Contents

Introduction

When studying a system, it is often the case that an understanding of its behaviour cannot be obtained by analyzing only its individual elements in isolation, but it is also necessary to consider the interactions between its components. In this regard, the discipline of network science offers a framework for rigorous investigation of such relationships, by simplifying the system down to a tractable mathematical object, called *network*. Among the various areas of this field, one which has underwent remarkable advances has been community detection, which deals with the recovery of the community structure in a network. Managing to decompose a network into its modules can be of fundamental importance to grasp the functioning of the underlying system. This work focuses on stochastic block models, a family of generative models which lie at the heart of one of the possible approaches that have been developed to infer community structures. The purpose of the dissertation is to review and integrate part of the relevant literature and offer some original implementations. After a brief review of network science, which lays the foundations of the discussion in the following sections, the standard version of the stochastic block model is presented. Next, some of its many extensions that have been proposed, namely the degree-corrected, the oriented, and the regularized ones, are derived. Then, the inference algorithms have been implemented in R and stochastic block models are tested by conducting simulations and applications to real-world data. The final part of this work focuses on the issue of specifying the number of communities to infer, and, among the possible methods of model selection, a Bayesian approach is considered and applied.

1 Recap on Network Science

Before presenting the main topic of this work, it is necessary to recap on the basics concepts underlying the science of networks and set the notation and terminology.

1.1 Basics of Networks

Network science is a recent discipline whose purpose is to study the structure and evolution of networks as they are observed in real data. A network, or graph, is a collection of n nodes (or vertices) joined by m links (or edges). It is denoted by $(N, \mathbf{A})$, where N is the set of nodes $N = \{1, ..., n\}$ and $\mathbf{A}$ is a real-valued $n \times n$ matrix, the *adjacency matrix*, where A_{ij} represents the relation between nodes i and j [8].

An edge having a direction, pointing from one node to another, is called directed; otherwise, it is called undirected. A network is said to be directed (undirected) if all its edges are directed (undirected). A network without links from a node to itself (self-edges) and multiple edges between the same nodes (multiedges) is called a *simple graph*; otherwise, it is called a *multigraph*. A network whose edges are associated with a value are called *weighted* or *valued networks*.

For a graph $(N, \mathbf{A})$ that is undirected, simple and unweighted, for all $\{i, j\} \in N^2$,

$$A_{ij} = \begin{cases} 1 \text{ if there is an edge between } i \text{ and } j, \\ 0 \text{ otherwise.} \end{cases} \tag{1}$$

As the graph has no self-edges, $A_{ii} = 0$ for all $i \in N$. Having neither weights nor multiedges, $A_{ij} \in \{0, 1\}$ for $i \neq j$. Finally, being undirected, $\mathbf{A}$ is symmetric.

For a multigraph, A_{ij} is equal to the number of edges between nodes i and j if $i \neq j$, whereas A_{ii} is equal to twice the number of self-edges of i.

If all vertices represent the same type of object, the graph is called a *one-mode network*; conversely, *two-mode networks*, also called *bipartite*, display two kinds of nodes, and edges that join only nodes of different kinds.

A *walk* in a network is any sequence of nodes such that every consecutive pair of nodes in the sequence is connected by an edge. It is called *path* if it does not intersect itself. The *geodesic distance*, or just 'distance', between two nodes is the number of edges of the shortest path between them, d_{ij}.

A component of an undirected network is a maximal subset of nodes of the network such that there exists a path between each pair of nodes in the subset. A network in which all nodes belong to the same single component is said to be *connected*.

A network is said to be *complete* if each pair of distinct nodes is joined by an edge. A *clique* is a set of nodes within an undirected network such that each pair of distinct member nodes is joined by an edge.

The *degree* k_i of a node i is the number of links connected to it. In directed networks, we distinguish between the *incoming degree* k_i^{in} and the *outgoing degree* k_i^{out}, depending on the direction of the considered edges.

1.2 Summary Statistics of Networks

An important property of a network is its average degree

$$\langle k \rangle = \frac{1}{n} \sum_{i=1}^{n} k_i = \frac{2m}{n}. \tag{2}$$

The *density* ρ of a network is the fraction of the maximum possible number of edges that are present:

$$\rho = \frac{m}{\binom{n}{2}} = \frac{\langle k \rangle}{n-1}. \tag{3}$$

A network is said to be *sparse* if $\rho \to 0$ as $n \to \infty$, which then implies that $\langle k \rangle$ must grow sublinearly. Some networks even display constant average degree, and are also called *extremely sparse*.

To measure the centrality, or importance, of a node in a network, various metrics have been proposed (see [12]). Among them there are:

1. *Degree centrality*:
$$\frac{k_i}{n-1}; \tag{4}$$

2. *Closeness centrality*, which measures the mean distance of a node to other nodes:
$$Cl_i = \frac{n}{\sum_j d_{ij}}; \tag{5}$$

3. *Betweenness centrality*, which measures how well situated a node is in terms of the

paths that it lies on:

$$x_i = \frac{1}{n^2} \sum_{s,t \in N} \frac{n_{st}^i}{g_{st}}, \tag{6}$$

where g_{st} is the number of shortest paths from node s to node t, and n_{st}^i is the number of shortest paths from s to t that pass through node i;

4. *Katz centrality*, which takes the form, in matrix notation:

$$x = (\mathbf{I} - \alpha \mathbf{A})^{-1} 1; \tag{7}$$

it can be viewed as an extension of degree centrality where a node's importance is affected not only by the number of its neighbors, but also by their prominence; moreover, this definition forces all nodes to have non-zero centrality, thereby making the metric more robust.

The *diameter* $d_{\max}$ of a network is the maximum shortest path in the network. The average path length $\langle d \rangle$ is the average distance between all pairs of nodes in the network.

To measure the density of the set of neighbors of a given node, the *local clustering coefficient* is used:

$$C_i = \frac{2m_i}{k_i(k_i - 1)}, \tag{8}$$

where m_i is the number of links between the k_i neighbors of node i. Thus, the *average clustering coefficient* $\langle C \rangle$ can be used to measure how tightly clustered the whole network is:

$$\langle C \rangle = \frac{1}{n} \sum_{i=1}^{n} C_i. \tag{9}$$

For the same purpose, the *global clustering coefficient* C_Δ is also used:

$$C_\Delta = \frac{3 \times \text{number of triangles}}{\text{number of connected triples}}, \tag{10}$$

where a triangle is a completely connected triad of nodes, and a *connected triple* refers to an ordered set of three nodes uvw such that u connectes to v and v connects to w.

Graphs can also be classified according to their *degree correlations*, i.e. the relationship between the degree of nodes that link to each other. A network is called *assortative* if connections tend to form between nodes of comparable degree; for instance, social networks are typically of these kind, as individuals with many connections (hubs) tend

to have relationships with one another, while it is relatively infrequent for low-degree vertices to link to hubs. On the opposite, a network is *disassortative* if nodes whose degree is dissimilar tend to create ties; an example is the World Wide Web, whose nodes are web pages connected by the hyperlinks they contain, as a small portion of popular pages enjoy a huge number of connections coming from and pointing to other websites. If nodes link to each other randomly, the network is termed *neutral*. Degree correlations can be inspected through the *degree correlation function*:

$$k_{\mathrm{nn}}(k) = \sum_{k'} k' P(k'|k), \tag{11}$$

where $P(k'|k)$ is the conditional probability that by following a link of a degree-k node, a degree-k' node is reached. If $k_{\mathrm{nn}}(k)$ increases with k, the network is assortative; if it decreases with k, the network is disassortative. However, the concept of assortative and disassortative networks is not only confined to the aspect of degree correlations, but, more in general, can refer to the tendency of nodes to connect to similar, or dissimilar, vertices under any characteristic of interest.

1.3 Degree Distributions

The **degree distribution**, p_k, of a network provides the probability that a randomly selected node in the network has degree k. It can be seen as a frequency distribution when describing data, or, if dealing with theoretical models, as a probability distribution.

Poisson Degree Distribution. Consider a set of nodes $N = \{1, ..., n\}$. If each of the $N(N-1)/2$ node pairs is connected by an edge with probability p, and links are formed independently, the resulting network $G(n, p)$ is called an *Erdős–Rényi (ER)* random graph. It is easy to show that the degree distribution of the ER random network $G(n, p)$ is binomial:

$$p_k = \binom{n-1}{k} p^k (1-p)^{n-1-k}. \tag{12}$$

Since most real networks are sparse, which means that $\langle k \rangle << n$, the distribution (12) can be approximated by the Poisson distribution

$$p_k = e^{-\langle k \rangle} \frac{\langle k \rangle^k}{k!}, \tag{13}$$

which is the preferred form of p_k thank to its analytical simplicity.

A random network can exhibit topologically different regimes (see [5]) depending on the value of the average degree $\langle k \rangle = p(n-1)$. For $\langle k \rangle < 1$, the network has few links and the relative size of the largest cluster n_G/n tends to 0 as n increases. As $\langle k \rangle$ grows above the critical point 1, the largest cluster, now called *giant component*, contains a non-zero fraction of the nodes. For $\langle k \rangle > \ln n$, the network becomes connected.

Power Law Degree Distribution. A scale-free network is a network whose degree distribution follows a power law, that is

$$p_k \sim k^{-\gamma}. \tag{14}$$

The reason for the term *scale-free* is that, in the limit of $n \to \infty$, the standard deviation of degrees σ_k diverges, provided that γ is between 2 and 3, which is typically the case. Thus, this leads to arbitrarily large nodes, which means that the graph can be considered to be without any internal scale. Conversely, if k is Poisson-distributed as in (13), then $\sigma_k = \langle k \rangle^{1/2}$, and so $\langle k \rangle$ serves as a scale for the network [5].

Compared to ER random networks, scale-free distributions display fat tails, which means that the graph has a larger number of small- and high-degree nodes, whereas there is a lower probability to observe $k \approx \langle k \rangle$. The presence of hubs has also the effect of shrinking distances in the network. However, the topology of these graphs is sensitive to the value of the degree exponent γ, and becomes indistinguishable from the random model for $\gamma > 3$.

Random scale-free networks can be generated by algorithms such as the Barabási–Albert model, which employs merely network growth and preferential attachment in its basic formulation: the number of nodes of a network is increased, and each time a new node is added, it connects with m links to pre-existing vertices, whose probability of being selected increases with their degree. The model can be extended through additional processes which affect the resulting degree exponent.

2 Stochastic Block Models

Having summarized the fundamental concepts of network science, the idea of community structure can be introduced. A *community* is a group of nodes that are more likely to connect to each other than to nodes from other communities. The goal of community detection is to find the natural division of a network into groups so that the inner density of each group is high and density between communities is low.

2.1 Standard Stochastic Block Model

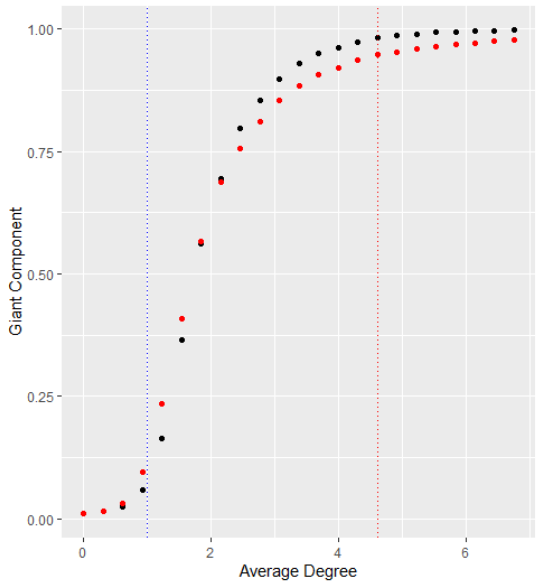
There are numerous possible approaches to community detection; here, the focus is on the one based on statistical inference, performed by fitting a model containing community structure to a network of interest. A canonical model employed for this task is the *stochastic block model (SBM).* In the basic SBM, n vertices are split among K communities, and, denoting by g_i the group to which vertex i belongs, undirected edges are formed independently between pairs of nodes i and j with probability $\psi_{g_i g_j}$, entry of a $K \times K$ matrix Ψ. The n-dimensional groups vector g can be fixed, or, alternatively, its elements can be distributed under a $K \times 1$ probability vector p on the communities. Thus, in this simple general case, we say that the pair (g, G) is drawn under SBM(n, p, Ψ), where g is the $n \times 1$ random vector containing the group assignment for each node, and G is a simple graph where vertices i and j are connected with probability $\psi_{g_i g_j}$ (see [4]). Given the parameters and a group assignment g, the probability of a graph G is:

$$P(G|g) = \prod_{r \leq s}^{K} \psi_{rs}^{e_{rs}} (1 - \psi_{rs})^{n_{rs} - e_{rs}}, \tag{15}$$

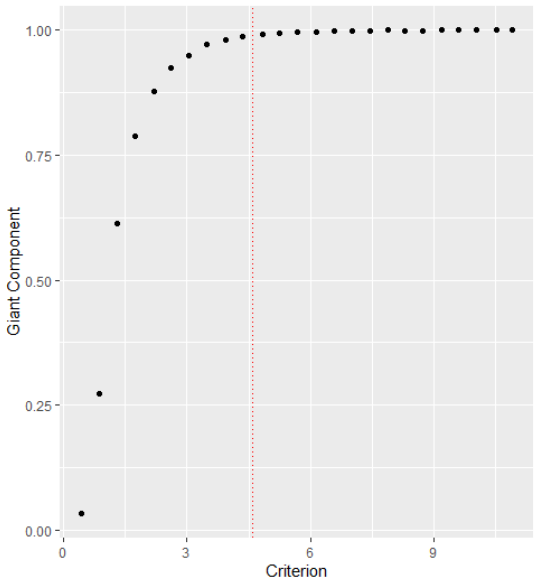
where e_{rs} is the number of edges between groups r and s, and n_{rs} is the maximum number of edges between the two groups, that is, the number of r, s pairs of nodes.

The SBM is called symmetric, written as SSBM$(n, K, \psi_{\text{in}}, \psi_{\text{out}})$, if Ψ takes the same value ψ_{in} for each entry on the main diagonal, and the same one ψ_{out} off diagonal, and all groups are assigned with equal probability to nodes.

If all entries of the matrix Ψ are equal to the same value ψ, the SBM collapses to the Erdős–Rényi model $G(n, \psi)$. Also, the SBM displays similar topological regimes to those discussed for the random graph in Subsection 1.3 (see [4]), where $\langle k \rangle = p(n-1)$. Indeed, in the case of a SSBM, the results of $\langle k \rangle > 1$ for the emergence of a giant component and of $\langle k \rangle > \ln n$ for connectivity hold with $\langle k \rangle = n \frac{\psi_{\text{in}} + \psi_{\text{out}}(k-1)}{k}$. In the case of the general



(a) ER vs. SSBM



(b) General SBM

Figure 1: Topological regimes of the stochastic block model. In (a) black dots are simulations of ER graphs, whereas red dots are from SSBM with $K = 2$. The blue line is at $\langle k \rangle = 1$, the red one is at $\langle k \rangle = \ln n$. In (b), a general SBM with $K = 2$ is simulated, with different values of (16). The red line is at $n \min_{r \in \{1,\dots,K\}} \|(\mathrm{diag}(p)\Psi)_i\|_1 = \ln n$.

SBM, we have that $\mathrm{SBM}(n, p, \Psi)$ is connected with high probability if

$$\min_{r \in \{1,\dots,K\}} n \|(\mathrm{diag}(p)\Psi)_i\|_1 > \ln n, \tag{16}$$

where $\mathrm{diag}(p)$ is the $k \times k$ diagonal matrix whose entries are the components of vector p.

In Figure 1, these thresholds are illustrated through the results of a simulation. In the first panel, the fraction of vertices belonging to the largest/giant component of the network is plotted against the average degree; for different values of $\langle k \rangle$, both an ER model and a SSBM are simulated 10000 times, and the average size of the giant component recorded. In the case of the SSBM, ψ_{in} and ψ_{out} are sampled from a uniform distribution, provided that the relation with the average degree is satisfied, and then setting $\psi_{\text{in}} > \psi_{\text{out}}$. In the second panel, the fraction of the largest component of networks generated from the general SBM is plotted against the left-hand side of (16); at each iteration, the ψ_{rs} are increased by a fixed amount, and (16) and component size are computed and averaged over 10000 simulations. All networks have $n = 100$, and the stochastic block models have two communities. The threshold below which a graph is not connected with high probability is shown in red.

It is convenient to allow the SBM to contain also multi-edges and self-edges. This is done by redefining ψ_{rs} to be the expected number of edges between nodes in groups r and s. Then, the number of links between a dyad (i, j) is drawn from a Poisson distribution with mean $\psi_{g_i g_j}$.

Consider a multigraph $G = (N, \mathbf{A})$, with n nodes. Let ω be a $K \times K$ matrix, with ω_{rs} being the expected value of the adjacency matrix element A_{ij} for each pair ij such that $i \in \{v : v \in N \cap g_v = r\}$ and $j \in \{v : v \in N \cap g_v = s\}$; this implies that $\psi_{g_i g_j} = \omega_{g_i g_j}$ if $i \neq j$, and $\psi_{g_i g_i} = \frac{1}{2}\omega_{g_i g_i}$. Thus, given ω and g, the probability of G is:

$$P(G|\omega, g) = \prod_{i<j} \left[\frac{\omega_{g_i g_j}^{A_{ij}}}{A_{ij}!} e^{-\omega_{g_i g_j}} \right] \prod_i \left[\frac{(\frac{1}{2}\omega_{g_i g_i})^{A_{ii}/2}}{(A_{ii}/2)!} e^{-\frac{1}{2}\omega_{g_i g_i}} \right]. \tag{17}$$

Although many real networks are simple, for a sparse and sufficiently large graph, this model is essentially equivalent to the previous one (15). Indeed, if the degrees of nodes grow sublinearly with n, the fraction of multi-edges and self-edges tends to 0.

In principle, we would fit this model to empirical data to perform *a posteriori* blockmodelling, that is, to detect block structure in the observed network.

Rewriting Eq. (17), taking the logarithm and neglecting constants, we get:

$$\ln P(G|\omega, g) = \sum_{rs} (m_{rs} \ln \omega_{rs} - n_r n_s \omega_{rs}), \tag{18}$$

where n_r is the number of vertices in group r and m_{rs} is the total number of edges between groups r and s if $r \neq s$, or twice that number if $r = s$.

Differentiating Eq. (18) with respect to ω_{rs}, we get $\hat{\omega}_{rs} = \frac{m_{rs}}{n_r n_s}$; substituting this into Eq. (18) and dropping constants, we find the profile log-likelihood for the group assignment g:

$$\mathcal{L}(G|g) = \sum_{rs} m_{rs} \ln \frac{m_{rs}}{n_r n_s}, \tag{19}$$

which is the objective function to maximize with respect to g to find the most likely partition into communities.

2.2 Degree-Corrected Stochastic Block Model

Despite its popularity, the simple stochastic block model suffers from severe limitations and often performs poorly when applied to real networks. This is due the fact that degrees of nodes in the same group have the same expected value and are Poisson-distributed. However, real networks exhibit degree heterogeneity, hence the SBM generates unrealistic networks and, when it is fitted to empirical data, it could end up partitioning graphs into groups of high- and low-degree nodes.

However, Karrer and Newman [9] have shown that through minor changes to the model in (17), a *degree-corrected stochastic block model (DC-SBM)* can incorporate degree heterogeneity and overcome these shortcomings.

This model differentiates itself from the standard one solely through a new set of parameters θ_i controlling the expected degrees of nodes, so that the expected value of A_{ij} is $\theta_i\theta_j\omega_{g_ig_j}$. Then we have

$$P(G|\omega,\theta,g) = \prod_{i<j}\left[\frac{(\theta_i\theta_j\omega_{g_ig_j})^{A_{ij}}}{A_{ij}!}e^{-\theta_i\theta_j\omega_{g_ig_j}}\right]\prod_i\left[\frac{(\frac{1}{2}\theta_i^2\omega_{g_ig_i})^{A_{ii}/2}}{(A_{ii}/2)!}e^{-\frac{1}{2}\theta_i^2\omega_{g_ig_i}}\right]. \tag{20}$$

However, the model is not completely specified yet, since the values θ_i can be arbitrarily increased, provided $\omega_{g_ig_j}$ is adequately decreased, and so must be normalized. To make θ_i equal the probability than an edge connected to g_i is connected to i itself, the following constraint is imposed:

$$\sum_i \theta_i\delta_{g_i,r} = 1, \tag{21}$$

where δ is the Kronecker delta function. We find that the maximum likelihood estimates of θ_i and ω_{rs} are

$$\hat{\theta}_i = \frac{k_i}{\kappa_{g_i}}, \quad \hat{\omega}_{rs} = m_{rs}, \tag{22}$$

where $\kappa_r = \sum_s m_{rs}$ is the sum of the degrees of the vertices in group r. This way, the expected degree sequence of the network is preserved, as $\langle k_i\rangle = k_i$, whereas in the standard SBM $\langle k_i\rangle = \kappa_{g_i}/n_{g_i}$, which is the same for all vertices in the same group. Making use of results (22), we get the profile likelihood:

$$\mathcal{L}(G|g) = \sum_{rs} m_{rs}\ln\frac{m_{rs}}{\kappa_r\kappa_s}. \tag{23}$$

Even though it is little different from (19), this model is capable of producing much more realistic networks, and has been shown to perform substantially better than the uncorrected counterpart. For this reason, it will be preferred to (19) in empirical analysis.

Network generation from this model can be easily implemented as shown in Algorithm 1: first, we draw m_{rs} from a Poisson distribution with mean ω_{rs}, or its half for $r = s$; then, the probability of an end of an edge connecting to a node i in r, given that it connects in r, is θ_i. This way, $A_{ij} \sim \text{Poisson}(\theta_i \theta_j \omega_{g_i g_j})$.

Algorithm 1: Generation from DC-SBM

Data: Set of vertices $N = \{1, ..., n\}$, Number of communities: K, Groups assignment vector: g, Expected degrees vector: d, $K \times K$ matrix Ω.
Result: Undirected Graph $G = (N, \mathbf{A})$
Initialize $K \times 1$ vector κ, $n \times 1$ vector θ, $K \times K$ matrix M.
$A \leftarrow \mathbf{0}_{K,K}$;
for $r = 1$ **to** K **do**
  $\kappa[r] \leftarrow \sum_{i=1}^{n} d[i]\delta_{g[i],r}$;
end
for $i = 1$ **to** n **do**
  $\theta[i] \leftarrow d[i]/\kappa[g[i]]$;
end
for $r = 1$ **to** K **do**
  for $s = r$ **to** K **do**
    if $s = r$ **then**
      $M[r, s] \sim \text{Poisson}(\Omega[r, s]/2)$
    else
      $M[r, s] \sim \text{Poisson}(\Omega[r, s])$
    end
    for $m = 1$ **to** $M[r, s]$ **do**
      Choose i among nodes, where nodes in group r are selected with probability $\theta[i]$ and other nodes with probability 0.
      Choose j among nodes, where nodes in group s are selected with probability $\theta[j]$ and other nodes with probability 0.
      if $i \neq j$ **then**
        $A[i, j] \leftarrow A[i, j] + 1$;
        $A[j, i] \leftarrow A[i, j]$
      else
        $A[i, j] \leftarrow A[i, j] + 2$
      end
    end
  end
end
return $(N, \mathbf{A})$

Community detection using the (DC-)SBM can be conducted in multiple ways. Newman

and Karrer ([9]) employed a node-moving algorithm (see Algorithm 2) which, starting from some partition into K groups, calculates the change $\Delta\mathcal{L}$ of (23) for each possible node-community move. The move leading to the largest $\Delta\mathcal{L}$ is performed. This process is repeated, excluding nodes that have already been moved, until all vertices have changed group. At this point, the partition of vertices in this process maximizing the profile likelihood (23) is used as starting point for another iteration of this kind. The algorithm stops when the likelihood no longer improves.

Algorithm 2: Node-Moving Algorithm for SBMs

Data: Undirected network $G = (N = \{1, ..., n\}, \mathbf{A})$, Number of communities: K, Initial parition: g_0.
Result: Groups assignment g
Stop $\leftarrow 0$;
$g \leftarrow g_0$;
$l_0 \leftarrow \mathcal{L}(G|g_0)$;
while $Stop = 0$ **do**
  $N' \leftarrow N$;
  for $t = 1$ **to** n **do**
    Computes the change in log-likelihood $\Delta\mathcal{L}$ for each possible move (i, s) of a node $i \in N'$ to a community $s \in \{1, ..., K\} \setminus \{g[i]\}$;
    Find move (i, s) with maximum $\Delta\mathcal{L}$;
    $g[i] \leftarrow s$;
    $N' \leftarrow N \setminus \{i\}$;
    Register the partition of the network g_t and its profile log-likelihood $\mathcal{L}(G|g_t)$
  end
  if $\max_t \mathcal{L}(G|g_t) > l_0$ **then**
    $g \leftarrow \arg\max_t \mathcal{L}(G|g_t)$;
    $l_0 \leftarrow \max_t \mathcal{L}(G|g_t)$
  else
    Stop $\leftarrow 1$
  end
end
return g

2.2.1 Extension to Digraphs

The degree-corrected SBM can be extended to directed graphs by replacing θ with two sets of parameters θ^{out} and θ^{in} (see [17]). Then, the number of directed edges from i to j is $A_{ij} \sim \text{Poisson}(\theta_i^{\text{out}}\theta_j^{\text{in}}\omega_{g_i,g_j})$, and the model becomes:

$$P(G|\omega, \theta^{\text{out}}, \theta^{\text{in}}, g) = \prod_{ij}\left[\frac{(\theta_i^{\text{out}}\theta_j^{\text{in}}\omega_{g_ig_j})^{A_{ij}}}{A_{ij}!}e^{-\theta_i^{\text{out}}\theta_j^{\text{in}}\omega_{g_ig_j}}\right], \tag{24}$$

where all couples ij are now considered, and no distinction is made between non-self-edges and self-edges, as the latter are represented by their actual number in the diagonal elements of the adjacency matrix of a directed network.

The parameters θ^{out} and θ^{in} are normalized through the constraints,

$$\sum_i \theta^{\text{out}} \delta_{g_i,r} = 1, \quad \sum_i \theta^{\text{in}} \delta_{g_i,r} = 1. \tag{25}$$

Denoting as κ_r^{out} the sum of the out-degrees in group r and as κ_r^{in} the sum of the in-degrees in group r, the maximum likelihood estimates for the paramters ω_{rs}, θ_i^{out} and θ_i^{in} are

$$\hat{\theta}_i^{\text{out}} = \frac{k_i^{\text{out}}}{\kappa_{g_i}^{\text{out}}}, \quad \hat{\theta}_i^{\text{in}} = \frac{k_i^{\text{in}}}{\kappa_{g_i}^{\text{in}}}, \quad \hat{\omega}_{rs} = m_{rs}, \tag{26}$$

where now m_{rs} is defined as the number of directed edges from group r to group s.

Then, new form of Eq. (23) is

$$\mathcal{L}(G|g) = \sum_{rs} m_{rs} \ln \frac{m_{rs}}{\kappa_r^{\text{out}} \kappa_s^{\text{in}}}. \tag{27}$$

By fixing the expected in- and out- degrees, the directed DC-SBM (DDC) is capable of generating and finding communities with heterogeneity in k_i^{in} and k_i^{out}. However, for the same reason, it cannot use the orientation of the edges when inferring the network partition, thereby performing poorly on directed networks with strongly asymmetric community structure. This can be fixed through an *oriented degree-corrected (ODC)* model (advanced by Zhu, Yan and Moore [17]), which generates an undirected graph according to the normal degree-corrected SBM, and then chooses the orientation of the edges through a specified $K \times K$ matrix ρ, such that $\rho_{rs} = 1 - \rho_{sr}$, by making an edge between nodes i and j point from i to j with probability ρ_{g_i,g_j}. Thus, the number of directed edges from i to j is distributed as $A_{ij} \sim \text{Poisson}(\theta_i \theta_j \omega'_{g_i,g_j})$, with $\omega'_{g_i,g_j} = \omega_{g_i,g_j} \rho_{g_i,g_j}$, which means that the ODC is the special case of the DDC where $\theta_i^{\text{in}} = \theta_i^{\text{out}} = \theta_i$. Therefore, its profile log-likelihood is

$$\mathcal{L}(G|g) = \sum_{rs} m_{rs} \ln \frac{m_{rs}}{\kappa_r \kappa_s}. \tag{28}$$

2.3 Blockmodelling and Assortativeness

A posteriori blockmodelling is a more general method with respect to traditional community detection, whose typical aim is finding a group assignment displaying a higher within-group density, i.e. an *assortative* community structure. While SBMs are capable of that, they can detect many other types of structure in a graph (see [9], [10]). This is evident, since the standard SBM assumes that the expected number of edges between two nodes depends only on their group memberships and matrix Ω; this means that nodes in the same block are statistically equivalent in their connectivity pattern. This *stochastic equivalence* does not require *assortativeness*, meaning ω_{rs} is low for $r \neq s$ and ω_{rr} is high.

SBMs can indeed generate graphs with many different types of partitions. The values of Ω control the group structure: if the elements on the main diagonal are higher than off-diagonal entries, the structure is assortative[1]; if the opposite is true, the model generates a disassortative community structure, potentially even bipartite; *core-periphery* and *onion structures* are produced, in the case of partitions which are neither assortative nor disassortative, having the within-densities range from very high to very low values, with intermediate between-densities (see [12]).

The agnosticism of the SBM and the DC-SBM to the assortativity of their solutions adds to their flexibility, but it could carry a risk if assortative structures are searched for, which is a frequent case: if non-assortative solutions give higher likelihood, assortative solutions of interest could remain unidentified, and the type of the discovered structure would depend on the starting point and on algorithm parameters (see [7]). Various approaches have been proposed to enforce assortative solutions. The *Regularised Stochastic Block Model (RSBM)*, proposed by Lu and Szymanski [11], is a generalization of the degree-corrected SBM, which allows to guide the inference algorithm towards a desired level of assortativeness. In the RSBM, each nodes i has two associated parameters I_i, O_i, and the expected number of edges between nodes i and j is $\omega_{g_i g_j} I_i I_j$ if $g_i = g_j$[2], and $\omega_{g_i g_j} O_i O_j$ if $g_i \neq g_j$. Substituting these values for the Poisson rates in Eq. (20) after having distinguished the two types of dyads, and deriving the profile log-likelihood, we get

$$\mathcal{L}(G|g, \{I_i\}, \{O_i\}) = \sum_{rs} m_{rs} \ln \frac{m_{rs}}{\Lambda_{rs}} + 2 \sum_i (k_i^+ \ln I_i + k_i^- \ln O_i) \tag{29}$$

[1] *Strong assortativity* when all diagonal terms are greater or equal than off-diagonal terms; *weak assortativity* when each diagonal term is greater or equal than the other terms in its row [7]

[2] Again, the half of this value for $i = j$.

where k_i^+ is the number of neighbors of i inside the community of i, $k_i^- = k - k_i^+$, and Λ_{rs} equals $\left(\sum_{i:g_i=r} I_i\right)^2$ if $r = s$, and $\sum_{i:g_i=r} O_i \sum_{i:g_i=s} O_i$ if $r \neq s$. This model reduces to the simple SBM if $I_i = O_i = 1$, and to the DC-SBM if $I_i = O_i = k_i$. Then, the prior in-degree ratios $f_i = I_i/(I_i + O_i)$ and $\theta_i = 1 - f_i$ are defined. Eq. (29) can be rewritten as

$$\mathcal{L}(G|g, \{I_i\}, \{O_i\}) = \sum_{rs} m_{rs} \ln \frac{m_{rs}}{\Lambda_{rs}} - 2\sum_i k_i H\left(\frac{k_i^+}{k_i}, f_i\right) + 2\sum_i k_i \ln \theta_i, \tag{30}$$

where $H\left(\frac{k_i^+}{k_i}, f_i\right) = -\frac{k_i^+}{k_i} \ln f_i - \frac{k_i^-}{k_i} \ln \theta_i$ is the cross-entropy between the observed in-degree ratio and the prior in-degree ratio. As the algorithm maximizes (30), it tends to reduce the cross-entropy terms, which penalize groups assignment with in-degree ratios distant from f_i. Choosing f_i to depend only on k_i, such as $f = \max\left(f, \frac{1}{k}\right)$, with $f \in (0, 1)$, the assortativity of the solution can be controlled with the single parameter f. Indeed, the more f is large, the more assortative community structures are detected.

2.4 Performance Evaluation

Any model for detecting community structure needs to be tested and its performance assessed and compared with that of its competitors. This can done through two different approaches: applying the methods on real-world networks whose partition is widely agreed upon (the *ground-truth division*), or testing them on synthetic networks. Although artificial networks are less realistic, their community structure is chosen by the researcher, and so it is certain and can be varied as needed. In this part, algorithms will be tested through the second approach, generating networks using Algorithm 1. Performance in finding communities can be evaluated through the *normalized mutual information (NMI)*, a measure for comparing the output of the algorithm and the known assignments (see [12]). This measure is based on the concept of *entropy* of a random variable X with probability mass function $P(x)$, defined as

$$H(X) = -\sum_{x \in \mathcal{X}} P(x) \ln P(x). \tag{31}$$

Similarly, the definition of conditional entropy of X given Y is

$$H(X|Y) = -\sum_{y \in \mathcal{Y}} P(y) \sum_{x \in \mathcal{X}} P(x|y) \ln P(x|y), \tag{32}$$

and it measures the amount of additional information contained in X, given that the value of Y is known. Let t be the vector of ground-truth assignments. Then, the conditional entropy of $t|g$ conveys how much more we learn about the network partition from the true assignments beyond what we got from our inference methods. The value of $H(t|g)$ ranges from 0, if the algorithm perfectly recovers the community structure, to $H(t)$, if the recovered partition is totally unrelated to the true one. One advantage of this approach is that its results do not depend on the labels on the two assignment vectors. Then, we define the *mutual information* as $I(t;g) = H(t) - H(t|g)$, which takes 0 in case of algorithm failure, and $H(t)$ with perfect performance. Finally, the NMI just normalizes this measure to the $[0, 1]$ range. It is usually defined as

$$N(t;g) = \frac{2I(t;g)}{H(t) + H(g)}. \tag{33}$$

In order to test the performance of the standard SBM and compare it with that of its degree-corrected counterpart, the following approach has been adopted, influenced by Karrer and Newman [9]. First, networks of 50 nodes have been generated from the DC-SBM as in Algorithm 1; the nodes have been partitioned in two communities at random and the parameters ω_{rs} have been set as

$$\omega_{rs} = \lambda\omega_{rs}^{\text{planted}} + (1-\lambda)\omega_{rs}^{\text{random}}, \tag{34}$$

where

$$\Omega_{\text{planted}} = \begin{bmatrix} \kappa_1 & 0 \\ 0 & \kappa_2 \end{bmatrix} \tag{35}$$

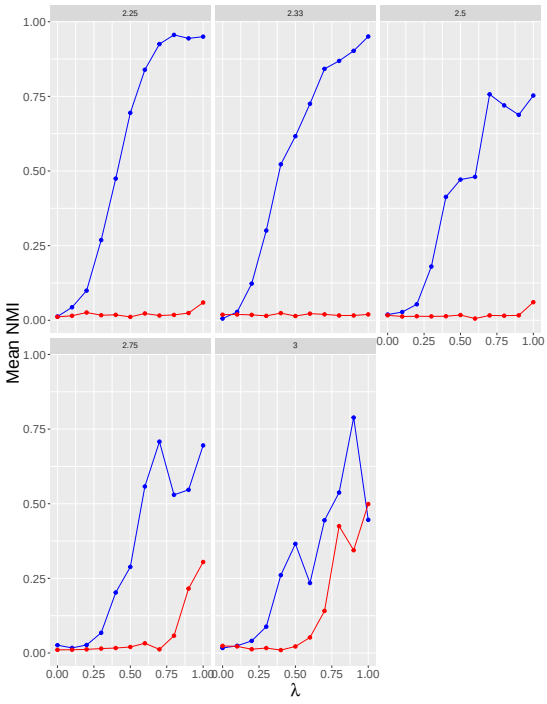
and $\omega_{rs}^{\text{random}} = \frac{\kappa_r\kappa_s}{2m}$ is the expected value of ω_{rs} in a random graph with fixed expected degrees. Thus, for $\lambda = 1$, the only edges will be placed between nodes of the same communities, while, for $\lambda = 0$, no community structure will be detectable. The expected degrees θ are selected by generating scale-free networks of 50 vertices and using their degree sequence. These networks are produced using the Barabási–Albert model, discussed in Subsection 1.3. The BA model in its basic form generates networks with $\gamma = 3$; however, in this case it has been extended with the presence of internal links: after each new node with m links is added, n new internal edges are placed between existing nodes, with pairs of high-degree vertices being more likely to be linked. This changes the form of the degree

exponent to

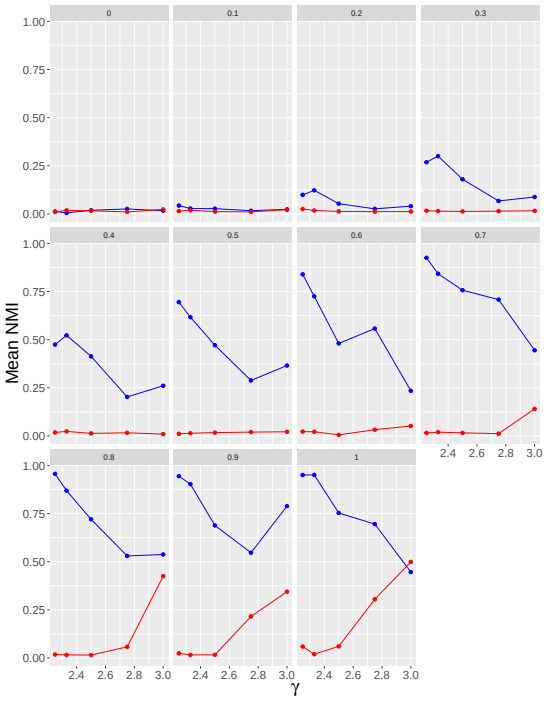
$$\gamma = 2 + \frac{m}{m + 2n}, \tag{36}$$

meaning $\gamma \in (2, 3]$[5]. The values of λ in the set $\{0, 0.2, 0.4, 0.6, 0.8, 1\}$ and the values of γ in $\{3, 2.75, 2.5, 2.33, 2.25\}$ have been considered, where the value of the degree exponent has been varied using the parameter n, keeping m fixed to 6. Then, for each of the λ, γ pairs, 40 networks have been generated. The node-moving algorithm in Algorithm 2 has been used on each of them, but on half of the graphs the traditional SBM has been applied, maximizing Eq. (19), while the DC-SBM has been implemented on the other half (23). Finally, the NMI (33) has been computed. Results are shown in Figure 2.

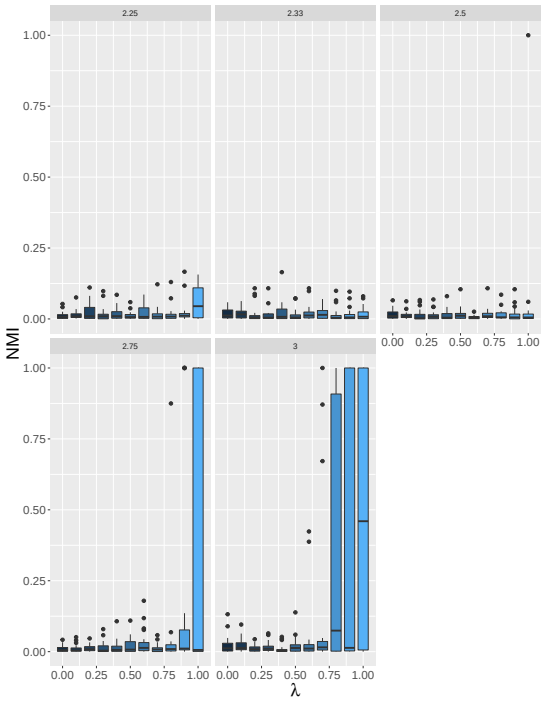
The DCSBM almost always outperformed the standard model, often substantially so. As expected, both algorithms get better at recovering the latent structure as λ increases, as edges are less and less influenced by randomness, but the uncorrected model is often incapable of decent results for a wide range of λ values (see panel a). The difference in performance seems to decrease for higher γ, a phenomenon which is clarified in panel b. Indeed, while for very small λ values the influence of the degree exponent on performance is negligible, community detection quickly becomes dependent on γ. For at least $\lambda > 0.2$, the DCSBM depends negatively on the degree exponent; in constrast, starting at least from $\lambda = 0.7$, the standard stochastic block model benefits from an increased γ. This last result was particularly expected: a scale-free network with $\gamma > 3$ is very similar to a random graph, exhibiting low degree heterogeneity. At the opposite, the more γ is near 2, the less homogeneous is a network. This data reinforce the statement that the basic SBM gives poor results when applied to non-homogeneous networks, which is usually the type of structure observed in real-world data. Then, we observe that, for high values of γ, the performance of the two models becomes more similar as λ approaches 1.



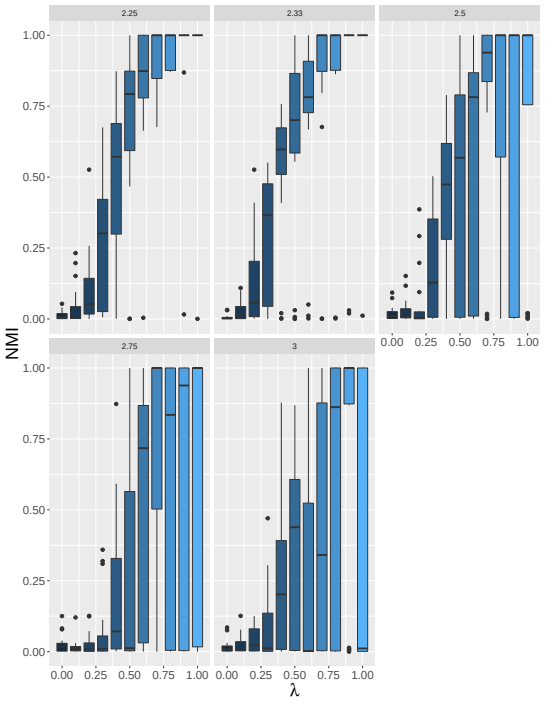
(a) SBM vs. DCSBM by γ



(b) SBM vs. DCSBM by λ



(c) SBM: Conditional Distribution of NMI



(d) DCSBM: Conditional Distribution of NMI

Figure 2: Comparison between the SBM and the DCSBM. In (a) the average NMI for DCSBM (blue) and for SBM (red) is plotted against λ, for each value of γ; γ is 2.25 at the top-left corner and increases up to 3 in the bottom-middle panel. In (b), γ is on the x-axis and λ is fixed for each graph. While (a) only shows the mean values, panels (c) and (d) show the entire conditional distribution of NMI obtained by the SBM and the DCSBM respectively.

2.5 Empirical Analysis: FAO Trade Network

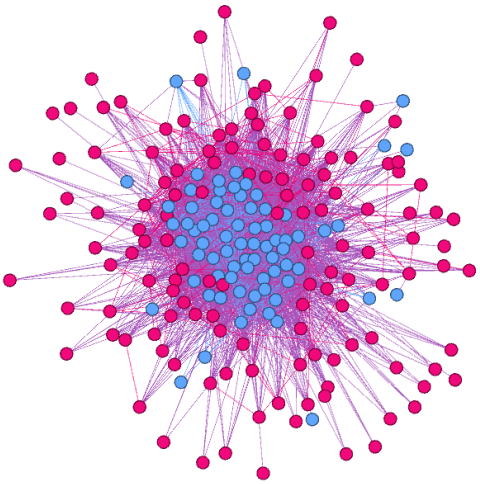
The section ends with an application of the SBMs to real network data. The data comes from the FAO multiplex trade network ([6], [3]), representing the worldwide food import/export relationships in 2010. Each layer in this dataset represents a network for a specific product, nodes are countries, and edges are directed, going from an exporter to the corresponding importer. Also, edges are weighted, with the weights being the value of the corresponding exchange in thousands of US dollars. In this brief empirical analysis, the network of international trade in crude materials, counting 207 connected nodes, is selected.

Various different approaches have been considered to infer some community structure in the graph and to observe the behaviour of the models. The first method consisted in making the graph undirected, and losing all weights. This returns a simple network, linking two countries if and only if they merely trade with each other the product of interest. Clearly, however, this transformation wastes lots of potentially useful information. The algorithm based on the DC-SBM has been applied, after having specified $K = 2$; indeed, given the nature of the data and the order of the networks, parsimony in selecting the number of groups has been favoured.

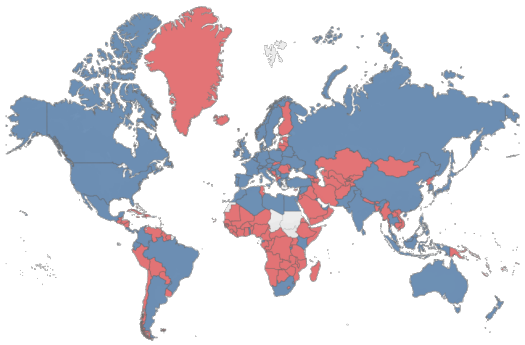
As shown in Figure 3, the algorithm appears to have returned some kind of core-periphery structure, with one community being constituted mostly by richer and more connected countries, which enjoy lots of links among themselves and numerous ties even with less developed nations, and the other being composed in large part by countries with fewer trading partners, and with a low tendency to link to each other. The core-periphery structure, implying $\omega_{rr} > \omega_{rs} > \omega_{ss}$, is indeed evident in the adjacency matrix of the graph, plotted in panel (c).

The second approach saw the graph preserving its directed and valued nature. Weighted edges have been converted to multiedges; however, before the conversion to a multigraph, weights have been divided by 500 and rounded up to the nearest integer. Evidently, also this decision, adopted to simplify the network, entails a loss of information and a risk for our estimation. Finally, the DCSBM generalised to digraphs (DDC) has been implemented. In the top panels of Figure 4, the detected groups under $K = 2$ are shown.

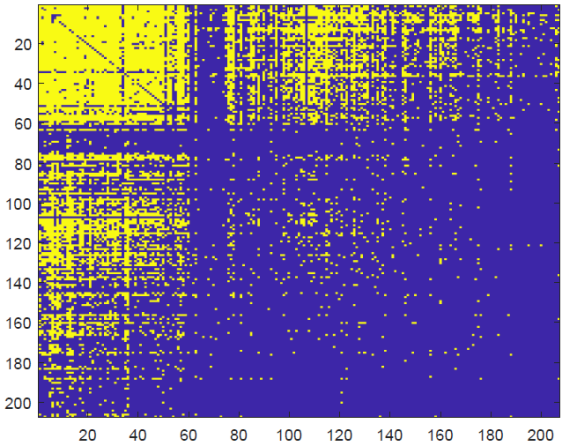
This time, the algorithm found a more assortative community structure. By looking at the map in panel (b), it also seems like the partition closely fits several macroregions



(a) Crude Materials, Network A



(b) Partition on the Map, Network A

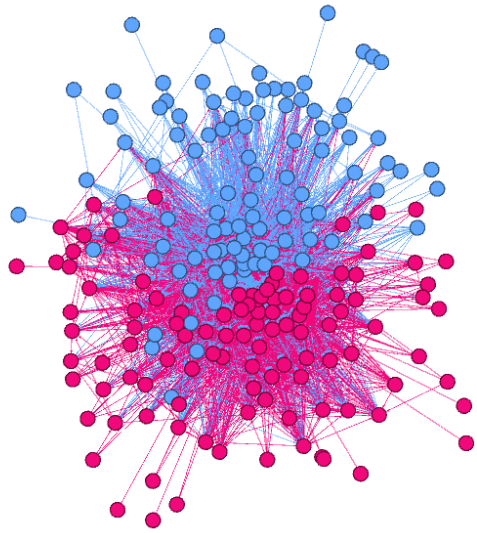


(c) Adjacency Matrix, Network A

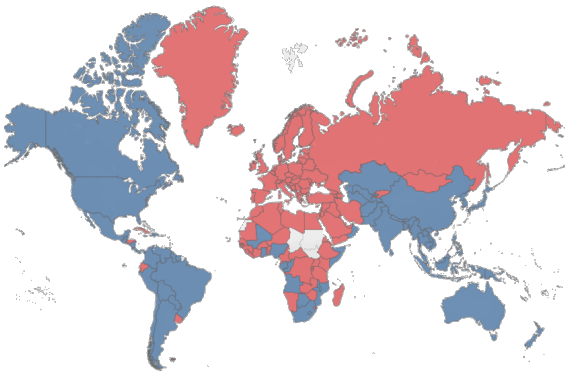
Figure 3: Panel (a) shows the crude materials network after being simplified, with nodes coloured according to the found partition. The same partition splits the countries on the world map in (b). In (c), the adjacency matrix, after nodes have been reordered according to their group to display the density within and between communities.

and continents: the entirety of Europe and the Middle East on one side, (the rest of) Asia, the Americas and Oceania on the other, with Africa split between the two. In the bottom panels of Figure 4, the same method has been applied, but increasing the number of groups to 3. One of the two communities found with $K = 2$ is identically returned under this specification, while the other one has been split in two parts. The resulting assignment is substantially less intuitive than the previous one, and it is difficult to give any interpretation of it.

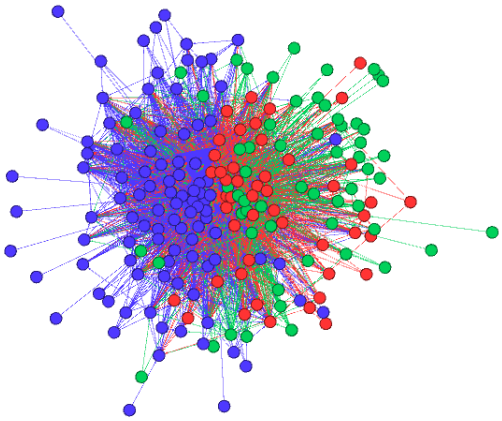
Until now, the orientation of the trade flows has not been used to determine the partition,



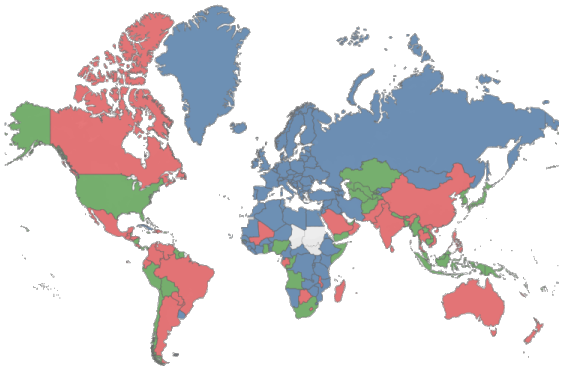
(a) Crude Materials, Network B



(b) Partition on the Map, Network B



(c) Crude Materials, Network C

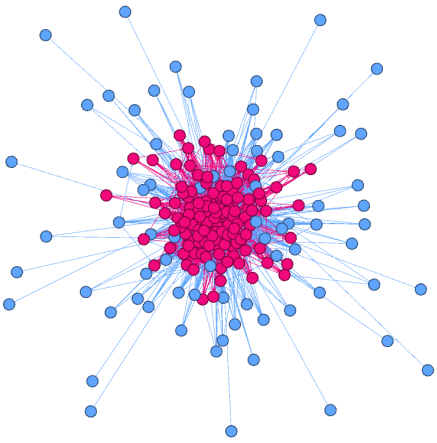


(d) Partition on the Map, Network C

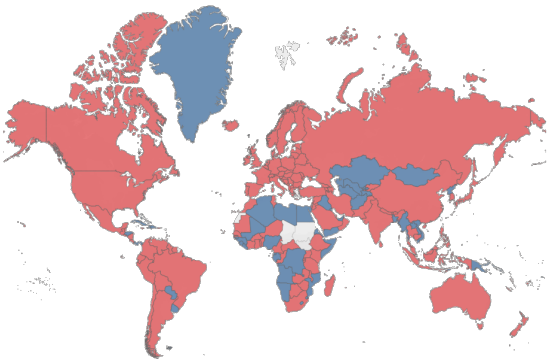
Figure 4: Panel (a) shows the crude materials network after the weights have been simplified and converted to multiedges, with nodes coloured according to the found partition. The same partition splits the countries on the world map in (b). In the bottom panels, the same graph has been analyzed, setting instead $K = 3$.

even though one could expect a trade network to have clusters of nodes engaging in asymmetric relationships. Figure 5 shows the partition obtained by applying the oriented degree-corrected SBM to the directed network free of weights and multiedges.

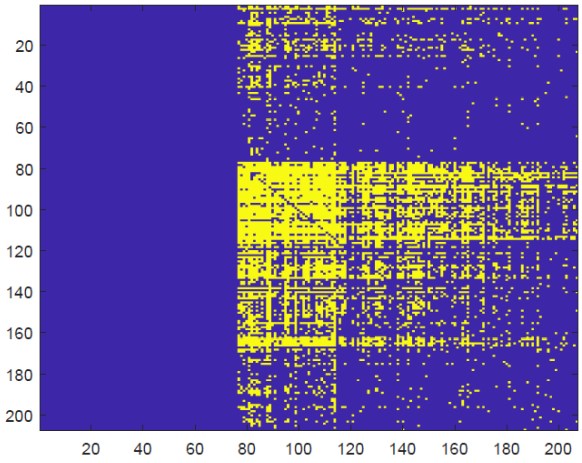
While the DDC had divided the graph in two almost-equal-size communities, here the ODC returns a partition where one group is twice as big as the other. Although this division is more similar to that in 3, the difference with the undirected case is made clear by the two adjacency matrices: first, in the initial approach, the group of less connected



(a) Crude Materials, Network D



(b) Partition on the Map, Network D



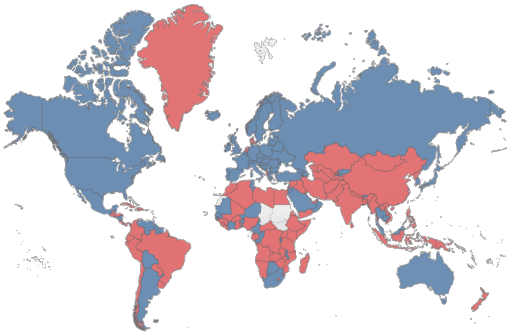
(c) Adjacency Matrix, Network D

Figure 5: The panels show the partition for the directed crude materials trade network without multiedges, as returned by the ODC model.

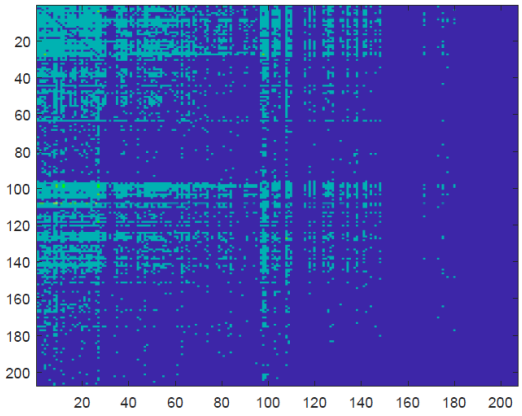
nodes was far larger, indeed the largest community of that network, whereas here the dense cluster contains most of the vertices; however, the main feature in panel (c) of Figure 5 is the complete absence of directed edges pointing to nodes in the smallest community. This extremely asymmetric, one-way community structure clearly was not in reach of the DDC employed in Network B (4). More specifically, what the ODC model did was to separate nodes having in-degree zero and not zero. Indeed, the blue-colored community perfectly coincide with the set of countries which do not import crude materials from anywhere in the world (and clearly have exports, otherwise they would have been excluded in this analysis).

Finally, the oriented model is implemented again, but this time on the multigraph generated

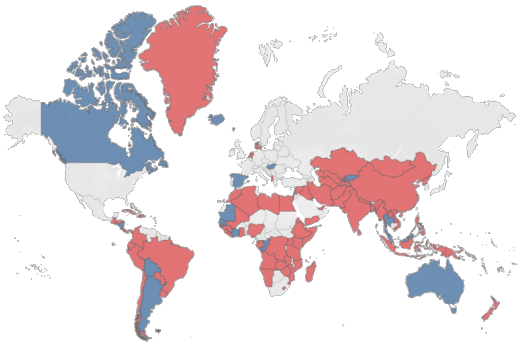
through the same weights transformation used under the DDC. The top panels of Figure 6 display the resulting assignment and structure.



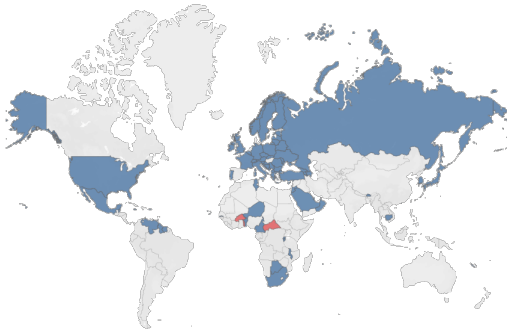
(a) Partition on the Map, Network E



(b) Adjacency Matrix, Network E



(c) Partition on the Net Exporters



(d) Partition on the Net Importers

Figure 6: The panels show the partition for the directed crude materials trade network with reduced multiedges, as returned by the ODC model. Panel (c) is a view of (a) restricted to countries whose crude materials imports are below their exports, and (d) displays the rest of the world.

By keeping more information with the application of some thresholds on the multiedges, the ODC-recovered communities change, although they maintain an asymmetric relationship, albeit less blatant, as shown in panel (6b). In the bottom panels of Figure 6, the same partition of panel (a) is reported, but only for countries which are net exporters of crude materials, in panel (c), and only for those that are net importers, in panel (d). Thus, by making use of links orientations, despite the severe loss of information due to the thresholds, the algorithm inferred a partition which is highly correlated with the countries' net exports for this specific global market. Indeed, out of the 78 net importers, 75 are put together in the same community, and out of the 129 net exporters, 109 are pooled in the other group.

3 Model Selection of Stochastic Block Models

A disadvantage which all the previously presented inferential techniques share is that they require the setting of the number of communities K. However, typically the researcher does not know *a priori* how many groups there are in the network, and so K must be estimated first. Among the possible approaches, the one presented here is based on Bayesian inference.

3.1 Bayesian Inference

While in frequentist inference parameters are considered fixed and unknown values, in Bayesian statistics, interpreting probability as a measure of degree of belief, parameters are treated as random variables. Thus, in Bayesian inference, the posterior distribution of some parameter of interest is obtained by updating the prior beliefs on the parameter with observed data. This is done through the Bayes' rule:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}, \tag{37}$$

where $P(\theta|D)$ is the posterior distribution of parameter θ, $P(D|\theta)$ is the likelihood function, expressing the probability of the observed data D given the parameter, $P(\theta)$ is the prior probability distribution of θ, and $P(D)$, the *evidence*, is a normalizing constant.

Applying Eq. (37) to the task of estimating K and g for a network G, we have

$$P(g, K|G) = \frac{P(G|g, K)P(g, K)}{P(G)}. \tag{38}$$

The choice of the prior $P(g, K)$ affects substantially the posterior distribution. In other circumstances, the prior distribution would be chosen based on previous data coming from the same model. This is not possible in this context, however, since networks usually do not come from a population, but are instead unique objects (see [13]). In the complete absence of prior information, a reasonable $P(g, K)$ would be one with minimal influence on the inference, i.e., an *uninformative prior* [16]. A useful method for choosing a prior which is noninformative given some available partial information is based on the *principle of maximum entropy* (see [15]), for which the selected prior distribution $P(x)$ is the one that maximizes the Shannon entropy (see Eq. 31), subject to any constraint based on some knowledge of the distribution.

3.2 Bayesian Model Selection

The estimation of the number of groups K in a graph can be achieved through the Bayesian model selection procedure proposed by Riolo, Cantwell, Reinert and Newman [14], employing the DC-SBM. The probability of a network G given the values of the parameters, $P(G|\omega, \theta, g, K)$, is given by Eq. (20). The likelihood $P(G|g, K)$ is obtained by integrating out the parameters θ and ω:

$$P(G|g,K) = \int\int P(G|\omega,\theta,g,K)P(\theta|g,K)P(\omega|K)d\theta d\omega, \tag{39}$$

where the priors for θ and ω have been assumed to be independent. The chosen prior on θ is the maximum entropy prior with no constraint, i.e. a uniform distribution over the values defined by Eq. (21). The chosen prior on ω_{rs} is the maximum entropy prior with constraint $E(\omega_{rs}) = p$, where p is the average probability of an edge in the network, which is the exponential distribution $P(\omega_{rs}) = (1/p)e^{-\omega_{rs}/p}$; the uniform distribution over all possible values would contrast with the fact that most networks are very sparse, with low probability of an edge between a dyad. Thus, by Eq. (39), the derived likelihood is:

$$P(G|g,K) = \prod_r \left[n_r^{\kappa_r} \frac{(n_r-1)!}{(n_r+\kappa_r-1)!} \right] \prod_{r<s} \left[\frac{m_{rs}!}{(pn_rn_s+1)^{m_{rs}+1}} \right] \prod_r \left[\frac{m_{rr}!}{(\frac{1}{2}pn_r^2+1)^{m_{rr}+1}} \right] \tag{40}$$

To perform inference using the Bayes' rule (38), the prior distribution $P(g, K)$ must be specified. The number of possible ordered partitions in a network of order n is given by the ordered Bell number

$$a_n = \sum_{K=1}^{n} \left\{ {n \atop K} \right\} K!, \tag{41}$$

where $\left\{ {n \atop K} \right\}$ are the Stirling numbers of the second kind, counting the number of possible partitions of n nodes into K nonempty unlabelled groups. $\left\{ {n \atop K} \right\} K!$ recovers instead the number of partitions into K (nonempty) labelled groups (see [13]). Hence, the noninformative prior on the group assignments is $P(g, K) = 1/a_n$. However, this flat prior is problematic, as it favors values of K comparable to the number of nodes; indeed, $\left\{ {n \atop K} \right\} K!$ is greater than $\left\{ {n \atop K-1} \right\} (K-1)!$ for K sufficiently smaller than n. Thus, this choice is not uninformative with respect to the number of communities. Similarly, the choice of recovering $P(g, K)$ by setting the uniform prior on K, $P(K) = 1/n$, and on $g|K$, $P(g|K) = \frac{1}{\left\{ {n \atop K} \right\} K!}$, would also entail unreasonable assumptions, as partitions sampled

from this $P(g|K)$ would exhibit similar group sizes. Indeed, one usually assumes a uniform distribution over all sets of community sizes n_r, provided they sum up to n and there is no empty group; the number of these choices is $\binom{n-1}{K-1}$. Then, the assignment of nodes over these groups is also sampled from a uniform distribution, where the number of possible choices is given by the multinomial coefficient $n!/\prod_r n_r!$. This leads to

$$P(g|K) = \frac{\prod_r n_r!}{n!}\binom{n-1}{K-1}^{-1}. \tag{42}$$

Now, selecting $P(K)$, there are arguments for choosing a strongly decreasing prior on K such as $1/K!$. Instead of directly choosing the density, Riolo et al. in [14] employ a process which, starting with node 1 in group 1, considers each node in turn and either puts it in the same group as the previous node, with probability $1-q$, or puts it in a newly created group, with probability q. This probability is then parametrized for convenience by $q = \mu/(n-1)$, where μ is the expected number of *new* groups created in the process and it is set to 1 as it showed to provide good results. It can be derived that in this case the probability of any group assignment is

$$P(g, K) = (n-2)^{-K}\prod_{r=1}^{K} n_r!, \tag{43}$$

which allows to recover the prior on K, substantially equivalent to $1/K!$.

The posterior distribution (38) could now be computed; however, this is not feasible, since, even though $P(G)$ is just a normalizing constant, it is usually a sum over a too large number of elements. A way to simulate from the posterior distribution would be using Markov chain Monte Carlo (MCMC) sampling. The MCMC scheme is the following: first, an *irreducible* Markov chain with stationary distribution equal to the desired one $P(g, k|G)$ is designed; then, after initializing the first state of the stochastic process, a trajectory of the Markov chain is simulated, and the states of this trajectory are approximately distributed as the distribution of interest.

Riolo et al. (2017) [14] propose a MCMC algorithm, whose steps have the following structure:

1. With probability $1 - 1/(n-1)$,
   (a) if $K = 1$ nothing happens, otherwise a pair of distinct groups r, s is chosen

uniformly at random from all such pairs, and a node chosen at random from group r is moved to group s;

(b) if a group r becomes empty, the label of group K is set to r.

With probability $1/(n-1)$,

(a) a pair of distinct groups labels r, s belonging to $\{1, ..., K+1\}$ is chosen uniformly at random from all such pairs; group r is relabeled as $K+1$; a new group is created with label r, and a node chosen at random from group s is moved to group r;

(b) if a group s becomes empty, the label of group $K+1$ is set to s, otherwise K is increased by 1.

2. The move $(g, K) \to (g', K')$ proposed above is accepted with probability

$$\alpha(g, K \to g', K') = \min\left(1, \frac{P(G|g', K')}{P(G|g, K)}\right). \tag{44}$$

If this process is indeed irreducible and having stationary distribution $P(g, k|G)$, the algorithm would work. Irreducibility is trivially satisfied, since every state g, k of the process is indeed accessible from every other. A sufficient condition for a Markov chain to have stationary distribution π is being π-reversible, which means $\pi(i)P_{ij} = \pi(j)P_{ji}$ for all states i, j, where $\{P_{ij}\}$ are the transition probabilities of the chain. In this case, this means

$$P(g, k|G)P(g, k \to g', k') = P(g', k'|G)P(g', k' \to g, k), \tag{45}$$

which can be rewritten as

$$\frac{P(g, k \to g', k')}{P(g', k' \to g, k)} = \frac{P(g', k'|G)}{P(g, k|G)} = \frac{P(g', k')}{P(g, k)} \frac{P(G|g', k')}{P(G|g, k)}. \tag{46}$$

For convenience, $P(g, k \to g', k')$ can be split into the probability π of proposing the relevant move (Step 1) and the probability α of accepting it (Step 2), so that

$$\frac{P(g, k \to g', k')}{P(g', k' \to g, k)} = \frac{\pi(g, k \to g', k')}{\pi(g', k' \to g, k)} \frac{\alpha(g, k \to g', k')}{\alpha(g', k' \to g, k)}. \tag{47}$$

Now, it is possible to show that for each type of move $g, k \to g', k'$,

$$\frac{\pi(g, k \to g', k')}{\pi(g', k' \to g, k)} = \frac{P(g', k')}{P(g, k)}. \tag{48}$$

At this point, comparing Eq. (46) and Eq. (47), the condition for π-reversibility of the Markov chain becomes

$$\frac{\alpha(g, k \to g', k')}{\alpha(g', k' \to g, k)} = \frac{P(G|g', k')}{P(G|g, k)}, \tag{49}$$

which is indeed satisfied by Eq. (44). Thus, the posterior $P(g, k|G)$ is the stationary distribution of the stochastic process, and the MCMC scheme will sample from this distribution.

3.3 Empirical Analysis: Corporate Network

Inferential methods which estimate also the number of groups are fundamental in circumstances in which no ground truth is available. To show an example application of the previously described algorithm, a network has been extracted from corporate data of 89706 Italian firms, stored in the AIDA database [1]. In this undirected graph, two firms are connected by a number of edges equal to the number of directors and senior managers they have in common; the subgraph induced by the set of nodes belonging to the giant component is constituted by 34433 businesses with 460591 ties. Instead of carrying out inference on an object with such dimensions, a smaller subset of the firms' network has been considered. One could hope that, through blockmodelling, proximity and power relations between companies, as well as potential undue positions of dominance in a sector or market, would emerge; therefore, the MCMC algorithm which has been discussed previously could prove useful in applications to this type of data, where it is unclear how many communities should be searched for.

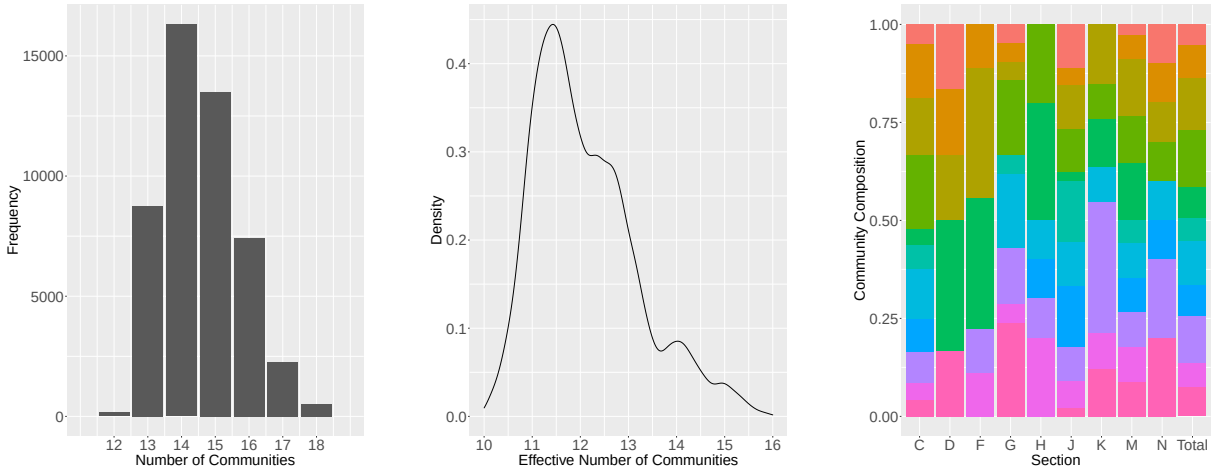
Here, the method is tested using the implementation in C provided by Riolo, Cantwell, Reinert and Newman at [2]. The program cannot return a number of communities larger than the one specified at the beginning, hence the choice of K is highly dependent on the context. Apart from reporting the number of communities and the log-likelihood at each step, this implementation also tracks the *effective number of groups* K_{eff}, defined in [14] as

$$K_{\text{eff}} = e^{H(g_i)}, \tag{50}$$

where $H(g_i)$ is the entropy of the group assignment $-\sum_{r=1}^{k} \frac{n_r}{n} \ln \frac{n_r}{n}$. This measure is equal to k if groups have equal size; otherwise, it will approximately measure only the number of large groups, substantially ignoring communities with very few nodes.

In this example, the graph has been restricted to (the largest component of) not-failed

publicly listed companies, amounting to 315 nodes and 1370 edges. The algorithm has been applied to this network setting the initial number of communities at $K = 25$, and the number of Monte Carlo steps at 50000. By plotting the log-likelihood through the process, the point at which its value becomes stable can be found, thereby discarding all output from previous steps. In this case, values of K for the first 1000 iterations have been ignored. Results are shown in Figure 7.



(a) Distribution of K

(b) Distribution of K_{eff}

(c) Community Shares by ATECO Section

Figure 7: Panel (a) shows the absolute frequencies of the values of K, and (b) plots the distribution of K_{eff}. In (c), the partition into communities is displayed, conditioned on the ATECO section of firms.

The panel on the left shows the posterior distribution for K, with mean 14, while K_{eff} has mean between 11 and 12. A potential reason to use community detection methods on this kind of networks is to observe whether relationships between firms, framed here exclusively as overlap in their top management, can be used to identify positions of control over a sector of the economy by a small cluster of actors. Panel (c) displays the share of vertices in each of the 11 non-negligible communities found by the algorithms, where the last bar includes all nodes, and the other ones comprise firms under a certain ATECO category. ATECO classes define the kind of economic activity performed by the business through multiple degrees of precision; in 7c, nodes are divided using the most general level of the classification, the section, and discarding the very under-represented classes. Making use of a richer network, if most of the companies belonging to some economic sector were part of one or few communities, then one could suppose the presence of a risk for consumers, and use this information for further analysis and to shape policy action. In this test, community composition, albeit varying across ATECO sections and with

respect to the entire network, does not seem to be particularly concentrated in a specific business area; however, this example makes clear how the possibilities to take advantage of blockmodelling and community detection algorithms in policymaking are virtually endless, as these methods offer a way to inspect the particularly complex structure of social and economic networks.

Conclusions

In this work, the stochastic block model and some of its extensions have been discussed and implemented. Although only a few of the possible variants of the model have been reviewed here, it has been shown how versatile and adaptive this family of methods can be, while preserving its mathematical tractability, and in the context of a clear and rigorous framework for detecting community structures. Just by what has been considered in this dissertation, the SBM can be adapted to networks with any degree sequence, to any desired level of assortativity in the recovered communities, and to directed graphs with asymmetric structures, but these are only some of the possible contexts of use. Also, it is capable of generating and finding any type of community patterns, by assuming only stochastic equivalence, and there are methods to extend the inference also to the number of groups. Just as network science allows or deepens the understanding of a system whose objects engage in intricate relations and could not be profitably studied in isolation, so community detection is necessary to derive the latent structure of a graph which otherwise would be handled under an unrealistic, shallower approach; with this in mind, stochastic block models represent a valuable and effective tool to investigate the functioning of systems in a vast set of circumstances.

A Code

All code used in this dissertation is written in R language. Networks fed to and returned by the functions below are objects from the `igraph` package.

Code - Generation from DC-SBM

The code below is the R implementation of the DC-SBM generation algorithm. The function produces an undirected graph, but it can be trivially transformed into its directed counterpart. It allows to set a parameter λ to interpolate between a random graph and any specified community structure. If no community structure is specified, an assortative one like in (35) is chosen. The elements of the group vector are expected to range from 1 to the number of communities.

```
DCSBM_und <- function(n, g, d, omega_planted = "canonical", lambda = 0.5){

  m <- sum(d)/2 #number of edges
  q <- max(g) #number of communities
  k <- NULL
  for (i in 1:q){
    k[i] <- sum(d[g == i]) #kappas
  }
  theta <- d/(k[g]) #theta vector

  #Matrix Omega:
  omega_random <- matrix(nrow = q, ncol = q)
  for (i in 1:q){
    for (j in 1:q){
      omega_random[i, j] <- (k[i]*k[j])/(2*m)
    }
  }

  if (all(omega_planted == "canonical")){
    omega_planted <- matrix(nrow = q, ncol = q)
    for (i in 1:q){
      for (j in 1:q){
        if (i == j){
          omega_planted[i, j] <- k[i]
```

```
      }else{
        omega_planted[i, j] <- 0
      }
    }
  }
}

omega <- lambda*omega_planted + (1 - lambda)*omega_random


#Step 1: drawing the total number of edges between two groups
  #from a Poisson distribution

ms <- matrix(nrow = q, ncol = q)

for (i in 1:q){
  for (j in i:q){
    if (i == j){
      ms[i, j] <- rpois(1, omega[i, j]/2)
    }else{
      ms[i, j] <- rpois(1, omega[i, j])
    }
  }
}


#Step 2: connecting each edge to dyads according to theta

nodes <- 1:n
edges <- NULL

for (g1 in 1:q){
  for (g2 in g1:q){
    if (ms[g1, g2] > 0){
      for (e in 1:ms[g1, g2]){
```

```
        a <- sample(nodes[g == g1], 1, prob = theta[g == g1])
        b <- sample(nodes[g == g2], 1, prob = theta[g == g2])
        temp <- c(a, b)
        edges <- rbind(edges, temp)
      }
    }
  }
}

#Returning the network

edges <- tibble(from = edges[, 1], to = edges[, 2])
rownames(edges) <- NULL

nodes <- tibble(id = 1:n, group = g)

network <- graph_from_data_frame(d = edges, vertices = nodes, directed = F)

}
```

Code - Node-Moving Algorithm

The following code is the R implementation of the DCSBM-based Karrer and Newman algorithm which has been used in this dissertation. The algorithm itself is at the end (`CD_DCSBM()`), after the definition of some functions it calls. It can be generalized to all methods seen in Section 2 just by changing the function computing the likelihood.

```
#mrs: number of edges between two groups r and s
mrs <- function(network, r, s){

  #adjacency matrix of the network
  adj <- as_adj(network, sparse = F)[V(network)$community == r,
                                     V(network)$community == s]

  temp <- sum(adj) #number of edges between two distinct groups
```

```
  #In the case of r = s:
  if (r == s){
    loops <- sum(diag(adj)) #number of self-loops
    temp <- (temp - loops)/2 + loops
    temp <- temp * 2
    #notice that as_adj counts the number of edges even on the main diagonal
  }

  result <- temp

  result
}

#kappa: sum of degrees of nodes in a group r
kappa <- function(network, r){
  sum(degree(network)[V(network)[V(network)$community == r]])
}

#Log-Likelihood (DC-SBM)
likelihood <- function(network){
  l <- 0
  comms <- unique(V(network)$community)
  for (i in comms){
    for (j in comms){
      if(mrs(network, i, j) == 0){
        l <- l + 0
      }else{
        l <- l +
          mrs(network, i, j)*log(mrs(network, i, j)/(kappa(network, i)*kappa(network, j)))

      }
    }
  }
  l
```

```
}

#Node-moving algorithm
CD_DCSBM <- function(network, q, c = "random") {

  iter <- 1 #Iteration number

  if(all(c == "random")){
    c <- sample(1:q, size = length(V(network)), replace = T)
  } #Set the initial partition

  V(network)$community <- c
  cat("Initial Likelihood:", likelihood(network), "\n")

  #Algorithm iteration
  repeat{
    cat("Iteration:", iter, "\n")

    #Benchmark (likelihood)
    if(iter == 1){
      benchmark_l <- likelihood(network)
    }else{
      benchmark_l <- likes[best_state]
    }

    moved <- length(V(network)) + 1 #to store moved nodes
    likes <- likelihood(network) #to store the likelihood of each state
    inspection_table <- NULL #to implement the best state

    #Repeating the step for each node in the network
    for(step in 1:length(V(network))){
      cat("Step:", step, " ")

      options <- NULL #vector of the likelihood deltas
      register <- NULL #store moves
```

```
  #Find the best possible move among nodes which have not been moved yet
  for (i in V(network)[-moved]){
    ir <- V(network)$community[i]
    trials <- unique(V(network)$community)[unique(V(network)$community) != ir]
    for (s in trials){
      r <- V(network)$community[i]
      V(network)$community[i] <- s
      new_like <- likelihood(network)
      V(network)$community[i] <- r
      options <- c(options, new_like - benchmark_l)
      reg <- data.frame(node = i, to = s)
      register <- rbind(register, reg)
    }
  }

  if (length(which(options == max(options))) > 1){
    mx <- sample(which(options == max(options)), 1)
  }else{
    mx <- which(options == max(options))
  }

  moved_id <- register[mx, ]$node
  new_comm <- register[mx, ]$to
  action <- tibble(id = moved_id, from = V(network)$community[moved_id])

  #Implement the best move
  V(network)$community[moved_id] <- new_comm
  moved <- c(moved, moved_id)
  likes <- c(likes, likelihood(network))
  inspection_table <- rbind(inspection_table, action)
}

if (length(which(likes == max(likes))) > 1){
  best_state <- sample(which(likes == max(likes)), 1)
```

```
    }else{
      best_state <- which(likes == max(likes))
    }

    #Implement the state where the system reached maximum likelihood
    if (best_state <= nrow(inspection_table)){
      temp <- inspection_table[best_state:nrow(inspection_table), ]
      V(network)[temp$id]$community <- temp$from
    }

    #Condition terminating the algorithm
    if(likes[best_state] <= benchmark_l){
      #Algorithm ends
      cat("Stopped.")
      break
    }

    iter <- iter + 1
    cat("Likelihood:", likelihood(network), "\n")
  }
  #Returning the network with attribute 'community' containing the recovered partition
  return(network)
}
```

Code - Barabási–Albert Model with Internal Links

The code below is the R implementation of the Barabási–Albert Model allowing for internal links. The initial number of nodes is fixed at 2.

```
BA_internal <- function(N, m = 2, n = 0){

  m_0 <- 2 #initial nodes
  t <- N - m_0 #time steps
  nodes <- c(1,2)
  edges <- tibble(from = 1, to = 2)

  for (s in 1:t){
```

```
new_node <- max(nodes) + 1 #new node
nodes_d <- NULL
for (j in nodes){
  nodes_d[j] <- sum(edges$from == j) + sum(edges$to == j)
  - sum((edges$from == j) & (edges$to == j))
}

#m new links are created
if (length(nodes) >= m){
  a <- sample(nodes, m, replace = F, prob = nodes_d)
  b <- rep(new_node, m)
}else{
  a <- sample(nodes, length(nodes), replace = F, prob = nodes_d) #PA
  b <- rep(new_node, length(nodes))
}
edges <- rbind(edges, tibble(from = b, to = a))

nodes <- c(nodes, new_node)
nodes_d <- NULL
for (j in nodes){
  nodes_d[j] <- sum(edges$from == j) + sum(edges$to == j)
  - sum((edges$from == j) & (edges$to == j))
}

#After the new node has been connected, if the number of internal
   #links n is greater than 0, n new links are generated
if (n > 0){
  nodes_couple <- NULL
  nodes_couple_d <- NULL
  for (i in nodes){
    for (j in i:max(nodes)){
      nodes_couple <- c(nodes_couple, list(c(i, j)))
      nodes_couple_d <- c(nodes_couple_d, nodes_d[i]*nodes_d[j]) #double PA
    }
  }
```

```
      for (int in 1:n){
        a <- sample(nodes_couple, 1, prob = nodes_couple_d)
        edges <- rbind(edges, c(a[[1]][1], a[[1]][2]))
      }
    }
  }

  #Returning the network
  network <- graph_from_data_frame(d = edges, vertices = nodes, directed = F)
  network
}
```

References

[1] Banca dati AIDA, Bureau van Dijk S.P.A. https://www.bvdinfo.com/.

[2] Implementation of the MCMC algorithm designed by Riolo, Cantwell, Reinert and Newman. www.umich.edu/~mejn/communities.

[3] Food and Agriculture Organization of the United Nations (FAO). http://www.fao.org/faostat/en/#data/TCL.

[4] Emmanuel Abbe. Community detection and stochastic block models: recent developments. *The Journal of Machine Learning Research*, 18(1):6446–6531, 2017.

[5] A.L. Barabási. *Network Science*. Cambridge University Press, 2016.

[6] Manlio De Domenico, Vincenzo Nicosia, Alexandre Arenas, and Vito Latora. Structural reducibility of multilayer networks. *Nature communications*, 6(1):1–9, 2015.

[7] Daniel Gribel, Thibaut Vidal, and Michel Gendreau. Assortative-constrained stochastic block models. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 6212–6218. IEEE, 2021.

[8] Matthew O Jackson. *Social and economic networks*. Princeton university press, 2010.

[9] Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107, 2011.

[10] Clement Lee and Darren J Wilkinson. A review of stochastic block models and extensions for graph clustering. *Applied Network Science*, 4(1):1–50, 2019.

[11] Xiaoyan Lu and Boleslaw K Szymanski. A regularized stochastic block model for the robust community detection in complex networks. *Scientific reports*, 9(1):1–9, 2019.

[12] Mark Newman. *Networks*. Oxford university press, 2018.

[13] Tiago P. Peixoto. Bayesian stochastic blockmodeling. *Advances in Network Clustering and Blockmodeling*, page 289–332, Nov 2019.

[14] Maria A Riolo, George T Cantwell, Gesine Reinert, and Mark EJ Newman. Efficient method for estimating the number of communities in a network. *Physical review e*, 96(3):032310, 2017.

[15] J.L. Stanford and S.B. Vardeman. *Statistical Methods for Physical Science*. Methods of Experimental Physics. Elsevier Science, 1994.

[16] Anne Randi Syversveen. Noninformative bayesian priors. interpretation and problems with construction and applications. *Preprint statistics*, 3(3):1–11, 1998.

[17] Yaojia Zhu, Xiaoran Yan, and Cristopher Moore. Oriented and degree-generated block models: generating and inferring communities with inhomogeneous degree distributions. *Journal of Complex Networks*, 2(1):1–18, 2014.