. 2016)
    - Intersection of network based methods to text analysis (Henry et al. 2016, Huang et al. 2015)
- Takeaway here is that these methods are useful when we study systems in which interactions are interdependent
- These interdependent relations may at times be of interest themselves or in other cases may just help us to better predict