econet: An R package for parameter-dependent network centrality measures

Marco Battaglini
Cornell University, EIEF and NBER

Valerio Leone Sciabolazza
University of Naples Parthenope

Eleonora Patacchini
Cornell University, EIEF

Sida Peng
Microsoft Research

Abstract

The R package **econet** provides methods for estimating parameter-dependent network centrality measures with linear-in-means models. Both nonlinear least squares and maximum likelihood estimators are implemented. The methods allow for both link and node heterogeneity in network effects, endogenous network formation and the presence of unconnected nodes. The routines also compare the explanatory power of parameter-dependent network centrality measures with those of standard measures of network centrality. Benefits and features of the **econet** package are illustrated using data from Battaglini and Patacchini (2018), which examine the determinants of US campaign contributions when legislators care about the behavior of other legislators to whom they are socially connected.

Keywords: network econometrics, heterogeneous peer effects, endogenous network formation, least-square estimators, maximum likelihood estimators, R.

1. Introduction

Since its inception, network analysis has mostly focused on the discovery of topological properties of network structures. This has changed dramatically over the past ten years. An emerging literature in economics has shown that network centrality measures, which were traditionally viewed as descriptive, have an interpretation within equilibrium models of behavior. The pioneer paper is Ballester, Calvó-Armengol, and Zenou (2006). This paper considers a model in which an agent's effort is triggered by the effort of his/her socially connected peers. It shows that the equilibrium outcome is a linear function of the agent's position in the network as measured by an indicator within the family of Katz-Bonacich centralities. Katz-Bonacich centralities (Katz 1953; Bonacich 1972, 1987) are network centrality measures that count all nodes that can be reached through a direct or indirect path, penalizing in different ways the contributions of distant nodes in determining a given node's centrality. The discount factor is captured by a parameter, thus making the measures of centrality parameter-dependent. The sociological literature has been treating this parameter as a nuisance parameter, and arbitrarily setting it to any value smaller than one. However, the contribution of Ballester *et al.* (2006) is to show that this parameter captures the strength of peer effects or social interactions that stem from the aggregation of dyadic peer influences. More specifically, the empirical counterpart of the Ballester *et al.* (2006) equilibrium condition is a linear model of social interactions, where the individual outcome is a linear function of the outcomes of the connected agents. The parameter capturing this influence is then used to measure the individual's importance in the network. Since then, a burgeoning empirical literature has used the linear-in-means model to prove that peer effects and the individual position in the social network play an important role in explaining many social and economic outcomes, including consumer behavior, voting patterns, job search, information diffusion, innovation adoption, international trade and risk sharing (see e.g. Jackson, Rogers, and Zenou 2017; Zenou 2016, for recent reviews)

Very recently Battaglini and Patacchini (2018) present a new theory of competitive vote-buying to study how interest groups allocate campaign contributions when legislators care about the behavior of other legislators to whom they are socially connected. The model provides an alternative microfoundation for network measures within the family of Katz-Bonacich centralities. This theory predicts that, in equilibrium, campaign contributions are proportional to a parameter-dependent measure of network centrality, similar to the one proposed by Ballester *et al.* (2006). While Ballester *et al.* (2006) is a purely theoretical paper, Battaglini and Patacchini (2018) test the theory with data from five recent U.S. Congresses. In doing so, it confronts a variety of empirical challenges. For example, the theories described above show the importance of combining the information on network centrality with additional information on characteristics of the agents, since these characteristics can magnify or reduce the role played by a central agent. Moreover, the theories provide a framework to study the role of network endogeneity. In fact, when agents are strategic in choosing their peers, and omitted variables (such as social skills) drive both agent's behavior and social connectedness, the estimation of the peer effects parameter might be flawed (Manski 1993).

The routines contains in the package **econet** for R (R Core Team 2017) allow for implementation of a number of variations of the linear-in-means model to obtain alternative centrality measures within the family of Katz-Bonacich centrality. Both nonlinear least squares (NLLS) and maximum likelihood (ML) estimators are provided. Several methods for dealing with the identification of network effects are implemented. Moreover, the **econet** package allows for comparison of the explanatory power of parameter-dependent network centrality measures with those of standard measures of network centrality (Wasserman and Faust 1994). As a result, **econet** expands the large set of tools available to R users interested in network analysis. Specifically, it has at least four merits. First, it complements the R packages implementing traditional centrality measures for binary networks, **igraph** (Csárdi 2018) and **sna** (Butts 2016), and weighted networks, **tnet** (Opsahl 2015), by introducing new eigensolutions-based techniques to rank agents' centrality. Second, whereas previous packages, such as **btergm** (Leifeld, Cranmer, and Desmarais 2018a), **hergm** (Schweinberger, Handcock, and Luna 2018), the **statnet** suite (Handcock *et al.* 2016), and **xergm** (Leifeld, Cranmer, and Desmarais 2018b), created environments for modeling the statistical processes underlying network formation, **econet** provides the first framework to investigate the socio-economic processes operating on networks (i.e. peer effects). Third, it completes the collection of functions for modeling spatial dependence in cross-sectional data provided by **spdep** (Bivand *et al.* 2018) and **splm** (Millo and Piras 2017), by allowing the users to: i) consider the presence of unconnected nodes, and ii) address network endogeneity. Finally, it equips the R archive with routines still unavailable in other commonly used software for the investigation of relational data, such as Matlab (The MathWorks Inc 2007), Pajek (Batagelj and Mrvar 2003), Python (Python Software Foundation

2018) and Stata (StataCorp. 2017). The example we use to showcase the functionality of our R package is taken from Battaglini and Patacchini (2018) and Battaglini, Leone Sciabolazza, and Patacchini (2018).

The rest of the paper is organized as follows. Section 2 briefly reviews the key elements for the estimation of parameter-dependent centralities, and presents a general taxonomy of the different approaches used to model the socio-economic processes operating on the network. Section 3 lays out various models and methods to deal with network endogeneity. Section 4 demonstrates the use of the main functions of the package **econet** to determine agent's centrality with examples. Section 5 concludes.

2. Network models of peer effects

Let $N = \{1, \dots, n\}$ be a finite set of agents in a network $g \equiv g_N$. We keep track of connections in the network g through its adjacency matrix $\boldsymbol{G} = [g_{ij}]$, where $g_{ij} = 1$ if i is connected to j $(j \neq i)$ and $g_{ij} = 0$ otherwise. We set $g_{ii} = 0$. For each network g with adjacency matrix $\boldsymbol{G} = [g_{ij}]$, the kth power of $\boldsymbol{G}$ given by $\boldsymbol{G}^k = \boldsymbol{G}^{(k \text{times})}_{\dots}\boldsymbol{G}$ keeps track of direct and indirect connections in g. More precisely, the (i,j)th cell of $\boldsymbol{G}^k$ gives the number of paths of length k in g between i and j. In particular, $\boldsymbol{G}^0 = \boldsymbol{I}$.

Definition 1 (Katz, 1953; Bonacich, 1987) *Given a vector $\boldsymbol{u} \in \mathbb{R}^n_+$, and $\phi \geq 0$ a small enough scalar, the vector of Katz-Bonacich centralities of parameter ϕ in network g is defined as:*

$$\boldsymbol{b}\left(g, \phi\right) = \left(\boldsymbol{I} - \phi \boldsymbol{G}\right)^{-1} \boldsymbol{u} = \sum_{p=0}^{\infty} \phi^p \boldsymbol{G}^p \boldsymbol{u}. \tag{1}$$

The first order necessary and sufficient condition for optimality in the behavioral model developed by Battaglini and Patacchini (2018), BP henceforth, can be written as:

$$\mathbf{y_r} = \alpha \cdot \left(\boldsymbol{I} - \phi \boldsymbol{G}\right)^{-1} + X_r^T \beta + \epsilon_r, \tag{2}$$

where $\mathbf{y_r}$ is the vector of outcomes for the n players in network r, X_r is a matrix collecting the characteristics of the agents and ϵ_r is a random error term. The coefficients α, ϕ and β are the parameters to estimate. Model 2 can be written as

$$\mathbf{y}_r = \alpha \cdot \boldsymbol{b}_1\left(g, \phi\right) + X_r^T \beta + \epsilon_r, \tag{3}$$

where $\boldsymbol{b}_1\left(g, \phi\right) = \boldsymbol{b}\left(g, \phi\right)$ from equation 1 when $\boldsymbol{u} = 1$.
For a sample with $\bar{r}$ networks, one can stack up the data by defining $y = (\mathbf{y}_1^T, \cdots, \mathbf{y}_{\bar{r}}^T)^T$, $\epsilon = (\boldsymbol{\epsilon}_1^T, \cdots, \boldsymbol{\epsilon}_{\bar{r}}^T)^T$, $\mathbf{b}(\phi) = \left(\boldsymbol{b}\left(g, \phi\right)^T, ..., \boldsymbol{b}\left(g, \phi\right)^T\right)^T$, $G = \text{diag}\{\boldsymbol{G}_r\}_{r=1}^{\bar{r}}$, and $X = (\boldsymbol{X}_1^T, ..., \boldsymbol{X}_{\bar{r}}^T)$.
For the entire sample, the model is:

$$y = \alpha \cdot \mathbf{b}(\phi) + X\beta + \epsilon \tag{4}$$

We extend model 4 by accounting for heterogeneity in network spillovers. Specifically, model 4 becomes:

$$y = \alpha[I - G(\phi I + \gamma \Lambda)]^{-1} 1 + X\beta + \epsilon \tag{5}$$

where $\Lambda = I \otimes z$ is a matrix with the values in the vector z on the diagonal, and all other values are 0. The vector z represents a given characteristic of the agents. Consistently, the term γ allows for the possibility that agents with different characteristics may be more or less susceptible to social spillovers.

The first order necessary and sufficient condition for optimality in the behavioral model developed by (Battaglini *et al.* 2018), BLP henceforth, is:

$$\mathbf{y}_r = (I - \phi G)^{-1} (\alpha + X_r^T \beta) + \epsilon_r \tag{6}$$

which can be rewritten as:

$$\mathbf{y}_r = \boldsymbol{b}_2 (g, \phi) + \epsilon_r \tag{7}$$

where $\boldsymbol{b}_2 (g, \phi) = \boldsymbol{b} (g, \phi)$ from (1) when $\boldsymbol{u} = (\alpha + \boldsymbol{X}_r^T \beta)$. Equation 7 shows that, in this case as well, the optimal behavior is proportional to a centrality measure within the Family 1. When we consider the case where the parameter ϕ associated with network externalities is not constant across agents, model 7 in matrix formulation becomes:

$$y = (I - \theta \Lambda G)^{-1} (\alpha + X\beta) + \epsilon \tag{8}$$

$$y = (I - \eta G \Lambda)^{-1} (\alpha + X\beta) + \epsilon \tag{9}$$

In equations 8 and 9, Λ is a diagonal matrix of ones. In equation 8, $\theta = \theta_0 + \theta_1 z$, where θ_0 is a rescaling factor, and θ_1 quantifies the interaction between the adjacency matrix G and the vector z. Consequently, θ_1 measures the extent to which the peers of an agent with a given characteristic are susceptible to her/his influence. Similarly, in equation 9, $\eta = \eta_0 + \eta_1 z$, where η_0 is a rescaling factor, and η_1 measures the extent to which an agent with a given characteristic z is more susceptible to the influence of her/his peers.

In addition, BLP also consider the possibility of heterogeneous links (rather than nodes). We consider the case in which agents belong to two different groups and interactions are different between and within groups. To allow for group effects, one can reorder the matrix G so that the first n_1 columns refer to agents in the first group, and the other $n_2 = n - n_1$ columns refer to agents in the second group. The matrix G can now be divided into four submatrices. The submatrices in the main diagonal (of dimensions $n_1 \cdot n_1$ and $n_2 \cdot n_2$) collect the interactions within groups, whereas the remaining two submatrices (of dimensions $n_1 \cdot n_2$ and $n_2 \cdot n_1$) collect interactions between groups. G can thus be decomposed in two $n \cdot n$ matrices G_{wit} and G_{btw}, with $G = G_{wit} + G_{btw}$. G_{wit} is a matrix that has the same top left and bottom right components as G and it is zero otherwise, and G_{btw} is a matrix that has the same bottom left and top right components as G and it is zero otherwise. As a result, model 7 becomes

$$y = (I - \phi_1 G_{wit} - \phi_2 G_{btw})^{-1} (\alpha + X\beta) + \epsilon \tag{10}$$

where ϕ_1 captures within-group spillovers, and ϕ_2 between-group spillovers. The taxonomy of models outlined above shows that centrality measures should be calculated according to the most appropriate behavioral model describing the agents' behaviors and the requested level of heterogeneity of agents and network links. In BP and BLP we bring these theories to the data and find support for their predictions in different contexts.

2.1. Estimation

Models 4 and 7 cannot be estimated by a simple OLS regression in which $\mathbf{y}$ represents the dependent variables and $\mathbf{b}(\phi)$ and X are the independent variables because $\mathbf{b}(\phi)$ is a nonlinear function of a parameter to be estimated, ϕ. We can, however, obtain estimates for α, ϕ and β using NLLS or ML.

The NLLS requests solving the nonlinear least-squares problem for equation 4 and 7. This task is performed by **econet** using the Levenberg-Marquardt algorithm implemented in the R package **minpack.lm** developed by Elzhov, Mullen, Spiess, and Bolker (2016).

The ML estimation requests ML functions that can be derived as follows, by assuming $\varepsilon \sim N(0, \sigma^2 I)$. For equation 4, assume $\varepsilon \sim N(0, \sigma^2 I)$, the log likelihood function is

$$\ln(L) = -\frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln\sigma^2 - \frac{1}{2}\left[y - (I - \phi G)^{-1}1 - X\beta\right]'\left[y - (I - \phi G)^{-1}1 - X\beta\right]/\sigma^2 \quad (11)$$

where n is the total sample size. Equation 5 will have the same form as equation 4, except substituting ϕG with $G(\phi I + \gamma\Lambda)$.

In a similar fashion, we consider the following maximum likelihood functions for equation 7, 8, 9 and 10. For equation 7, the log likelihood function corresponding to 12 is:

$$y = \frac{n}{2}\ln(2\pi) - \frac{1}{2}\ln|\Omega| - \frac{1}{2}\left[y - (I - \phi G)^{-1}X\beta\right]'\Omega^{-1}\left[y - (I - \phi G)^{-1}X\beta\right] \quad (12)$$

where $\Omega = \sigma^2(I - \phi G)^{-1}(I - \phi G')^{-1}$. Equation 8, 9 and 10 have the same likelihood function as above except substituting ϕG with $\theta\Lambda G$ or $\eta G\Lambda$ or $\phi_1 G_{wit} - \phi_2 G_{btw}$. When the network is sparse, isolates (i.e. individuals with no neighbors) will exist.In this case, we can modify the likelihood function to expedite the process. Rewrite

$$G = \begin{pmatrix} G^c_{n_1\times n_2} & \mathbf{0}_{n_1\times n_2} \\ \mathbf{0}_{n_2\times n_1} & \mathbf{0}_{n_2\times n_1} \end{pmatrix} \quad (13)$$

where n_1 is the size of all connected individuals and n_2 is the sizes of isolates.

$$(I - \phi G)^{-1} = \begin{pmatrix} (I - \phi G^c_{n_1\times n_2})^{-1} & \mathbf{0}_{n_1\times n_2} \\ \mathbf{0}_{n_2\times n_1} & I_{n_2\times n_1} \end{pmatrix} \quad (14)$$

Define $y = \left(y^{c'}, y^{u'}\right)'$ and $X = \left(X^{c'}, X^{u'}\right)'$. The likelihood function 12 can be transformed into:

$$\begin{aligned} y = {} & \frac{n_1}{2}\ln(2\pi) - \frac{1}{2}\ln|\Omega_1| - \frac{1}{2}\left[y^c - (I - \phi G^c)^{-1}X^c\beta\right]'{\Omega_1}^{-1}\left[y^c - (I - \phi G^c)^{-1}X^c\beta\right] + \\ & + \frac{n_2}{2}\ln(2\pi) - \frac{n_2}{2}\sigma^2 - \frac{1}{2}\left[y^u - X^u\beta\right]'\left[y^u - X^u\beta\right] \end{aligned} \quad (15)$$

Where $\Omega_1 = \sigma^2(I - \phi G^c)^{-1}(I - \phi G^c)^{-1}$.

Our package allows the adjacency matrix to be used as a direct input. This setting simplifies the data processing procedure compared with other R packages like **spdep** when dealing with social network data. Packages like **spdep** are designed for spatial data. In this environment, the network data is required to be imported as neighbor pairs. However, social network data differs from spatial data since isolates may exists (nodes that have no connection with all other nodes). Networks containing isolates are not compatible with the data structure for packages like **spdep**.

Our package not only provides a way to get around this problem but also proposes an efficient algorithm when including those isolates. Instead of inverting the entire adjacency matrix, we show in the algebra above that one only needs to invert the adjacency matrix for connected nodes. The likelihood function can be written as a sum of the spatial auto-regressive (SAR) likelihood function for connected nodes and a standard linear likelihood function of isolates (see equation 15).

3. Addressing network endogeneity

An important difference between the analysis of cross-sectional data dependence in social science and in other disciplines (such as geography, physics, statistics and computer science) is that in the social sciences the agents' cross-sectional allocations are the result of the choices of the agents as much as their behavior over the observed topologies. As a result, the data structure (network topology) can be endogenous, and inference neglecting this issue would be invalid. The simplest way to tackle the problem is to model network formation using a homophily model (see e.g. Fafchamps and Gubert 2007; Mayer and Puller 2008; Lai and Reiter 2017; Apicella, Marlowe, Fowler, and Christakis 2012; Attanasio, Barr, Cardenas, Genicot, and Meghir 2012) where the existence of a link between i and j, $g_{i,j}$, is explained by the distance between i and j in terms of characteristics, according to the model

$$g_{i,j} = \delta_o + \sum_l \delta_{l+1}|x_i^l - x_j^l| + u_{i,j} \tag{16}$$

where x_i^l for $l = 1, ...L$ are i's characteristics. As standard in the literature on dyadic link formation, the main assumption underlying model 16 is dyadic independence, i.e., the assumption that each agent's choices are not influenced by others' decisions, and therefore each link in the network occurs with the same probability.

Fafchamps, Leij, and Goyal (2010) and Graham (2015) suggest a variation of this model where this assumption can be tested. They suggest to include in the model the length of the shortest distance between i and j Fafchamps *et al.* (2010), or the number of shared friends between i and j Graham (2015).

Let us denote this additional variable, $\kappa_{i,j}$:

$$g_{i,j} = \delta_o + \delta_2\kappa_{i,j} + \sum_l \delta_{l+w}|x_i^l - x_j^l| + u_{i,j} \tag{17}$$

A statistically significant estimate of the parameter δ_2 would suggest that the presence of a link depends on the presence of links at path lengths higher than 2 (or on the number of shared friends), thus indicating a violation of the hypothesis of dyadic independence.

A different variation of model 16 is proposed by Graham (2016). The model in Graham (2016) accounts for agents' unobserved heterogeneity by adding fixed effects for agents i and j:

$$g_{i,j} = \delta_o + \delta_1\omega_{i,j} + \sum_l \delta_{l+1}|x_i^l - x_j^l| + \xi_i + \xi_j + u_{i,j}, \tag{18}$$

These models of network formation can help mitigate concerns about network endogeneity in linear models of peer effects, such as models 4 and 7, in one of two ways:i) they can be used to predict network connections on the basis of exogenous agents' characteristics and then use the predicted network topology as an instrument for the actual network structure, or ii) they can be used as a first step selection equation to derive a correction for network endogeneity à la Heckman.

BLP follow approach ii). Under the assumption that $\varepsilon_r = (\varepsilon_{1,r}, ..., \varepsilon_{n,r})'$ and $\eta_r = (u_{i,1,r}, ..., u_{i,n,r})'$ are jointly normal with $E(\epsilon_{i,r}^2) = \sigma_\epsilon^2$, $E(\epsilon_{i,r} u_{i,j,r}) = \sigma_{\epsilon\eta}$ for all $i \neq j$, $E(\eta_{i,j,r}\eta_{ik,r}) = \sigma_\eta^2 \; \forall j = k$, and $E(\eta_{i,j,r}\eta_{i,k,r}) = 0 \; \forall j \neq k$, the expected value of the error term conditional on the link formation is $E(\epsilon_{i,r} | \eta_{i,1,r}, \ldots, \eta_{i,n-1,r}) = \psi \cdot \sum_{j \neq i} \eta_{i,j,r}$, where $\psi = \sigma_{\epsilon\eta}/\sigma_\eta^2$. It follows that model 7 can be written as:

$$\mathbf{y}_r = [I - \phi \boldsymbol{G}]^{-1} \cdot [\alpha \cdot \mathbf{1} + \boldsymbol{X}_r \beta + \psi \xi_r + \varepsilon_r] \tag{19}$$

where $\xi_{i,r} = \sum_{j \neq i} u_{i,j,r}$ with $\xi_r = (\xi_{i,r}, ... \xi_{n,r})'$. The term $\psi \xi_r$ now captures the selection bias. The model can then be estimated with NLLS (see Section 2). Under these assumptions, approach i) and approach ii) should produce similar results since both deliver consistent estimators.

Under approach i), however, inference is complicated because the selectivity term ξ is a generated regressor from a previous estimation and no closed form solution is available for the NLLS standard errors estimates in a network context. For this reason, BLP use bootstrapped standard errors. Because of the inherent structural dependency of network data, the design of the resampling scheme for this bootstrap procedure needs special consideration. The residual vector in equation 19 does not contain i.i.d. elements, and one cannot sample with replacement from this vector. Then, BLP use the residual bootstrap procedure, which is common in spatial econometrics (see Anselin 1990), where resampling is performed on the structural errors, under the assumption that they are i.i.d. In practice, by having in hand the residual vector, $\boldsymbol{u}$, the vector of structural errors are derived from $\varepsilon = [I - \widehat{\phi} G] \boldsymbol{u}$.

An important challenge in using the two step procedures for both approach i) and ii) is finding exclusion restrictions, that is factors that affect network formation only.[1] This is notoriously a difficult task. BLP use an original instrument: connections between agents which are made during adolescence. Those connections are powerful predictors of social contacts later on in life, but clearly predetermined to decisions taken in the adulthood.

4. Implementing econet

We now turn the discussion to the implementation of the functions contained in **econet**. **Econet** implements the set of linear models of social interactions introduced in Section 2, where an agent's outcome is a function of the outcomes of the connected agents in the network. The routines provide both NLLS and ML estimators. The possible sources of endogeneity that could hinder the identification of a causal effect in the model can be addressed by implementing the two-step correction procedures described in Section 3. The estimated parameter capturing the impact of the social network on agent's performance is then used to measure the individual

[1] Technically, the two step model is identified even using exactly the same set of regressors in both stages since the dyad-specific repressors used in the first stage (the network formation stage) are expressed in absolute values of differences. These differences do not appear in the outcome equation. Identification is thus achieved by exploiting non-linearities specific to the network structure of our model. While this strategy has been used in the applied network literature (see e.g. Goldsmith-Pinkham and Imbens 2013; Hsieh and Fei 2016), it may be a tenous source of identification in some cases.

Model	Hypothesis	Equation	Centrality measure $\boldsymbol{b}(g, \phi)$	
Model A	`lim`	4	ϕ: homogeneous	$\boldsymbol{b}(g, \phi) = (I - \phi G)^{-1} 1$
	`het`	5	ϕ: heterogenous by node type	$\boldsymbol{b}(g, \phi) = [I - G(\phi I + \gamma \Lambda)]^{-1} 1$
Model B	`lim`	7	ϕ: homogeneous	$\boldsymbol{b}(g, \phi) = (I - \phi G)^{-1} 1$
	`het_l`	8	ϕ: heterogenous outgoing influence by node type	$\boldsymbol{b}(g, \phi) = (I - \theta \Lambda G)^{-1} 1$
	`het_r`	9	ϕ: heterogenous ingoing influence by node type	$\boldsymbol{b}(g, \phi) = (I - \eta G \Lambda)^{-1} 1$
	`par`	10	ϕ: heterogenous by link type	$\boldsymbol{b}(g, \phi) = (I - \phi_1 G_{wit} \phi_2 G_{btw})^{-1} 1$

Table 1: Field specification in `net_dep`

importance in the network, obtaining a weighted version of Katz-Bonacich centrality. Finally, the explanatory power of the parameter-dependent centrality can be compared with those of standard measures of network centrality. It is worth emphasizing that **econet** allows the inclusion of unconnected agents, for whom the Katz-Bonacich centrality is constant.

Specifically, **econet** provides four functions. The first one is `net_dep`, which allows one to estimate a model of social interactions and compute the relative weighted Katz-Bonacich centralities of the agents. Different behavioral models can be chosen (i.e. those provided by BP and BLP). Moreover, the hypothesis of homogenous or heterogenous spillovers can be tested. The second function is `boot`, which is built to obtain valid inference when the NLLS estimator with Heckman correction is used. The third function is `horse_race`, which allows one to compare the explanatory power of parameter-dependent centralities relative to other centrality measures. The forth function is `quantify`, and it is used to assess the effect of control variables in the framework designed by BLP.

4.1. Detailing the functions

The modeling choices presented so far are implemented by the function `net_dep`.

The first three arguments of this function are: i) `formula`, an object of class `formula` which specifies the independent variable and the controls; ii) `data`, an object of class `data.frame` containing the values of the variables included in *formula*, iii) `G`, an object of class `Matrix` where the generic element g_{ij} is used to track the connection between i and j in the social network. `G` can be unweighted (i.e. $g_{ij} = 1$ if i and j are connected, and 0 otherwise), or weighted (e.g., g_{ij} collects the intensity of the relations between i and j) and column-normalized. The matrix must be arranged in the same order of the data, and row and column names must indicate agents' ids.

The next two arguments in `net_dep`, `model` and `hypothesis`, are used to specify the model to be estimated through an object of class `character`, as documented in Table 1. Specifically, the argument `model` indicates the framework to be applied: i.e., it is set to `Model_A` or `Model_B` to implement respectively the framework by BP and BLP. The argument `hypothesis` is necessary to indicate whether peer effects are assumed to be homogeneous `model = "lim` or heterogeneous (`model = c("het", "het_l", "het_r")`).

The argument `z` is used to specify the source of heterogeneity for the peer effects parameter ϕ (i.e., the variable z in equation 5, 8, 9), or the groups in which the network should be partitioned when `model="par"`. Specifically, `z` is a numeric vector where the generic element i refers to agent's i characteristic (e.g. it takes 1 if i is a female, and 0 otherwise).

In order to correct for the potential bias arising from network endogeneity, we include in the function four arguments: i) `endogeneity`, a logical object equal to `TRUE` if `net_dep` should implement the two-step correction procedure as e.g. in model 19, and `FALSE` otherwise; ii) `correction`, an object of class `character` that is set to indicate whether `net_dep` should implement a Heckman correction (`correction = "heckman"`), or an instrumental variable approach (`correction = "iv"`); iii) `exclusion_restriction`, an object of class `Matrix` used to specify the matrix to instrument the endogenous network; iv) `first_step`, an object of class `character` which specifies the network formation model to be used in the first step of the procedure for both `correction = "heckman"` and `correction = "iv"`. This argument can be equal to `standard` (equation 16), `shortest` (equation 17, as in Fafchamps *et al.* 2010), (equation 17, as in Graham 2015), `fe` (equation 18), `degree`, where the difference in degree centrality between agents is an additional regressor of equation 16.

Finally, the argument `estimation` allows the user to choose the estimation technique. This is an object of class `character` that can be one of two options: `NLLS` (nonlinear least squares), which implements the Levenberg-Marquardt optimization algorithm for solving the nonlinear least-squares problem using the function `nlsLM` contained in the R package **minpack.lm** (Elzhov *et al.* 2016) or `MLE` (maximum likelihood), which uses the function `mle2` of the R package **bbmle** to implement the quasi-Newton method by Bolker (2017) for maximum-likelihood bound constrained optimization.

The output of `net_dep` consists of a list of three objects: i) the point estimates and relative standard errors of the model's parameters; ii) the vector of agents' network centrality; iii) the point estimates and relative standard errors of the parameters of the first stage model, if `endogeneity=TRUE`.

We provide below a list of examples to illustrate the functionality of `net_dep`. Since in all examples we reject the hypothesis of normality of the errors, the NLLS estimation method is used.

Exercise 1: Katz-Bonacich centrality with parameter constant across agents

In the first example, we estimate the association between a congressman's network centrality and the amount of dollars he received from interest groups to finance his electoral campaign for the 111th Congress using model 4. The variables used to control for the effect of legislators' characteristics are: party affiliation (`party`); gender (`gender`); chairmanship (`nchair`); whether or not the congressman has at least one connection in the network (`isolate`).

The network used for this exercise represents the connections between agents which are made during adolescence. The network is constructed using information on the educational institutions attended by the congressmen. Specifically, we assume that a tie exists between two congressmen if they graduated from the same institution within eight years of each other. We set a link between two congressmen, g_{ij}, to be equal to the number of schools they both attended within eight years of each other; then we row normalize the social weights so that $\sum_i g_{ij} = 1$ for any i. This analysis is a simplified version (in terms of both data and controls included in the model specification) of the analysis in BP.

```
R> library("econet")
R> # Select data from the 111h congress
R> data("a_db_alumni")
R> data("G_alumni_111")
```

```
R> db_model_A <- subset(a_db_alumni, time == 3)
R> G_model_A <- a_G_alumni_111
R> # Specify which variables are factors
R> are_factors <- c("party", "gender", "nchair", "isolate")
R> db_model_A[are_factors] <- lapply(db_model_A[are_factors] ,factor)
R> # Divide Dependent variable by 1e+06
R> db_model_A$PAC <- db_model_A$PAC/1e+06
R> # Specify formula
R> f_model_A <- formula("PAC ~ gender + party + nchair + isolate")
R> # Specify starting values
R> starting <- c(alpha = 0.47325, beta_gender1 = -0.26991,
+> beta_party1 = 0.55883, beta_nchair1 = -0.17409,
+> beta_isolate1 = 0.18813, phi = 0.21440)
R> # Run net_dep
R> lim_model_A <- net_dep(formula = f_model_A, data = db_model_A,
+> G = G_model_A, model = "model_A", estimation = "NLLS",
+> hypothesis = "lim", start.val = starting)
R> # Store results in different objects
R> lim_estimate_model_A <- lim_model_A[[1]]
R> lim_centrality_model_A <- lim_model_A[[2]]
R> # Print results
R> summary(lim_estimate_model_A)
```

```
Formula: PAC ~ alpha * solve_block(I - phi * G) %*% Ones + beta_gender1 * gender1 +
beta_party1 * party1 + beta_nchair1 * nchair1 + beta_isolate1 * isolate1

Parameters:
Estimate Std. Error t value Pr(>|t|)
alpha          0.47325    0.17969   2.634  0.00876 **
beta_gender1  -0.26991    0.10504  -2.570  0.01052 *
beta_party1    0.55883    0.08363   6.682  7.5e-11 ***
beta_nchair1  -0.17409    0.19004  -0.916  0.36016
beta_isolate1  0.18813    0.18196   1.034  0.30178
phi            0.21440    0.27005   0.794  0.42768
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.815 on 420 degrees of freedom

Number of iterations to convergence: 2
Achieved convergence tolerance: 1.49e-08
```

In this model, the estimate of the intercept α directly measures the impact of the network centrality. We find a positive effect of a legislator's centrality on campaign contributions, showing that more connected congressmen are likely to receive more attention from interest groups. The estimate of the spillover effect, assumed to be the same for all congressmen, is also positive (even though is not statistically significant in this example). As already mentioned, the values in the weighted Katz-Bonacich centrality can be extracted from the second object stored in the output of `net_dep` (e.g. `lim_centrality_Model_A`).

Let us now repeat the same exercise for model 7. We use a simplified version (in terms of both data and controls included in the model specification) of the analysis presented in BLP. Our goal here is to investigate the association of legislative networks with congressmen's legislative effectiveness (LES). Network ties are defined here as the number of bills that j cosponsored with i. Also in this case, we impose that $\sum_i g_{ij} = 1$. The underlying idea is that legislators' productivity is affected by the productivity of the other legislators with whom they interact. However, while alumni connections are formed during adolescence and can thus be reasonably assumed to be exogenous to a congressman's political activity, cosponsorships are instead endogenous, since legislators are clearly strategic in choosing with whom to cosponsor a bill. The function `net_dep` allows the users to control for network endogeneity using the Heckman correction procedures described in Section 2.1. The legislators' characteristics used in this context are the same used in the previous example except the variable `isolate`, since all congressmen have at least one cosponsorship link.

```
R> # Select data from the 111th congress
R> data("db_cosponsor")
R> data("G_alumni_111")
R> db_model_B <- subset(db_cosponsor, time == 3)
R> G_model_B <- G_cosponsor_111
R> G_exclusion_restriction <- G_alumni_111
R> # Specify which variables are factors
R> are_factors <- c("gender", "party", "nchair")
R> db_model_B[are_factors] <- lapply(db_model_B[are_factors] , factor)
R> # Specify formula
R> f_model_B <- formula("les ~gender + party + nchair")
R> # Specify starting values
R> starting <- c(alpha = 0.23952, beta_gender1 = -0.22024,
+> beta_party1 = 0.42947, beta_nchair1 = 3.09615,
+> phi = 0.40038, unobservables = 0.07714)
R> # Run net_dep
R> lim_model_B <- net_dep(formula = f_model_B, data = db_model_B,
+> G = G_model_B, model = "model_B", estimation = "NLLS",
+> hypothesis = "lim", endogeneity = TRUE, correction = "heckman",
+> first_step = "standard", exclusion_restriction = G_exclusion_restriction,
+> start.val = starting)
R> # Store results in different objects
R> lim_estimate_model_B <- lim_model_B[[1]]
R> lim_centrality_model_B <- lim_model_B[[2]]
R> lim_estimate_first_step_model_B <- lim_model_B[[3]]
R> # Print results
R> summary(lim_estimate_model_B)

Formula: les ~ solve_block(I - phi * G) %*% (alpha * Ones + beta_gender1 *
gender1 + beta_party1 * party1 + beta_nchair1 * nchair1 +
beta_unobservables * unobservables)
```

```
Parameters:
Estimate Std. Error t value Pr(>|t|)
alpha               0.23952    0.07130   3.359 0.000851 ***
beta_gender1       -0.22024    0.14052  -1.567 0.117787
beta_party1         0.42947    0.10111   4.247 2.65e-05 ***
beta_nchair1        3.09615    0.25951  11.931  < 2e-16 ***
phi                 0.40038    0.06136   6.525 1.90e-10 ***
beta_unobservables  0.07714    0.06138   1.257 0.209519
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.106 on 433 degrees of freedom

Number of iterations to convergence: 2
Achieved convergence tolerance: 1.49e-08
```

```
R> summary(lim_estimate_first_step_model_B)
```

```
Call:
lm(formula = formula_first_step, data = e_first_step)

Residuals:
Min        1Q    Median        3Q       Max
-0.008437 -0.001936 -0.000997  0.000791  0.149182

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)           1.481e-03  3.809e-05  38.887  < 2e-16 ***
exclusion_restriction  4.657e-03  4.129e-04  11.279  < 2e-16 ***
gender1              -6.638e-05  2.135e-05  -3.109  0.00188 **
party1                2.366e-03  1.940e-05 121.922  < 2e-16 ***
nchair1              -4.180e-04  3.436e-05 -12.163  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.004247 on 192277 degrees of freedom
Multiple R-squared:  0.07383,        Adjusted R-squared:  0.07381
F-statistic:  3832 on 4 and 192277 DF,  p-value: < 2.2e-16
```

The output about present estimation results of model 19 are presented, along with the estimates of the first step model 16 (i.e. `lim_estimate_first_step_model_B`). Because of the two-step procedure, standard errors of model 7 are bootstrapped using the function `boot`. This function takes the following arguments: i) `fit`, the first object of `net_dep`'s output; ii) `group`, a numeric vector used to specify if the resampling should be performed within specific groups; iii) `niter`, an object of class numeric which indicates the iterations of the bootstrap;

iv) `weights`, a logical object equal to `TRUE` if the object `fit` is estimated with the argument `to_weight` different from `NULL`, and `FALSE` otherwise.

```
R> boot_lim_estimate <- boot(fit = lim_estimate_model_B, hypothesis = "lim",
+> group = NULL, niter = 1000, weights = FALSE)
res <- data.frame(coefficient = coefficients(lim_estimate_model_B),
+> boot.Std.Error = boot_lim_estimate)
R> # Compute new t values
R> res[, "boot.t.value"] <- res[, "coefficient"] / res[, "boot.Std.Error"]
R> # Compute new p values
R> res[, "boot.p.value"] <- 2 * pt( - abs(res[, "boot.t.value"]),
+> df = nobs(lim_estimate_model_B) - nrow(res))
R> # Print results
R> res
```

```
coefficient boot.Std.Error boot.t.value boot.p.value
alpha               0.23951648     0.08724885     2.745211 6.298903e-03
beta_gender1       -0.22023753     0.14248594    -1.545679 1.229126e-01
beta_party1         0.42946569     0.12335344     3.481587 5.490851e-04
beta_nchair1        3.09614813     0.26463420    11.699728 1.153402e-27
phi                 0.40038484     0.06099259     6.564483 1.496393e-10
beta_unobservables  0.07714043     0.05978823     1.290228 1.976602e-01
```

The results of this second exercise show positive and significant network effects on the effectiveness of agents, meaning that congressmen benefit from their interactions with the colleagues conscripted to their own causes. In this simple example, network endogeneity does not seem to be a major concern, since the correlation between the unobservables of link formation and outcome equation (ϕ in model 7) is not statistically significant. The results show that the alumni network (`G_exclusion_restriction`) is an important predictor of the cosponsorship network (`G_Model_B`).

While in model 4 the estimated effect of network centrality is captured by one parameter α, in model 7 it requires further elaboration since it varies with individual characteristics. For the k-th covariate in model 7, if $\phi > 0$, the centrality's marginal effect is $(I - \varphi G)^{-1}(I\beta_k)$, which is a $n \cdot n$ matrix with its (i,j)-th element representing the effect of a change in the characteristic k for agent j on the outcome of agent i. The diagonal elements capture the direct effect of a marginal change in the characteristic k for agent i. The elements outside the diagonal instead capture the indirect effects, that is the effects on the outcome of i triggered by variation of the characteristic k in other agents. The direct effects are comparable to the OLS estimated effects without considering the network effects. The important difference in comparing the estimates of the covariates in the models with and without network effects is precisely that when $\phi > 0$, the marginal effect of the k-th covariate in model 7 is not just β_k but it also depends on the individual's position in the network (i.e.m on the individual's network centrality).

The function `quantify` allows the user to run this task. Specifically, it provides the estimated impacts of the agents' characteristics with network effects (which is equivalent to the estimated impacts of agents' network centrality by characteristic) and compares them with the OLS estimates. Because, as we said before, the marginal effects of characteristics are different for different agents, the function `quantify` reports the mean, standard deviation, maximum and minimum for both direct and indirect effects. It requires two arguments: i) `fit`, which takes the first object stored in `net_dep`, i.e. the estimates; ii) `Gn`, which takes the adjacency matrix passed to `net_dep` through the argument `G` to obtain the estimates.

```
R> quantify(fit = lim_estimate_model_B, Gn = G_model_B)

beta Direct_mean Direct_std Direct_max Direct_min Indirect_mean
1 -0.2202     -0.2205     0.0002     -0.2202    -0.2217       -0.0003
2  0.4295      0.4299     0.0003      0.4323     0.4295        0.0007
3  3.0961      3.0992     0.0025      3.1162     3.0961        0.0047
4  0.0771      0.0772     0.0001      0.0776     0.0771        0.0001
Indirect_std Indirect_max Indirect_min
1       0.0005       0.0000      -0.0149
2       0.0009       0.0290       0.0000
3       0.0068       0.2091       0.0000
4       0.0002       0.0052       0.0000
```

The estimation results show the mean, standard deviation, maximum and minimum for both direct and indirect effects. Perhaps unsurprisingly, it appears that the indirect effects are smaller than the direct effects.

As already mentioned, agents' centrality measures are stored in the second object of `net_dep`'s output, and it can be used for different applications. For example, we can use it to rank agents' positions in the Congress social space, or to study the centrality distribution in the Republican and the Democratic party, as we do in Figure 1. Figure 1 can be replicated using the following script:

```
R> library("ggplot2")
R> # Associate congressmen centrality to party affiliation
R> df <- data.frame(centrality = lim_centrality_model_B,
+> party = lim_estimate_model_B$data$party1)
R> # Specify party categories and colors
R> df[, "party"] <- ifelse(df[, "party"] == 0, "Republican", "Democrat")
R> df[, "colour"] <- ifelse(df[, "party"] == "Republican", "red", "blue")
R> # Plot the distributions
R> ggplot(data = df, aes(centrality), colour = df[, "colour"]) +
+> geom_histogram(binwidth = 0.25, aes(fill = factor(colour)), col=I("black"),
+> alpha=I(.7)) +
+> facet_grid(party ~.) + theme_bw() +
+> labs(x = "Parameter-Dependent Centrality", y = "Frequency") +
+> scale_fill_manual(values = c("blue", "red")) + theme(legend.position = "none")
```

Exercise 2: Katz-Bonacich centrality with heterogenous by node parameter

The code below estimates model 5 using gender as the relevant dimension of heterogeneity. Gender is a dummy variable which takes 1 if the legislator is a female, and 0 otherwise.

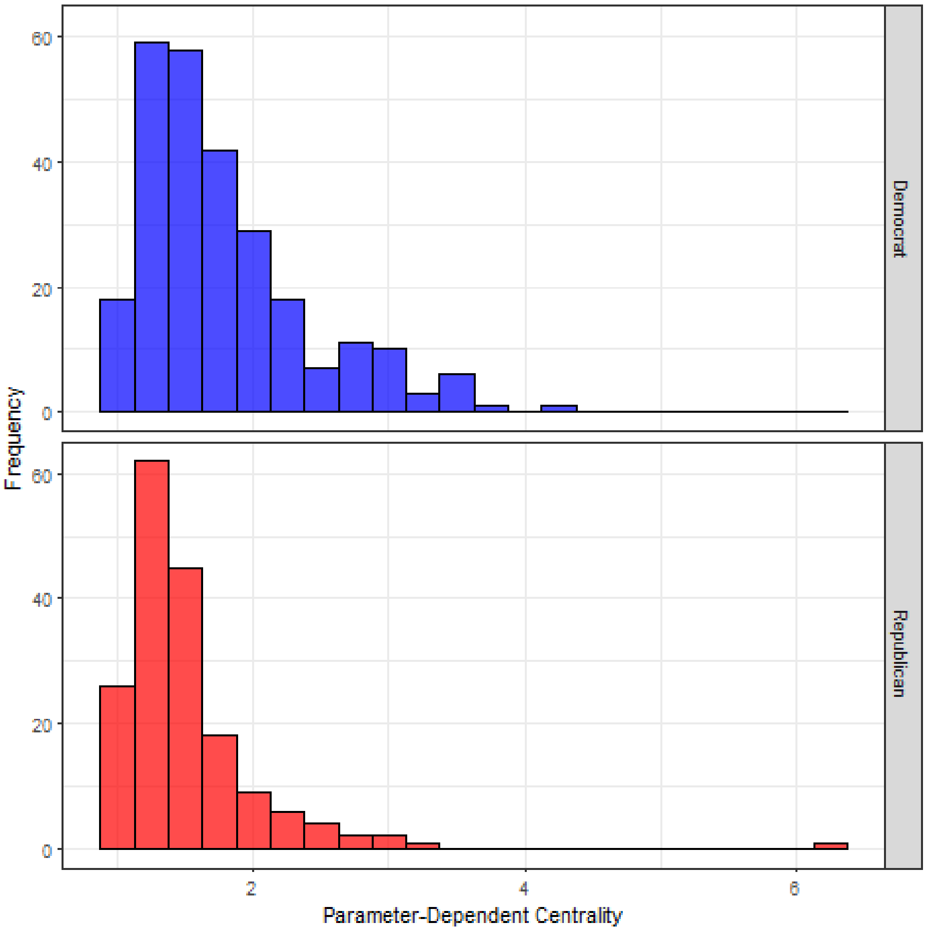

Figure 1: BLP model: Distribution of Parameter-Dependent Network Centrality.

```
R> # Create gender variable
R> z <- as.numeric(as.character(db_model_A[, "gender"]))
R> # Specify formula
R> f_het_model_A <- formula("PAC ~ party + nchair + isolate")
R> # Specify starting values
R> starting <- c(alpha = 0.44835, beta_party1 = 0.56004,
+> beta_nchair1 = -0.16349, beta_isolate1 = 0.21011,
+> beta_z = -0.26015, phi = 0.34212, gamma = -0.49960)
R> # Run net_dep
R> het_model_A <- net_dep(formula = f_het_model_A, data = db_model_A,
+> G = G_model_A, model = "model_A", estimation = "NLLS",
+> hypothesis = "het", z = z, start.val = starting)
R> # Store results in different objects
R> het_estimate_model_A <- het_model_A[[1]]
R> het_centrality_model_A <- het_model_A[[2]]
R> # Print results
R > summary(het_estimate_model_A)
```

```
Formula: PAC ~ alpha * solve_block(I - G %*% (phi * I + gamma * G_heterogeneity)) %*% Ones
beta_party1 * party1 + beta_nchair1 * nchair1 + beta_isolate1 * isolate1 + beta_z * z

Parameters:
Estimate Std. Error t value Pr(>|t|)
alpha          0.44835    0.17942   2.499   0.0128 *
beta_party1    0.56004    0.08342   6.713 6.21e-11 ***
beta_nchair1  -0.16349    0.18984  -0.861   0.3896
beta_isolate1  0.21011    0.17927   1.172   0.2418
beta_z        -0.26014    0.10655  -2.441   0.0150 *
phi            0.34212    0.25377   1.348   0.1783
gamma         -0.49960    0.34662  -1.441   0.1502
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.8138 on 419 degrees of freedom

Number of iterations to convergence: 2
Achieved convergence tolerance: 1.49e-08
```

The results show that in this simple example γ is not significant, suggesting that female congressmen do not have a different level influence than their male peers. Note that the weighted Katz-Bonacich centrality derived from the spillover effect are stored in the object `het_centrality_model_A`.

If we use equation 8 and 9, we distinguish between outgoing and incoming influence, that is if females are more (or less) able to influence and to be influenced by their peers respectively. The code below implements this analysis. For ease of exposition, we do not consider the possible endogeneity of the social network.

```
R> # Create gender variable
R> z <- as.numeric(as.character(db_model_B[, "gender"]))
R> # Specify formula
R> f_het_model_B <- formula("les ~ party + nchair")
R> # Specify starting values
R> starting <- c(alpha = 0.23952, beta_party1 = 0.42947,
+> beta_nchair1 = 3.09615, beta_z = -0.12749,
+> theta_0 = 0.42588, theta_1 = 0.08007)
R> # Run net_dep
R> het_model_B_l <- net_dep(formula = f_het_model_B, data = db_model_B,
+> G = G_model_B, model = "model_B", estimation = "NLLS",
+> hypothesis = "het_l", z = z, start.val = starting)
R> # Specify starting values
R> starting <- c(alpha = 0.04717, beta_party1 = 0.51713,
+> beta_nchair1 = 3.12683, beta_z = 0.01975,
+> eta_0 = 1.02789, eta_1 = 2.71825)
R> # Run net_dep
R> het_model_B_r <- net_dep(formula = f_het_model_B, data = db_model_B,
+> G = G_model_B, model = "model_B", estimation = "NLLS",
+> hypothesis = "het_r", z = z, start.val = starting)
R> # Store results in different objects
R> het_l_estimate_model_B <- het_model_B_l[[1]]
R> het_l_centrality_model_B <- het_model_B_l[[2]]
R> het_r_estimate_model_B <- het_model_B_r[[1]]
R> het_r_centrality_model_B <- het_model_B_r[[2]]
R> # Print results
R> summary(het_l_estimate_model_B)

Formula: les ~ solve_block(I - (theta_0 * I - theta_1 * G_heterogeneity) %*%
G) %*% (alpha * Ones + beta_party1 * party1 + beta_nchair1 * nchair1 + beta_z * z)

Parameters:
Estimate Std. Error t value Pr(>|t|)
alpha         0.22740    0.07584   2.998  0.00287 **
beta_party1   0.41382    0.10198   4.058 5.88e-05 ***
beta_nchair1  3.07797    0.26148  11.771  < 2e-16 ***
beta_z       -0.12749    0.21199  -0.601  0.54789
theta_0       0.42587    0.07418   5.741 1.77e-08 ***
theta_1       0.08007    0.12320   0.650  0.51611
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.108 on 433 degrees of freedom

Number of iterations to convergence: 4
Achieved convergence tolerance: 1.49e-08

R> summary(het_r_estimate_model_B)

Formula: les ~ solve_block(I - G %*% (eta_0 * I - eta_1 * G_heterogeneity)) %*%
(alpha * Ones + beta_party1 * party1 + beta_nchair1 * nchair1 + beta_z * z)

Parameters:
Estimate Std. Error t value Pr(>|t|)
alpha         0.04717    0.06867   0.687  0.49251
beta_party1   0.51713    0.09259   5.585 4.13e-08 ***
beta_nchair1  3.12683    0.25680  12.176  < 2e-16 ***
beta_z        0.01976    0.15098   0.131  0.89595
eta_0         1.02790    0.22400   4.589 5.85e-06 ***
eta_1         2.71825    0.91547   2.969  0.00315 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.099 on 433 degrees of freedom
```

```
Number of iterations to convergence: 1
Achieved convergence tolerance: 1.49e-08
```

The results shows that females do not seem to be able to influence their socially connected peers (θ is not significant), but they are helpful to their colleagues ($\eta_1 > 0$). In terms of network analysis, this implies that all else being equal, congressmen located close to female colleagues benefit from their position since they can leverage females to be more effective in their legislative activity. In this case, the weighted Katz-Bonacich centralities are stored in `het_l_centrality_model_B` and `het_r_centrality_model_B`.

Exercise 3: Katz-Bonacich centrality with heterogenous by link parameter

In this last exercise, we show how to explore the hypothesis that relations within and between parties might have a different impact on the congressmen's LES. The code below estimates model 10 where the within and between effects are shaped by party membership.

```
R> # Specify partition vector
R> z <- as.numeric(as.character(db_model_B[, "party"]))
R> # Specify partition vector
R> starting <- c(alpha = 0.242486, beta_gender1 = -0.229895,
+> beta_party1 = 0.42848, beta_nchair1 = 3.0959,
+> phi_within = 0.396371, phi_between = 0.414135)
R> # Run net_dep
R> party_model_B <- net_dep(formula = f_model_B,
+> data = db_model_B,
+> G = G_model_B, model = "model_B", estimation = "NLLS",
+> hypothesis = "par", z = z, start.val = starting)
R> # Store results in different objects
R> party_estimate_model_B <- party_model_B[[1]]
R> party_centrality_model_B <- party_model_B[[2]]
R> # Print results
R> summary(party_estimate_model_B)
```

```
Formula: les ~ solve_block(I - phi_within * G_within - phi_between * G_between) %*%
(alpha * Ones + beta_gender1 * gender1 + beta_party1 * party1 + beta_nchair1 * nchair1)

Parameters:
Estimate Std. Error t value Pr(>|t|)
alpha         0.24249    0.08988   2.698 0.007251 **
beta_gender1 -0.22990    0.14120  -1.628 0.104221
beta_party1   0.42848    0.12623   3.394 0.000751 ***
beta_nchair1  3.09590    0.26008  11.904  < 2e-16 ***
phi_within    0.39637    0.07306   5.425 9.66e-08 ***
phi_between   0.41414    0.24175   1.713 0.087420 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
Residual standard error: 1.108 on 433 degrees of freedom

Number of iterations to convergence: 1
Achieved convergence tolerance: 1.49e-08
```

The estimation results show that connections within one's own party and between parties are both significant for advancing a piece of legislation. The weighted Katz-Bonacich centrality can be found in the object `party_centrality_model_B`.

4.2. Centrality measure comparison

The R package **econet** allows also to evaluate the explanatory power of parameter-dependent centralities relative to other centrality measures, which depends on network topology only. More specifically, **econet** considers the following measures, which are computed using the R package `igraph` (Csárdi 2018): indegree centrality, outdegree centrality, degree centrality, betweenness centrality, incloseness centrality, outcloseness centrality, closeness centrality. It also reports eigenvector centrality, which is calculated using the R package **sna** (Butts 2016).[2] We then implement an augmented version of models 4 and 7 where we add one (or more) centrality measures in the matrix of individual characteristics X_r^T. By doing so we can run an horse race across different centrality measures.

The arguments of the function `horse_race` are similar to the `net_dep` ones. The different ones are: i) `centralities`, an object or a vector of class `character` specifying the names of the centrality measure(s) to be used; ii) `directed`, a logical object which is set to `TRUE` if the network is directed, and `FALSE` otherwise; iii) `weighted`, a logical object equal to `TRUE` if links between agents have weights, and `FALSE` otherwise; and iv) `normalization`, an object of class character which can be used to normalize centrality measures before the estimations. [3]

The output of this function is also similar to `net_dep`, i.e. a list containing the results of the main estimates in the first object, a `data.frame` listing the centrality measures in the second object, and the results of the first step estimation in the third object (if `endogeneity = TRUE`). An example of how this model is implemented and results are stored is provided in the following example, where we use a linear regression model where betweenness centrality is a regressor.

[2] Indegree centrality is the number of incoming links of one node; outdegree centrality is the number of outgoing links of one node; degree centrality is the sum of in and out degree; betweenness centrality is the number of times a node falls on the shortest path between two other nodes; incloseness centrality is the inverse of the average distance of one node from all the other nodes passing through incoming links; outcloseness centrality is the inverse of the average distance of one node from all the other nodes passing through outcoming links; closeness centrality is the sum of in and out closeness, and eigenvector centrality is proportional to the sum of the centrality of agent's neighbors (see Jackson 2010, for further details).

[3] The options available are: `NULL`, no normalization; `bygraph`, divide by the number of nodes in the network minus 1 (for degree and closeness) or the number of possible links in the network (betweenness), `bycomponent`, divide by the number of nodes in agent's component minus 1 (for degree and closeness) or the number of possible links in agent's component (betweenness); `bymaxgraph`, divide by the maximum centrality value in the network; bymaxgraph, divide by the maximum centrality value in agent's component.

```
R> # Specify starting values
R> starting <- c(alpha = 0.214094, beta_gender1 = -0.212706,
+> beta_party1 = 0.478518, beta_nchair1 = 3.09234,
+> beta_betweenness = 7.06287e-05, phi = 0.344787)
R> # Horse race
R> horse_model_B <- horse_race(formula = f_model_B, centralities = "betweenness",
+> directed = TRUE, weighted = TRUE, normalization = NULL,
+> data = db_model_B, G = G_model_B,
+> model = "model_B", estimation = "NLLS", start.val = starting)
R> # Store results in different objects
R> betweenness_estimate_model_B <- horse_model_B[[1]][[1]]
R> horse_estimate_model_B <- horse_model_B[[1]][[2]]
R> horse_centrality_model_B <- horse_model_B[[2]]
R> # Print results
R> summary(betweenness_estimate_model_B)

Call:
lm(formula = to_fit, data = data, weights = to_weight)

Residuals:
Min      1Q  Median      3Q     Max
-3.6051 -0.5665 -0.1877  0.2559 10.8641

Coefficients:
Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.300e-01  9.042e-02   3.650 0.000295 ***
gender1     -8.904e-02  1.445e-01  -0.616 0.538011
party1       7.054e-01  1.139e-01   6.194 1.36e-09 ***
nchair1      3.202e+00  2.644e-01  12.112  < 2e-16 ***
betweenness  1.851e-04  3.985e-05   4.645 4.51e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.133 on 434 degrees of freedom
Multiple R-squared:  0.3654,        Adjusted R-squared:  0.3595
F-statistic: 62.46 on 4 and 434 DF,  p-value: < 2.2e-16


> summary(horse_estimate_model_B)

Formula: les ~ solve_block(I - phi * G) %*% (alpha * Ones + beta_gender1 *
gender1 + beta_party1 * party1 + beta_nchair1 * nchair1 +
beta_betweenness * betweenness)

Parameters:
Estimate Std. Error t value Pr(>|t|)
alpha             2.141e-01  7.676e-02   2.789  0.00552 **
```

```
beta_gender1      -2.127e-01  1.408e-01  -1.511  0.13162
beta_party1        4.785e-01  1.105e-01   4.331 1.85e-05 ***
beta_nchair1       3.092e+00  2.593e-01  11.927  < 2e-16 ***
beta_betweenness  7.063e-05  4.561e-05   1.549  0.12219
phi                3.448e-01  7.101e-02   4.856 1.68e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.105 on 433 degrees of freedom

Number of iterations to convergence: 1
Achieved convergence tolerance: 1.49e-08
```

The results of this estimation show evidence that being able to broker connections in the network, as measured by betweenness centrality in the object `betweenness_estimate_model_B` is associated with higher legislative effectiveness. The effect disappears when we include network effects, that is when we add betweenness centrality as an additional regressor in the linear model of social interactions (equation 7). This suggests that Katz-Bonacich centrality is a more robust predictor of effectiveness in this context. Estimates are stored in the object `horse_estimate_model_B`, whereas centrality measures in the object `horse_centrality_model_B`.

5. Conclusions

We have described the key elements for the estimation of parameter-dependent centralities derived from equilibrium models of behavior, and discussed the use of the package **econet** for the implementation of such metrics. The methods described in the paper are derived from several modifications to the linear-in-means model - for which both nonlinear least squares and maximum likelihood estimators are provided - and they allow one to model both link and node heterogeneity in network effects, endogenous network formation and the presence of unconnected nodes. Furthermore, they provide the means to compare the explanatory power of parameter-dependent network centrality measures with those of standard measures of network centrality. A number of examples are used to walk the reader through the discussion and orientate the application of these methods to new potential directions of research.

References

Anselin L (1990). "Spatial dependence and spatial structural instability in applied regression analysis." *Journal of Regional Science*, **30**(2), 185–207. doi:10.1111/j.1467-9787.1990.tb00092.x.

Apicella CL, Marlowe FW, Fowler JH, Christakis NA (2012). "Social networks and cooperation in hunter-gatherers." *Nature*, **481**(7382), 497. doi:10.1038/nature10736.

Attanasio O, Barr A, Cardenas JC, Genicot G, Meghir C (2012). "Risk pooling, risk preferences, and social networks." *American Economic Journal: Applied Economics*, **4**(2), 134–167. doi:10.1257/app.4.2.134.

Ballester C, Calvó-Armengol A, Zenou Y (2006). "Who's who in networks. Wanted: The key player." *Econometrica*, **74**(5), 1403–1417. doi:10.1111/j.1468-0262.2006.00709.x.

Batagelj V, Mrvar A (2003). "Pajek - Analysis and Visualization of Large Networks." In M Jönger, P Mutzel (eds.), *Graph Drawing Software*, chapter 3, pp. 77–103. Springer, Berlin.

Battaglini M, Leone Sciabolazza V, Patacchini E (2018). "Effectiveness of Connected Legislators." *NBER Working Paper*, (24442). doi:10.3386/w24442.

Battaglini M, Patacchini E (2018). "Influencing Connected Legislators." *Journal of Political Economy*. Forthcoming.

Bivand R, Altman M, Anselin L, Assunção R, Berke O, Bernat A, Blanchet G, Blankmeyer E, Carvalho M, Christensen B, Chun Y, Dormann C, Dray S, Gómez-Rubio V, Gubri M, Halbersma R, Krainski E, Legendre P, Lewin-Koh N, Li H, Ma J, Mallik A, Millo G, Mueller W, Ono H, Peres-Neto P, Piras G, Reder M, Tiefelsdorf M, Westerholt R, Yu D (2018). ***spdep**: Spatial Dependence: Weighting Schemes, Statistics and Models.* R package version 0.7.7, URL https://cran.r-project.org/web/packages/spdep/.

Bolker B (2017). ***bbmle**: Tools for General Maximum Likelihood Estimation.* R package version 1.0-20, URL https://cran.r-project.org/web/packages/bbmle/.

Bonacich P (1972). "Factoring and weighting approaches to status scores and clique identification." *Journal of mathematical sociology*, **2**(1), 113–120. doi:10.1080/0022250X.1972.9989806.

Bonacich P (1987). "Power and centrality: A family of measures." *American journal of sociology*, **92**(5), 1170–1182. doi:10.1086/228631.

Butts C (2016). ***sna**: Tools for Social Network Analysis.* R package version 2.4, URL https://cran.r-project.org/web/packages/sna/.

Csárdi G (2018). ***igraph**: Network Analysis and Visualization.* R package version 1.2-1, URL https://cran.r-project.org/web/packages/igraph/.

Elzhov TV, Mullen KM, Spiess AN, Bolker B (2016). ***minpack.lm**: R Interface to the Levenberg-Marquardt Nonlinear Least-Squares Algorithm Found in MINPACK, Plus Support for Bounds.* R package version 1.2-1, URL https://cran.r-project.org/web/packages/minpack.lm/.

Fafchamps M, Gubert F (2007). "The formation of risk sharing networks." *Journal of development Economics*, **82**(7), 326–350. doi:10.1016/j.jdeveco.2006.05.005.

Fafchamps M, Leij MJ, Goyal S (2010). "Matching and network effects." *Journal of the European Economic Association*, **8**(1), 203–231. doi:10.1111/j.1542-4774.2010.tb00500.x.

Goldsmith-Pinkham P, Imbens GW (2013). "Social networks and the identification of peer effects." *Journal of Business & Economic Statistics*, **31**(3), 253–264. doi:10.1080/07350015.2013.801251.

Graham BS (2015). "Methods of identification in social networks." *Annual Review of Economics*, **7**(1), 465–485. doi:10.1146/annurev-economics-080614-115611.

Graham BS (2016). "An econometric model of network formation with degree heterogeneity." *Econometrica*, **85**(4), 1033–1063. doi:10.3982/ECTA12679.

Handcock MS, Hunter DR, Butts CT, Goodreau SM, Krivitsky PN, Bender-deMoll S, Morris M (2016). ***statnet**: Software Tools for the Statistical Analysis of Network Data.* R package version 2016.9, URL https://cran.r-project.org/web/packages/statnet/.

Hsieh C, Fei LL (2016). "A social interactions model with endogenous friendship formation and selectivity." *Journal of Applied Econometrics*, **31**(2), 301–319. doi:10.1002/jae.2426.

Jackson MM (ed.) (2010). *Social and economic networks. Princeton.* Princeton university press, Princeton.

Jackson MO, Rogers BW, Zenou Y (2017). "The economic consequences of social-network structure." *Journal of Economic Literature*, **55**(1), 49–95. doi:10.1257/jel.20150694.

Katz L (1953). "A new status index derived from sociometric analysis." *Psychometrica*, **18**(1), 39–43. doi:10.1007/BF02289026.

Lai B, Reiter D (2017). "Democracy, political similarity, and international alliances, 1816-1992." *Journal of Conflict Resolution*, **44**(2), 203–227. doi:10.1177/0022002700044002003.

Leifeld P, Cranmer SJ, Desmarais BA (2018a). ***btergm**: Temporal Exponential Random Graph Models by Bootstrapped Pseudolikelihood.* R package version 1.9.1, URL https://cran.r-project.org/web/packages/btergm/.

Leifeld P, Cranmer SJ, Desmarais BA (2018b). ***xergm**: Extensions of Exponential Random Graph Models.* R package version 1.8.2, URL https://cran.r-project.org/web/packages/xergm/.

Manski CF (1993). "Identification of endogenous social effects: The reflection problem." *The review of economic studies*, **60**(3), 531–542. doi:10.2307/2298123.

Mayer A, Puller SL (2008). "The old boy (and girl) network: Social network formation on university campuses." *Journal of public economics*, **92**(1), 329–347. doi:10.1016/j.jpubeco.2007.09.001.

Millo G, Piras G (2017). ***splm**: Econometric Models for Spatial Panel Data.* R package version 1.4-10, URL https://cran.r-project.org/web/packages/splm/.

Opsahl T (2015). ***tnet**: Software for Analysis of Weighted, Two-Mode, and Longitudinal Networks.* R package version 3.0.14, URL https://cran.r-project.org/web/packages/tnet/.

Python Software Foundation (2018). *Python: Language Reference, version 2.7.15.* Python Software Foundation, Wilmington DE, USA. URL https://www.python.org/psf/.

R Core Team (2017). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Schweinberger M, Handcock MS, Luna P (2018). ***hergm**: Hierarchical Exponential-Family Random Graph Models.* R package version 3.2-1, URL https://cran.r-project.org/web/packages/hergm/.

StataCorp (2017). *Stata: Stata Statistical Software: Release 15.* StataCorp LLC, College Station, TX, USA. URL https://www.stata.com/.

The MathWorks Inc (2007). *Matlab The Language of Technical Computing, Version 7.5.* The MathWorks, Inc, Natick (MA), USA. URL https://www.mathworks.com/products/matlab/.

Wasserman S, Faust K (1994). *Social network analysis: Methods and applications.* 1st edition. Cambridge University Press, Cambridge.

Zenou Y (2016). "Key players." In Y Bramoullé, A Galeotti, B Rogers (eds.), *Oxford Handbook on the Economics of Networks*, chapter 10, pp. 244–274. Oxford University Press, Oxford.

Affiliation:

Marco Battaglini
Department of Economics
Cornell University
Ithaca, NY 14850, USA
and EIEF *and also* NBER
E-mail: battaglini@cornell.edu

Valerio Leone Sciabolazza
Department of Business and Economics
University of Naples Parthenope
Naples, 80132, Italy
E-mail: valerio.leonesciabolazza@parthenope.it

Eleonora Patacchini
Department of Economics
Cornell University
Ithaca, NY 14850, USA
and EIEF
E-mail: ep454@cornell.edu

Sida Peng
Office of Chief Economist
Microsoft Research
Redmond, WA 14865, USA
E-mail: sidpeng@microsoft.com