# Are Covariates and Latent Positions Confounded in Euclidean Latent Space Models?

Philip Leifeld

University of Glasgow

≈ 2,000 words (excluding Figures and References)

Address for correspondence:

Philip Leifeld, University of Glasgow, School of Social and Political Sciences, Adam Smith Building, 40 Bute Gardens, Glasgow, G12 8RT, United Kingdom.

philip.leifeld@glasgow.ac.uk.

# 1 Latent Space Models and Network Inference

Several models for inferential network analysis have been advanced in the last fifteen years, among them exponential random graph models (ERGM) (Cranmer and Desmarais 2011) and latent space models (LSM) (Hoff et al. 2002).

Latent space models have been primarily developed for two purposes: inference on the exogenous part of the network-generating process and scaling of actors according to their structural positions. For network inference, one "controls" for the dependencies inherent in the network by relegating them to a latent social space. On top, parameters for exogenous covariates are estimated and interpreted substantively. That is, "once the higher-order dependencies are taken into account, the dyadic data can be analyzed by techniques such as regression that assume the data are independent of one another" (Ward et al. 2007).

In this paper, I argue that this important purpose of LSMs can suffer from severe interpretation problems in many plausible research settings because the definition of the latent space is "over-sensitive." In particular, interpretation of coefficients for exogenous covariates in Euclidean LSMs is infeasible because variation that ought to be explained by covariates is often picked up by the latent space, which leads to estimates with flipped signs compared to other common network models. In this contribution, I examine the conditions under which this problem occurs using Monte Carlo simulations and with the aid of exponential random graph models.

Two popular specifications of the latent space exist: a Euclidean space and bilinear space (Hoff et al. 2002). In this contribution, I focus exclusively on Euclidean LSMs as described in the original exposition by Hoff et al. (2002) and as implemented in the `latentnet` package (Krivitsky and Handcock 2008) as part of the popular `statnet` suite of packages (Handcock et al. 2008) for `R` (`R` Core Team 2015). Note, however, that other specifications like the bilinear model are also popular in political science (Hoff and Ward 2004). I also limit the contribution made here to binary outcome networks like collaboration between political actors because binary network data are most common and because this makes the analysis easily replicable using competing network methods.

In the following sections, I first present a theoretical argument on the nature of the bias, then conduct Monte Carlo simulation experiments to demonstrate the extent of the problem, and before concluding, I present a brief empirical example where a Euclidean LSM would lead to wrong conclusions about the exogenous part of the data-generating process (henceforth DGP).

## 2 Theory

In latent space models, the observations $N_{ij}$ in network matrix $N$ are conditionally independent given dyadic covariates $\mathbf{x}_{ij}$, their parameters $\boldsymbol{\theta}$, and the positions $z_i$ and $z_j$ in the latent social space (for details on this exposition, see Hoff et al. 2002). Thus the probability of observing the network is the product of the individual probabilities for each dyadic observation, given the covariates, their estimates, and the latent space positions of $i$ and $j$:

$$P(N|Z, X, \theta) = \prod_{i \neq j} P(N_{ij}|z_i, z_j, \mathbf{x}_{ij}, \theta) \quad (1)$$

In other words, the dependencies between different dyads $N_{ij}$ are relegated to the coordinates of individual nodes $i$ and $j$ in the latent space, and these positions are estimated along with the parameters for the covariates.

The dyadic tie probabilities $\eta_{ij}$ can be expressed as log odds, which results in a logistic regression equation. The probability of each pair of nodes $N_{ij}$ being tied is a function of the distance between $i$ and $j$ in the social space, which is the Euclidean distance between the positions $z_i$ and $z_j$ over the $k$ dimensions of the latent space:

$$\eta_{ij} = \log \text{odds}(N_{ij} = 1|z_i, z_j, \mathbf{x}_{i,j}, \alpha, \beta) = \alpha + \beta' \mathbf{x}_{i,j} - |z_i - z_j| \quad (2)$$

The distances must satisfy the triangle inequality,

$$d_{ij} \leq d_{ik} + d_{kj} \quad \forall \{i, j, k\}, \tag{3}$$

meaning that any direct distance between $i$ and $j$ should not be larger than distances over indirect paths between $i$ and $j$ through other nodes $k$. This condition essentially constitutes the latent social space. The goal of the estimation process in an LSM is primarily to find a distance matrix $D$ that satisfies this condition and thus maps the dependence structure in the network matrix to a matrix of distances where adjacent nodes in the network have smaller distances than non-adjacent nodes. The positions of the nodes in the latent space are later derived from $D$ by means of multidimensional scaling or other techniques.

The triangle inequality therefore captures the notion of adjacency of nodes in the network, including local dependency structures such as reciprocity and transitivity. However, it is also the source of bias for exogenous covariates.

In most empirical networks, centrality is distributed unevenly. Some actors are connected to many nodes while most actors are connected to few. In LSMs, the triangle inequality requires that these central nodes are indeed placed at the center of the latent space because they have minimal average latent or path distances to the majority of other nodes. Being connected makes them central in the latent space.

At the same time, exogenous nodal covariates are often correlated with the centrality of nodes. In fact, testing for nodal covariate effects is equivalent to testing whether nodes that have a specific attribute also have more ties in the network, which is equivalent to their centrality. In other words, modeling exogenous nodal covariate effects means testing whether actors with certain attributes are more central in the network.

By design, this introduces a correlation between the attributes of nodes and their position in the latent space. More precisely, scoring high on an attribute that is indeed relevant for network formation causes a node to have small average distances to other nodes and thus a central position in the latent space.

As both the latent distances and the nodal attributes are included in the same regression equation, one effect cancels out the other effect. This leads to a situation where the sign of a coefficient for an exogenous covariate is reversed due to the presence of the collinear latent distances.

While this reversed direction of significant nodal covariate effects is not problematic for estimation and fit of the model per se, it is delusive whenever one wants to interpret covariate effects substantively.

# 3 Monte Carlo Simulations

I conceive two Monte Carlo simulation experiments in order to illustrate the nature of the problem. The first experiment demonstrates how inclusion of a covariate in the DGP leads to bias. I use Markov Chain Monte Carlo methods to simulate 50 artificial networks per condition based on an ERGM equation (Morris et al. 2008; Cranmer and Desmarais 2011) using different model specifications. All simulated networks contain 30 nodes. An ERGM and a Euclidean LSM with two dimensions and the same equation as in the DGP are estimated each time, and the coefficients for the covariate are compared between models (see first panel of Figure 1). Condition (a) includes an `edges` term ($h_{\text{edges}} = \sum_{i \neq j} N_{ij}$) with a parameter of $-3.5$ and a target-specific nodal covariate ($h_{\text{nodeicov}} = \sum_{i \neq j} N_{ij} \vec{x}_j$) with a parameter value of 2. The covariate is randomly sampled from a standard normal distribution. Condition (b) replaces the `nodeicov` term by a sender-specific covariate ($h_{\text{nodeocov}} = \sum_{i \neq j} N_{ij} \vec{x}_i$). Condition (c) uses a relational covariate instead ($h_{\text{edgecov}} = \sum_{i \neq j} N_{ij} X_{ij}$). Condition (d) is the same as specification (a) but with additional `random sender` and `random receiver` effects in the latent space estimation and an additional `in-2-star` term ($h_{\text{istar}(2)} = \sum_{j \neq k} N_{ji} N_{ki}$) in the ERGM estimation to check whether an explicit control for popularity in the ERGM

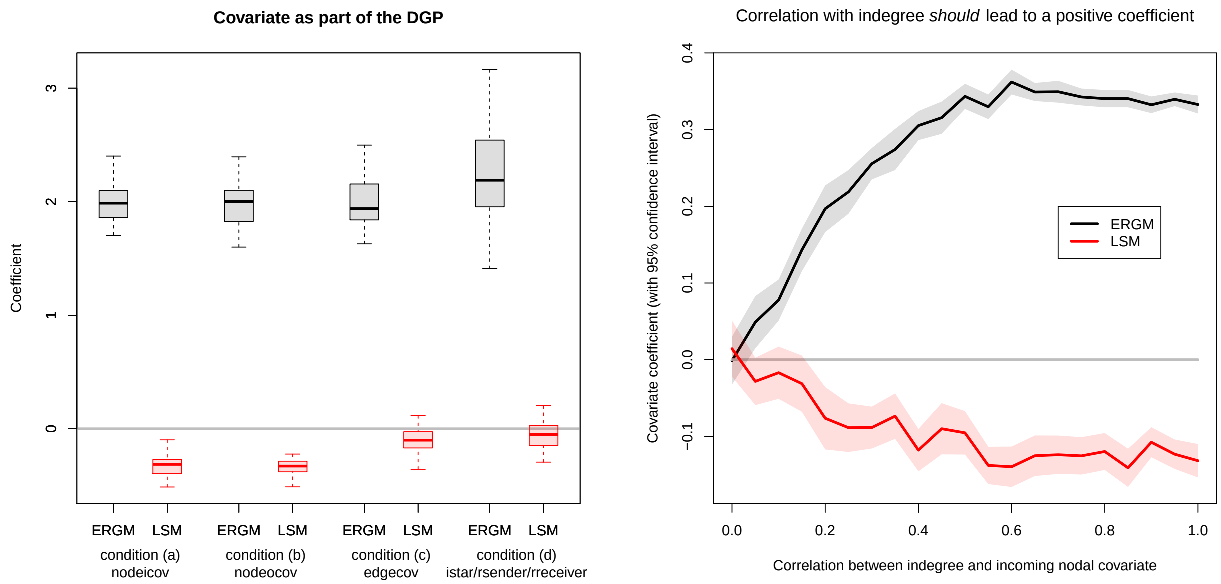


Figure 1: Simulation 1 (left panel): When a random covariate is part of the DGP, its coefficient of 2.0 can be recovered by the ERGM, but not by the LSM under various conditions. Simulation 2 (right panel): With increasing ex-post correlation between a target-specific covariate and the degree centrality of the target nodes, the LSM coefficient for the covariate reverses its direction while the ERGM shows a plausible magnitude.

introduces a similar bias as in the LSM; this is not the case. The results indicate that the ERGM recovers the original coefficient of 2 while the coefficient for the covariate is biased downward in the LSM under various conditions, even when random activity and popularity effects are included. This indicates that a significant covariate effect (as incorporated directly in the DGP here) leads to severe bias to the point where the sign is reversed.

The second experiment induces this bias ex post facto. 50 Bernoulli random graphs with an `edges` parameter of $-2$ (density $= 0.12$) are simulated at each step. The covariate is not part of the DGP, but it is included in the estimation equations for the ERGM and the LSM. This leads to unbiased coefficients of 0 in both models when the covariate is not correlated with the DGP. A Cholesky decomposition is employed to manipulate the covariate such that it is correlated with the indegree centrality scores of the target nodes. With increasing correlation (at intervals of 0.05), bias is experimentally induced (second panel of Figure 1). One should expect a positive coefficient (as featured in the ERGM) because the

| | ERGM | LSM |
|---|---|---|
| Edges | $-4.99\ (1.16)^{***}$ | $-1.63\ [-2.08;\ -1.17]^{*}$ |
| Preference similarity (edgecov) | $0.11\ (0.07)$ | $0.13\ [-0.03;\ 0.27]$ |
| Interest group homophily (nodematch) | $1.04\ (0.30)^{***}$ | $0.75\ [0.06;\ 1.45]^{*}$ |
| Alter = government actor (nodeicov) | $0.59\ (0.18)^{**}$ | $-0.30\ [-0.71;\ 0.07]$ |
| Ego = scientific actor (nodeocov) | $0.07\ (0.22)$ | $-0.82\ [-1.31;\ -0.36]^{*}$ |
| Number of shared forums (edgecov) | $0.30\ (0.05)^{***}$ | $0.49\ [0.40;\ 0.59]^{*}$ |
| Scientific communication (edgecov) | $2.86\ (0.64)^{***}$ | $2.22\ [1.40;\ 3.18]^{*}$ |
| Influence attribution (edgecov) | $0.93\ (0.18)^{***}$ | $0.09\ [-0.28;\ 0.45]$ |
| Reciprocity (mutual) | $0.81\ (0.25)^{***}$ | |
| GWESP ($\alpha = 0.1$) | $2.38\ (0.99)^{*}$ | |
| GDESP ($\alpha = 0.1$) | $-0.13\ (0.05)^{**}$ | |

$^{***}p < 0.001$, $^{**}p < 0.01$, $^{*}p < 0.05$ (or 0 outside the confidence interval).

Table 1: ERGM and LSM replication of Leifeld and Schneider (2012). Several covariates that are correlated with centrality are severely biased in the LSM (see highlighted rows).

covariate is increasingly positively correlated with the DGP. However, the LSM yields an increasingly negative coefficient. A similar pattern occurs with relational covariates and a correlation with the dependent variable (not displayed). This illustrates how the increasing correlation between covariate and the latent space leads to biased interpretation, all else being experimentally controlled.

# 4 Empirical Example

To demonstrate that this problem can severely affect empirical research, I replicate the ERGM analysis of Leifeld and Schneider (2012) using an LSM. The authors explain information exchange between 30 actors in a policy domain using several endogenous model terms and exogenous covariates.

One of the covariates is a categorical attribute called `Alter = government actor`, which indicates whether the target node is a state actor. The argument is that government actors are attractive lobbying targets because they have decision-making authority, therefore other actors seek access and establish communication ties to them. This leads to a positive expectation for the coefficient.

Table 1 shows the replication of the ERGM from the original article, which does display a positive coefficient, and the LSM replication without the endogenous model terms as they are supposed to be captured by the latent space. The LSM coefficient is negative, in contrast to the ERGM coefficient. In this example, one would erroneously conclude that government actors are unattractive communication targets and therefore reject the hypothesis.

Similarly, a relational covariate called `Influence attribution` yields a negative coefficient although one would expect influential actors to be attractive communication and lobbying targets (as confirmed by the ERGM). Actors who are central in the network are also influential and are government actors. These actors have small average distances to all other nodes in the latent space, and simultaneous inclusion of the covariates and these collinear distances makes the covariate estimates turn negative or insignificant.

# 5 Conclusion

In this contribution, I outlined a theory of covariate bias in Euclidean latent space models. I presented two Monte Carlo simulation experiments. In one of them, I could demonstrate that covariates as part of the DGP lead to biased estimates even when the same process is estimated and random sender and receiver effects are included. In the other experiment, I induced the bias according to the aforementioned theory in order to show that the correlation between covariate and nodal centrality is indeed responsible for the problem. A subsequent empirical replication exercise illustrated the real-world relevance of the issue.

Future research should test whether Euclidean LSMs with different parameters (e. g., dimensionality, model terms) and bilinear models are similarly affected. In particular, a respecification of Equation 2 using inner products as suggested by Hoff et al. (2002) in order to account for sender activity and/or receiver popularity,

$$\log \text{odds}(N_{ij} = 1|z_i, z_j, \mathbf{x}_{ij}, \alpha, \beta) = \alpha + \beta'\mathbf{x}_{ij} + \frac{z_i' z_j}{|z_j|}, \quad (4)$$

may potentially rectify the problem. In any case, users of Euclidean latent space models as implemented in current software solutions (Krivitsky and Handcock 2008) should be cautious with regard to substantive interpretation of exogenous effects.

# References


Cranmer, S. J. and Desmarais, B. A. (2011). Inferential network analysis with exponential random graph models. *Political Analysis*, 19(1):66–86.

Handcock, M. S., Hunter, D. R., Butts, C. T., Goodreau, S. M., and Morris, M. (2008). statnet: Software tools for the representation, visualization, analysis and simulation of network data. *Journal of Statistical Software*, 24(1):1–11.

Hoff, P. D., Raftery, A. E., and Handcock, M. S. (2002). Latent space approaches to social network analysis. *Journal of the American Statistical Association*, 97(460):1090–1098.

Hoff, P. D. and Ward, M. D. (2004). Modeling dependencies in international relations networks. *Political Analysis*, 12(2):160–175.

Krivitsky, P. N. and Handcock, M. S. (2008). Fitting position latent cluster models for social networks with latentnet. *Journal of Statistical Software*, 24(2):1–23.

Leifeld, P. and Schneider, V. (2012). Information exchange in policy networks. *American Journal of Political Science*, 56(3):731–744.

Morris, M., Handcock, M. S., and Hunter, D. R. (2008). Specification of exponential-family random graph models: Terms and computational aspects. *Journal of Statistical Software*, 24(4):1–24.

R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Ward, M. D., Siverson, R. M., and Cao, X. (2007). Disputes, democracies, and dependencies: A reexamination of the Kantian peace. *American Journal of Political Science*, 51(3):583–601.

# Online Appendix

## Replication Code

```
library("statnet")    # tested with version 2014.2.0
library("latentnet")  # tested with version 2.7.0
library("ergm")       # tested with version 3.2.4
library("texreg")     # tested with version 1.34.7
set.seed <- 12345
nsim <- 50
n <- 30

# Simulation experiment 1
results <- matrix(NA, ncol = 8, nrow = nsim)
colnames(results) <- rep(c("ERGM", "LSM"), 4)
for (i in 1:nsim) {
  print(i)
  nodalcov <- rnorm(n, 0, 1)

  # condition (a): nodeicov
  nodalcov.mat <- matrix(rep(nodalcov, n), nrow = n, byrow = TRUE)
  nw <- simulate.formula(network(n) ~ edges + edgecov(nodalcov.mat),
      coef = c(-3.5, 2))
  results[i, 1] <- coef(ergm(nw ~ edges + edgecov(nodalcov.mat)))[2]
  lsm <- ergmm(nw ~ euclidean(2) + edgecov(nodalcov.mat))
  results[i, 2] <- summary(lsm)$pmean$coef.table[2, 1]

  # condition (d): nodeicov and rsender, rreceiver, istar(2)
  results[i, 7] <- coef(ergm(nw ~ edges + edgecov(nodalcov.mat) + istar(2)))[2]
  lsm <- ergmm(nw ~ euclidean(2) + edgecov(nodalcov.mat) + rsender + rreceiver)
```

```
  results[i, 8] <- summary(lsm)$pmean$coef.table[2, 1]

  # condition (b): nodeocov
  nodalcov.mat <- matrix(rep(nodalcov, n), nrow = n, byrow = FALSE)
  nw <- simulate.formula(network(n) ~ edges + edgecov(nodalcov.mat),
      coef = c(-3.5, 2))
  results[i, 3] <- coef(ergm(nw ~ edges + edgecov(nodalcov.mat)))[2]
  lsm <- ergmm(nw ~ euclidean(2) + edgecov(nodalcov.mat))
  results[i, 4] <- summary(lsm)$pmean$coef.table[2, 1]

  # condition (c): edgecov
  nodalcov <- rnorm(n * n, 0, 1)
  nodalcov.mat <- matrix(nodalcov, nrow = n)
  nw <- simulate.formula(network(n) ~ edges + edgecov(nodalcov.mat),
      coef = c(-3.5, 2))
  results[i, 5] <- coef(ergm(nw ~ edges + edgecov(nodalcov.mat)))[2]
  lsm <- ergmm(nw ~ euclidean(2) + edgecov(nodalcov.mat))
  results[i, 6] <- summary(lsm)$pmean$coef.table[2, 1]
}

pdf("sim1.pdf")
boxplot(results, at = c(1, 2.5, 4.5, 6, 8, 9.5, 11.5, 13), outline = FALSE,
    border = "#FFFFFFFF", main = "Covariate as part of the DGP",
    ylab = "Coefficient")
axis(1, at = c(1.75, 5.25, 8.75, 12.25), tick = FALSE, padj = 1.5,
    labels = c("condition (a)\nnodeicov", "condition (b)\nnodeocov",
    "condition (c)\nedgecov", "condition (d)\nistar/rsender/rreceiver"))
lines(c(0, 14), c(0, 0), col = "gray", lwd = 3)
boxplot(results, at = c(1, 2.5, 4.5, 6, 8, 9.5, 11.5, 13), outline = FALSE,
    add = TRUE, border = c("black", "red"), col = c("#00000022", "#FF000022"))
dev.off()


# Simulation experiment 2
nsim <- 50
n <- 30
correlation <- seq(0, 1, 0.05)
ergm.coef <- matrix(NA, nrow = nsim, ncol = length(correlation))
lsm.coef <- ergm.coef
for (i in 1:nsim) {
  print(i)
```

```
  for (j in 1:length(correlation)) {
    nw <- simulate.formula(network(n) ~ edges, coef = -2)
    ideg <- degree(nw, cmode = "indegree")
    nodalcov <- rnorm(n, 0, 1)
    nodalcov <- correlation[j] * ideg + sqrt(1 - correlation[j]^2) * nodalcov
    nodalcov.mat <- matrix(rep(nodalcov, n), nrow = n, byrow = TRUE)
    ergm.coef[i, j] <- coef(ergm(nw ~ edges + edgecov(nodalcov.mat)))[2]
    lsm <- ergmm(nw ~ euclidean(2) + edgecov(nodalcov.mat))
    lsm.coef[i, j] <- summary(lsm)$pmean$coef.table[, 1][2]
  }
}

ergm.mean <- numeric(ncol(ergm.coef))
ergm.ci.l <- numeric(ncol(ergm.coef))
ergm.ci.u <- numeric(ncol(ergm.coef))
lsm.mean <- numeric(ncol(lsm.coef))
lsm.ci.l <- numeric(ncol(lsm.coef))
lsm.ci.u <- numeric(ncol(lsm.coef))

q <- qnorm(0.975)
n <- nrow(ergm.coef)
for (i in 1:ncol(ergm.coef)) {
  m <- mean(ergm.coef[, i])
  ergm.mean[i] <- m
  s <- sd(ergm.coef[, i])
  ergm.ci.l[i] <- m - ((q * s) / sqrt(n))
  ergm.ci.u[i] <- m + ((q * s) / sqrt(n))

  m <- mean(lsm.coef[, i])
  lsm.mean[i] <- m
  s <- sd(lsm.coef[, i])
  lsm.ci.l[i] <- m - ((q * s) / sqrt(n))
  lsm.ci.u[i] <- m + ((q * s) / sqrt(n))
}

plot.min <- min(c(ergm.ci.l, lsm.ci.l), na.rm = TRUE)
plot.max <- max(c(ergm.ci.u, lsm.ci.u), na.rm = TRUE)

pdf("sim2.pdf")
plot(correlation, ergm.mean, type = "l", ylim = c(plot.min, plot.max), lwd = 3,
    ylab = "Covariate coefficient (with 95% confidence interval)",
```

```
    xlab = "Correlation between indegree and incoming nodal covariate",
    main = substitute(paste("Correlation with indegree ", italic("should"),
    " lead to a positive coefficient")))
lines(correlation, rep(0, length(correlation)), col = "gray", lwd = 3)
polygon(c(correlation[!is.na(ergm.ci.l)], rev(correlation[!is.na(ergm.ci.u)])),
    c(ergm.ci.l[!is.na(ergm.ci.l)], rev(ergm.ci.u[!is.na(ergm.ci.u)])),
    col = "#00000022", border = NA)
lines(correlation, lsm.mean, type = "l", ylim = c(plot.min, plot.max), lwd = 3,
    col = "red")
polygon(c(correlation, rev(correlation)), c(lsm.ci.l, rev(lsm.ci.u)),
    col = "#FF000022", border = NA)
legend(0.7, 0.2, col = c("black", "red"), legend = c("ERGM", "LSM"), lwd = 3)
dev.off()


# Empirical replication example
# (download CSV files: http://hdl.handle.net/1902.1/17004)
comm <- read.table(file = "committee.csv", header = TRUE, row.names = "label",
    sep = ";")
infrep <- read.table(file = "infrep.csv", header = TRUE, row.names = "label",
    sep = ";")
pol <- read.table(file = "pol.csv", header = TRUE, row.names = "label",
    sep = ";")
scifrom <- read.table(file = "scifrom.csv", header = TRUE, row.names = "label",
    sep = ";")
scito <- read.table(file = "scito.csv", header = TRUE, row.names = "label",
    sep = ";")
intpos <- read.table(file = "intpos.csv", header = TRUE, row.names = "label",
    sep = ";")
types <- read.table(file = "orgtypes.csv", header = TRUE, row.names = "label",
    sep = ";")

sci <- as.matrix(scito) * t(as.matrix(scifrom)) # Equation 1 in the paper
prefsim <- dist(intpos, method = "euclidean", diag = FALSE, upper = TRUE)  # E 2
prefsim <- max(prefsim) - prefsim  # Equation 3
prefsim <- as.matrix(prefsim)
committee <- crossprod(as.matrix(comm), as.matrix(comm))  # Equation 4
diag(committee) <- 0
types <- as.character(types[, 1])
pol <- as.matrix(pol)
infrep <- as.matrix(infrep)
```

```
nw.pol <- network(pol) # political/stratgic information exchange
set.vertex.attribute(nw.pol, "orgtype", types)

model2.ergm <- ergm(nw.pol ~ edges + edgecov(prefsim) + nodemix("orgtype",
    base = -7) + nodeifactor("orgtype", base = -1) + nodeofactor("orgtype",
    base = -5) + edgecov(committee) + edgecov(sci) + edgecov(infrep) + mutual +
    gwesp(0.1, fixed = TRUE) + gwdsp(0.1, fixed = TRUE), eval.loglik = TRUE,
    check.degeneracy = TRUE, control = control.ergm(seed = 12345))

model2.lsm <- ergmm(nw.pol ~ edges + edgecov(prefsim) +
    nodemix("orgtype", base = -7) + nodeifactor("orgtype", base = -1) +
    nodeofactor("orgtype", base = -5) + edgecov(committee) + edgecov(sci) +
    edgecov(infrep))

coef.names <- c(
    "Edges",
    "Preference␣similarity␣(edgecov)",
    "Interest␣group␣homophily␣(nodematch)",
    "Alter␣=␣government␣actor␣(nodeicov)",
    "Ego␣=␣scientific␣actor␣(nodeocov)",
    "Number␣of␣shared␣forums␣(edgecov)",
    "Scientific␣communication␣(edgecov)",
    "Influence␣attribution␣(edgecov)",
    "Reciprocity␣(mutual)",
    "GWESP␣($\\alpha␣=␣0.1$)",
    "GDESP␣($\\alpha␣=␣0.1$)"
)

texreg(list(model2.ergm, model2.lsm), custom.coef.names = coef.names,
    include.aic = FALSE, include.bic = FALSE, include.loglik = FALSE,
    single.row = TRUE, file = "table.tex", use.packages = FALSE,
    dcolumn = TRUE, booktabs = TRUE, label = "infex", caption =
    paste("ERGM␣and␣LSM␣replication␣of␣Leifeld␣and␣Schneider␣(2012).",
    "Several␣covariates␣that␣are␣correlated␣with␣centrality␣are␣severely",
    "biased␣in␣the␣LSM␣(see␣highlighted␣rows)."), custom.model.names =
    c("ERGM", "LSM"))


save(results, ergm.coef, lsm.coef, model2.ergm, model2.lsm,
    file = "simulations.RData")
```