PROCEEDINGS A

rspa.royalsocietypublishing.org

Research

Check for updates

Article submitted to journal

Subject Areas:
Applied Mathematics and Theoretical Physics, Statistics and Operational Research

Keywords:
core–periphery, spectral methods, low-rank approximation, directed networks

Author for correspondence:
Andrew Elliott
e-mail: aelliott@turing.ac.uk

Core–Periphery Structure in Directed Networks

Andrew Elliott[1,2], Angus Chiu[2], Marya Bazzi[1,3,4], Gesine Reinert[1,2] and Mihai Cucuringu[1,2,4]

[1] The Alan Turing Institute, London, UK
[2] Department of Statistics, University of Oxford, UK
[3] Mathematics Institute, University of Warwick, UK
[4] Mathematical Institute, University of Oxford, UK

Empirical networks often exhibit different meso-scale structures, such as community and core–periphery structure. Core–periphery typically consists of a well-connected core, and a periphery that is well-connected to the core but sparsely connected internally. Most core–periphery studies focus on undirected networks. We propose a generalisation of core–periphery to directed networks. Our approach yields a family of core–periphery blockmodel formulations in which, contrary to many existing approaches, core and periphery sets are edge-direction dependent. We focus on a particular structure consisting of two core sets and two periphery sets, which we motivate empirically. We propose two measures to assess the statistical significance and quality of our novel structure in empirical data, where one often has no ground truth. To detect core–periphery structure in directed networks, we propose three methods adapted from two approaches in the literature, each with a different trade-off between computational complexity and accuracy. We assess the methods on benchmark networks where our methods match or outperform standard methods from the literature, with a likelihood approach achieving the highest accuracy. Applying our methods to three empirical networks – faculty hiring, a world trade data-set, and political blogs – illustrates that our proposed structure provides novel insights in empirical networks.

THE ROYAL SOCIETY PUBLISHING

1. Introduction

Networks provide useful representations of complex systems across many applications [1], such as physical, technological, information, biological, financial, and social systems. A network in its simplest form is a graph in which nodes represent entities and edges represent pairwise interactions between these entities. In this paper, we consider directed unweighted networks.

Given a network representation of a system, it can be useful to investigate so-called meso-scale features that lie between the micro-scale (local nodes properties) and the macro-scale (global network properties). Typical meso-scale structures are community structure (by far the most commonly studied), core–periphery structure, role structure, and hierarchical structure [1–3]; often, more than one of these is present in a network, see for example [2] or [4].

Here we focus on core–periphery structure. The concept of core–periphery was first formalised by Borgatti and Everett [5]. Typically, core–periphery structure is a partition of an undirected network into two sets, a *core* and a *periphery*, such that there are dense connections within the core and sparse connections within the periphery. Furthermore, core nodes are reasonably well-connected to the periphery nodes [5]. Extensions allow for multiple core–periphery pairs and nested core–periphery structures [2,4,6]. Algorithms for detecting (different variants) of core–periphery structure include approaches based on the optimisation of a quality function [2,5,7–9], spectral methods [10–12], and notions of core–periphery based on transport (e.g., core nodes are likely to be on many shortest paths between other nodes in the network) [12,13]. Core–periphery detection has been applied to various fields such as economics, sociology, international relations, journal-to-journal networks, and networks of interactions between scientists; see [14] for a survey.

Many methods for detecting core–periphery were developed for undirected networks. Although these can be (and in some cases have been) generalised to directed graphs, they do not also generalise the definition of a discrete core and periphery to be edge-direction dependent, but rather, either disregard the edge-direction or consider the edge in each direction as an independent observation [2,5,15,16], or use a continuous structure [17]. A notable exception is [18], but with a different notion of core than the one pursued here. The discrete structure which is most closely related to our notion of directed core–periphery structure is the bow-tie structure [19,20]. Bow-tie structure consists of a core (defined as the largest strongly connected component), an in-periphery (all nodes with a directed path to a node in the core), an out-periphery (all nodes with a directed path from a node in the core), and other sets containing any remaining nodes [20–22].

In this paper, we propose a generalisation of the block-model introduced in [5] to directed networks, in which the definition of both core and periphery are edge-direction dependent. Moreover, we suggest a framework for defining cores and peripheries in a way that accounts for edge direction, which yields as special cases a bow-tie-like structure and the structure we focus on in the present paper. Our accompanying technical report explores a small number of additional methods [23]. Extensions to continuous formulations (e.g., as in [24]) or multiple types of meso-scale structure are left to future work.

We suggest three methods to detect the proposed directed core–periphery structure, which each have a different trade-off between accuracy and computational complexity. The first two methods are based on the HITS algorithm [25] and the third on likelihood maximisation. We illustrate the performance of methods on synthetic and empirical networks. Our comparisons to bow-tie structure and illustrate that the structure we propose yields additional insights about empirical networks. Our main contributions are (1) a novel framework for defining cores and peripheries in directed networks; (2) scalable methods for detecting these structures; (3) a comparison of said methods and (4) a systematic approach to method selection for empirical data.

This paper is organised as follows. In Section 2 we consider directed extensions to the classic core–periphery structure. We introduce a novel block-model for directed core–periphery structure that consists of four sets (two periphery sets and two core sets) and a two-parameter synthetic

model that can generate the proposed structure. In Supplementary Information (SI) A, we consider alternative formulations. We further introduce a pair of measures to assess the quality of a detected structure; the first one is a test of statistical significance, and the second one is a quality function that enables comparison between different (statistically significant) partitions. In Section 3, we introduce three methods for detecting the proposed directed core–periphery structure. Section 4 illustrates the performance of our methods on synthetic benchmark networks, and validates the use of our proposed partition quality measures. In Section 5, we apply the methods to two real-world data sets (a third data set is shown in SI E). Section 6 summarises our main results and offers directions for future work.

The code for our proposed methods and the implementation for bow-tie structure (provided by the authors of [26]) is available at `https://github.com/alan-turing-institute/directedCorePeripheryPaper`.

2. Core–periphery structure

We encode the edges of an n-node network in an adjacency matrix $\mathbf{A} = (A_{u,v})_{u,v=1,\ldots,n}$, with entry $A_{u,v} = 1$ when there is an edge from node u to node v, and $A_{u,v} = 0$ otherwise. We partition the set of nodes into core and periphery sets, resulting in a block partition of the adjacency matrix, and a corresponding block probability matrix. In the remainder of the paper, we use the term "set" for members of a node partition and "block" for the partition of a matrix. We shall define a random network model on n nodes partitioned into k blocks via a $k \times k$ probability matrix $\mathbf{M}$, whose entries M_{ij} give the probability of an edge from a node in block i to a node in block j, independently of all other edges.

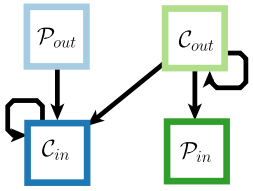
Core–periphery in undirected networks The most well-known quantitative formulation of core–periphery structure in undirected networks was introduced by Borgatti and Everett [5]; they propose both a discrete and a continuous model for core–periphery structure. In the discrete notion of core–periphery structure, [5] suggests that an ideal core–periphery structure should consist of a partition of the node set into two non-overlapping sets: a densely connected core and a loosely connected periphery, with dense connections between the core and the periphery. The probability matrix of a network with the idealised core–periphery structure in [5] and the corresponding network-partition representation are given in (2.1);

$$\mathbf{M_0} = \begin{array}{c|cc} & \text{Core} & \text{Periphery} \\ \hline \text{Core} & 1 & 1 \\ \text{Periphery} & 1 & 0 \end{array} \qquad \text{Core} \quad \text{Per.} \qquad , \tag{2.1}$$

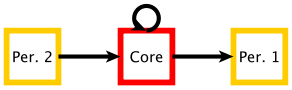
where the network-partition representation on the right-hand-side shows edges within and between core and periphery sets. In adjacency matrices of real-world data sets, any structure of the form Eq. (2.1), if present, is likely observed with random noise perturbations.

Core–periphery structure in directed networks We now introduce a block model for directed core–periphery where the definitions of the core and periphery sets are edge-direction-dependent. Starting from Eq. (2.1), a natural extension to the directed case is to split each of the sets into one that only has incoming edges and another that only has outgoing edges. This yields four sets, which we denote $\mathcal{C}_{in}$ (*core-in*), $\mathcal{C}_{out}$ (*core-out*), $\mathcal{P}_{in}$ (*periphery-in*) and $\mathcal{P}_{out}$ (*periphery-out*), with respective sizes $n_{\mathcal{P}_{out}}$, $n_{\mathcal{C}_{in}}$, $n_{\mathcal{P}_{in}}$, and $n_{\mathcal{C}_{out}}$. We assume that edges do not exist between the periphery sets, and thus that every edge is incident to at least one node in a core set. Respecting edge direction, we place edges between *core-out* and all 'in' sets, and between each 'out' set and *core-in*. As in Eq. (2.1), the two core sets are fully internally connected, and the two periphery sets have no internal edges. There are no multiple edges, but self-loops are permitted. The probability matrix and corresponding network-partition are given in (2.2);

$$\mathbf{M} = \begin{array}{c|cccc} & \mathcal{P}_{out} & \mathcal{C}_{in} & \mathcal{C}_{out} & \mathcal{P}_{in} \\ \hline \mathcal{P}_{out} & 0 & \mathbf{1} & 0 & 0 \\ \mathcal{C}_{in} & 0 & \mathbf{1} & 0 & 0 \\ \mathcal{C}_{out} & 0 & \mathbf{1} & \mathbf{1} & \mathbf{1} \\ \mathcal{P}_{in} & 0 & 0 & 0 & 0 \end{array} \qquad \text{(2.2)}$$



We refer to the structure in $\mathbf{M}$ as an 'L'-shape structure. There are other directed core–periphery structures that one can pursue. In Supplementary Information (SI) A, we provide a framework of which Eq. (2.2) is one example, and a block model formulation of bow-tie structure is another example. The particular formulation of the well-known bow-tie structure that falls within our framework is the directed core–periphery structure Eq. (2.3), where only periphery sets have a definition that is edge-direction dependent, and where we assume that the core and peripheries form a hard partition [22]



(2.3)

In general, bow-tie can allocate nodes to several sets – there is a core set, an incoming periphery set, an outgoing periphery set and four additional sets corresponding to other connection patterns. There are several known real-world applications of bow-tie structure, such as the internet [20] and biological networks [27]. We note that the structure in Eq. (2.2) is not a mere extension of the bow-tie structure as, in contrast to bow-tie, the flow is not uni-directional.

We motivate the structure in Eq. (2.2) with a few examples. Consider networks that represent a type of information flow, with two sets that receive information ($\mathcal{C}_{in}$ and $\mathcal{P}_{in}$) and two sets that send information ($\mathcal{C}_{out}$ and $\mathcal{P}_{out}$). Furthermore, within each of these categories, there is one set with core-like properties and another set with periphery-like properties. Inspired by [3], in a Twitter network for example, $\mathcal{C}_{in}$ and $\mathcal{P}_{in}$ could correspond to consumers of information, with $\mathcal{C}_{in}$ having the added property of being a close-knit community that has internal discussions (e.g., interest groups) rather than individuals collecting information independently (e.g., an average user). The sets $\mathcal{C}_{out}$ and $\mathcal{P}_{out}$ could correspond to transmitters of information, with $\mathcal{C}_{out}$ having the added property of being a well-known close-knit community (e.g., broadcasters) rather than individuals spreading information independently (e.g., celebrities). Another class of examples is networks that represent a type of social flux, when there are two sets that entities move out of, and two sets that entities move towards. Furthermore, within each of these categories, there is one with core-like properties and one with periphery-like properties. For example, in a faculty hiring network of institutions, $\mathcal{C}_{out}$ may correspond to highly-ranked institutions with sought-after alumni, while $\mathcal{C}_{in}$ may correspond to highly sought-after institutions which take in more faculty than they award Ph.D. degrees. For the periphery sets, $\mathcal{P}_{out}$ may correspond to lower-ranked institutions who have placed some faculty in the core but do not attract faculty from higher-ranked institutions, and $\mathcal{P}_{in}$ may correspond to a set of institutions which attract many alumni from highly-ranked ones. These ideas will be showcased on real-world data in Section 5, where we also illustrate that the structure in Eq. (2.2) yields insights that are not captured by the bow-tie structure.

Synthetic model for directed core–periphery structure We now describe a stochastic block model that will be used as a synthetic graph model to benchmark our methods. For any two nodes u, v, let $X(u, v)$ denote the random variable which equals 1 if there is an edge from u to v, and 0 otherwise. We refer to $X(u, v)$ as an edge indicator. For an edge indicator which should equal 1 according to the idealised structure (Eq. (2.2)), let p_1 be the probability that an edge is observed. Similarly for an edge indicator which should be 0 according to the perfect structure (Eq. (2.2)), let p_2 be the probability that an edge is observed. Interpreting p_1 as *signal* and p_2 as *noise*, we assume that $p_1 > p_2$ so that the noise does not overwhelm the true structure in Eq. (2.2). We represent this

model as a stochastic block model, denoted by $DCP(p_1, p_2)$, which has independent edges with block probability matrix

$$p_1\mathbf{M} + p_2(\mathbf{1} - \mathbf{M}) = \begin{vmatrix} p_2 & p_1 & p_2 & p_2 \\ p_2 & p_1 & p_2 & p_2 \\ p_2 & p_1 & p_1 & p_1 \\ p_2 & p_2 & p_2 & p_2 \end{vmatrix}. \tag{2.4}$$

Setting $p_1 = 1$ and $p_2 = 0$ recovers the idealised block structure in Eq. (2.2). The 'L'-shape structure in Eq. (2.4) defines a partition of a network into two cores and two peripheries (see Eq. (2.2) for the idealised case $DCP(1, 0)$). We refer to this partition as a "planted partition" throughout the paper. The $DCP(p_1, p_2)$ model allows one to increase the difficulty of the detection by reducing the difference between p_1 and p_2, and to independently modify the expected density of edges matching (respectively, not matching) the planted partition by varying p_1 (respectively, p_2). A case of particular interest is when only the difference between p_1 and p_2 is varied; this is the $DCP(1/2 + p, 1/2 - p)$ model, where $p \in [0, 0.5]$. This model yields the idealised block structure in Eq. (2.2) when $p = 0.5$, and an Erdős-Rényi random graph when $p = 0$.

Fig. 1 displays example adjacency matrices obtained from Eq. (2.4), with $n = 400$ and equally-sized sets $n_{\mathcal{P}_{out}} = n_{\mathcal{C}_{in}} = n_{\mathcal{C}_{out}} = n_{\mathcal{P}_{in}} = 100$. In the first 3 panels, $p_2 = 0.1$ and p_1 varies. As p_1 decreases with fixed p_2, the 'L'-shaped structure starts to fade away and the network becomes sparser. The last three panels show realisations of $DCP(1/2 + p, 1/2 - p)$ adjacency matrices for $p \in \{0.4, 0.2, 0.05\}$, $n = 400$, and four equally-sized sets. The 'L'-shaped structure is less clear for smaller values of p.

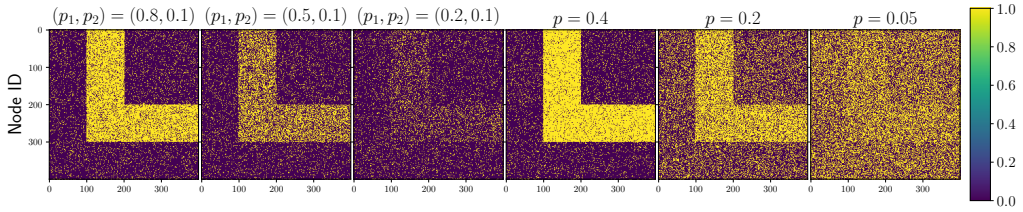


Figure 1. Heatmaps illustrating our model. We present heatmaps of the original adjacency matrix, with $n = 400$ nodes. We generate the first three adjacency matrices with $DCP(p_1, p_2)$ and the next three adjacency matrices with $DCP(1/2 + p, 1/2 - p)$. Blocks are equally-sized in both cases.

Measures of statistical significance and partition quality In empirical networks, there is often no access to ground truth. It is thus crucial to determine whether a detected partition is simply the result of random chance and does not constitute a meaningful division of a network. Furthermore, different detection methods can produce very different partitions (e.g., by making an implicit trade-off between block-size and edge-density), and it can be very helpful in practice to have a systematic approach for choosing between methods according to a specific criteria of "partition quality". As criteria of partition quality, we employ a p-value arising from a Monte Carlo test and an adaptation of the modularity quality function of a partition (see, e.g., Eq. (7.58) in [1]).

The p-value is given by a Monte Carlo test to assess whether the detected structure could plausibly be explained as arising from random chance, modelled either by a directed Erdős-Rényi (ER) model without self-loops or a directed configuration model as in [28]. The test statistic is the difference between the probability of connection within the 'L'-structure, with that outside of the 'L'-structure, i.e.,

$$\frac{\sum_{u,v=1}^{n} M_{g_u,g_v} A_{uv}}{\sum_{u,v=1}^{n} M_{g_u,g_v}} - \frac{\sum_{u,v=1}^{n} (1 - M_{g_u,g_v}) A_{uv}}{\sum_{u,v=1}^{n} (1 - M_{g_u,g_v})},$$

where $\mathbf{M}$ is as in Eq. (2.2), and g_u is the set assign to node u. To directly measure partition quality, we extend the *core–periphery modularity* measure from [4,29], by replacing the block and community indicators with indicators that match the 'L'-structure, i.e.,

$$DCPM(g) = \frac{1}{m}\sum_{u=1}^{n}\sum_{v=1}^{n}\left(A_{uv} - \langle \mathbf{A}\rangle\right) M_{g_u g_v}, \tag{2.5}$$

where m is the number of edges (with bi-directional edges counted twice) and $\langle \mathbf{A}\rangle = \frac{m}{n^2}$. We call this measure *directed core–periphery modularity* ($DCPM$). $DCPM$ lies in the range of $(-1, 1)$. If there is only one block, then $DCPM = 0$. If the 'L'-structure is achieved perfectly, then the number of edges is $m = n_{P_{out}} n_{C_{in}} + (n_{C_{in}})^2 + n_{C_{out}} n_{C_{in}} + (n_{C_{out}})^2 + n_{C_{out}} n_{P_{in}}$ and $DCPM = 1 - \frac{1}{n^2}(n_{P_{out}} n_{C_{in}} + n_{C_{out}}^2 + n_{C_{out}} n_{C_{in}} + n_{C_{in}}^2 + n_{P_{out}} n_{C_{in}}) = 1 - \frac{m}{n^2}$. If instead, all edges not on the 'L' are present, then $DCPM = -(n_{P_{out}} n_{C_{in}} + n_{C_{out}}^2 + n_{C_{out}} n_{C_{in}} + n_{C_{in}}^2 + n_{P_{out}} n_{C_{in}})/n^2$. $DCPM$ is related to the general form core–periphery quality function introduced in [10].

We note that in Eq. (2.5), the null model we compare the observed network against is the expected adjacency matrix under an Erdős-Rényi null model, where each edge is generated with the same probability $\frac{m}{n^2}$, independently of all other potential edges, and the expected number of edges is equal to m, the observed number of edges. Such a null model was used in [4] to derive a quality function for detecting multiple core–periphery pairs in undirected networks. As high-degree nodes tend to end up in core sets, and low-degree nodes in periphery sets (see for example Fig. 4 in this paper), using a null model that controls for node degree directly in the quality function can mask a lot of the underlying core–periphery structure [4,18,29]. To circumvent this issue, the authors in [29] modify the core–periphery block structure definition by incorporating an additional block that is different from the core block and its corresponding periphery block. For the purpose of this paper, we use an Erdős-Rényi null model and leave the exploration of further null models to future work.

For networks with ground truth (e.g., synthetic networks with planted structure), the accuracy of a partition is measured by the Adjusted Rand Index (ARI) [30] between the output partition of a method and the ground truth, using the implementation from [31]. ARI takes values in $[-1, 1]$, with 1 indicating a perfect match, and an expected score of approximately 0 under a given model of randomness. A negative value indicates that the agreement between two partitions is less than what is expected from a random labelling. In SI D(a), we give a detailed description of the ARI, and also consider the alternative similarity measures VOI (Variation of Information [32]) and NMI (Normalised Mutual Information [33]).

3. Core–periphery detection in directed networks

Several challenges arise when considering directed graphs, which makes the immediate extension of existing algorithms from the undirected case difficult. As the adjacency matrix of a directed graph is no longer symmetric, the spectrum becomes complex-valued. Graph clustering methods which have been proposed to handle directed graphs, often consider a symmetrised version of the adjacency matrix, such as SAPA [34]. However, certain structural properties of the network may be lost during the symmetrisation process, which provides motivation for the development of new methods. In this section, we describe three methods for detecting this novel structure. We pay particular attention to scalability, a crucial consideration in empirical networks, and order the methods by run time, from fast to slow. The first two methods are based on an adaptation of the popular HITS algorithm [25], and the third method is based on likelihood-maximisation.

(a) The Hyperlink-Induced Topic Search (HITS) algorithm

Our first method builds on a well-known algorithm in link analysis known as Hyperlink-Induced Topic Search (HITS) [25]. The HITS algorithm was originally designed to measure the importance of webpages using the structure of directed links between the webpages [35]; authoritative webpages on a topic should not only have large in-degrees (i.e., they constitute hyperlinks on

many webpages) but should also considerably overlap in the sets of pages that point to them. Referring to authoritative webpages for a topic as "authorities" and to pages that link to many related authorities as "hubs", it follows that a good hub points to many good authorities, and that a good authority is pointed to by many good hubs. The HITS algorithm assigns two scores to each of the n nodes, yielding a n-dimensional vector $\mathbf{a}$ of "authority scores" and a n-dimensional vector $\mathbf{h}$ of "hub scores", with $\mathbf{a} = \mathbf{A}^T\mathbf{h}$ and $\mathbf{h} = \mathbf{A}\mathbf{a}$.

To each node we assign core– and periphery–scores based on the HITS algorithm which we then cluster to obtain a hard partition; we call this the HITS method. Appealing features of the HITS algorithm include (1) it is highly scalable; (2) it can be adapted to weighted networks; and (3) it offers some theoretical guarantees on the convergence of the iterative algorithm (see [25]).

Algorithm for HITS

(i) Initialisation: $\mathbf{a} = \mathbf{h} = \mathbf{1}_n$. Alternate between the following two steps: (a) update $\mathbf{a} = \mathbf{A}^T\mathbf{h}$; (b) update: $\mathbf{h} = \mathbf{A}\mathbf{a}$. Stop when the change in updates is lower than a pre-defined threshold.
(ii) Normalise $\mathbf{a}$ and $\mathbf{h}$ to become unit vectors in some norm [35].
(iii) Compute the $n \times 4$ score matrix $\mathbf{S}^{HITS} = [\mathbf{P}_{out}^{HITS}, \mathbf{C}_{in}^{HITS}, \mathbf{C}_{out}^{HITS}, \mathbf{P}_{in}^{HITS}]$ using the node scores

$$C_{in}^{HITS}(u) = h(u), \qquad P_{in}^{HITS}(u) = \max_v(C_{out}^{HITS}(v)) - C_{out}^{HITS}(u)\,, \tag{3.1}$$

$$C_{out}^{HITS}(u) = a(u), \qquad P_{out}^{HITS}(u) = \max_v(C_{in}^{HITS}(v)) - C_{in}^{HITS}(u)\,. \tag{3.2}$$

(iv) Normalise $\mathbf{S}^{HITS}$ so that each row has an L_2–norm of 1 and apply k-means++ to partition the node set into four clusters;
(v) Assign each of the clusters to a set based on the likelihood of each assignment under our stochastic block model formulation (see Section 2).

Remark 3.1. *(i) To motivate the scores Eqs. (3.1) and (3.2), a node should have a high authority score if it has many incoming edges, whereas it would have a high hub score if it has many outgoing edges. Based on the idealised block structure in Eq. (2.2), nodes with the highest authority scores should also have a high $\mathbf{C}_{in}^{HITS}$ score, and nodes with the highest hub scores should also have a high $\mathbf{C}_{out}^{HITS}$ score.*
(ii) For step (i) of the algorithm we use the implementation from NetworkX [36] which computes the hub and authority scores using the leading eigenvector of $\mathbf{A}^T\mathbf{A}$. As [25] proved that the scores converge to the principal left and right singular vectors of $\mathbf{A}$, provided that the initial vectors are not orthogonal to the principal eigenvectors of $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$, this is a valid approach.
(iii) Using the same connection between the HITS algorithm and SVD from Kleinberg [37], our scores based on the HITS algorithm can be construed as a variant of the low-rank method in [12], in which we only consider a rank-1 approximation and use the SVD components directly.
(iv) A scoring variant is explored in SI B, with Eqs. (3.1) and (3.2) performing best on our benchmarks.
(v) Intuitively, the row normalisation of $\mathbf{S}^{HITS}$ from step (iv) allows the rows of $\mathbf{S}^{HITS}$ (vectors in 4-dimensional space) to not only concentrate in four different directions, but also to concentrate in a spatial sense and have a small within-set Euclidean distance [38,39].
(vi) Using k-means++ [40] alleviates the issues of unstable clusterings retrieved by k-means [41].

(b) The Advanced Hits method

We now modify the HITS algorithm such that it considers four distinct scores (rather than two core scores, from which we then compute the periphery scores); we call the resulting method the *Advanced Hits* method, and abbreviate the corresponding algorithm as ADVHITS. We do this by incorporating information about the idealised block structure into the algorithm (which, as we show in Section 4, yields better results on synthetic networks). Namely, instead of using hub and authority scores, in each set, we reward a node for having edge indicators that match the structure

in Eq. (2.2) and penalise otherwise, through the reward-penalty matrix associated to $\mathbf{M}$, given by

$$\mathbf{D} = 2\mathbf{M} - 1 = \begin{vmatrix} -1 & 1 & -1 & -1 \\ -1 & 1 & -1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 \end{vmatrix} = \begin{vmatrix} \mathbf{d}_1 & \mathbf{d}_2 & \mathbf{d}_3 & \mathbf{d}_4 \end{vmatrix} = \begin{vmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \\ \mathbf{e}_4 \end{vmatrix},$$

where $\mathbf{d_i}$ is the i^{th} column vector of $\mathbf{D}$, and $\mathbf{e_j}$ is the j^{th} row vector of $\mathbf{D}$. The first column/row corresponds to $\mathcal{P}_{out}$, the second column/row to $\mathcal{C}_{in}$, and so on. We use the matrix $\mathbf{D}$ to define the AdvHits algorithm, with steps detailed below.

Algorithm for AdvHits

(i) Initialisation:

$$\mathbf{S}^{Raw} = [\mathbf{S}_1^{Raw}, \mathbf{S}_2^{Raw}, \mathbf{S}_3^{Raw}, \mathbf{S}_4^{Raw}] = [\mathbf{P}_{out}^{Raw}, \mathbf{C}_{in}^{Raw}, \mathbf{C}_{out}^{Raw}, \mathbf{P}_{in}^{Raw}] = \mathbf{U}_n\,,$$

where $\mathbf{U}_n$ is a $n \times 4$ matrix of independently drawn uniform $(0, 1)$ random variables;

(ii) For nodes $u \in \{1, \ldots, n\}$ let $B(u) = \min\{P_{out}^{Raw}(u), C_{in}^{Raw}(u), C_{out}^{Raw}(u), P_{in}^{Raw}(u)\}$, and calculate, for sets $i \in \{1, 2, 3, 4\}$,

$$S_i^{Nrm}(u) = \frac{S_i^{Raw}(u) - B(u)}{\sum_{k=1}^{4} \left(S_k^{Raw}(u) - B(u)\right)}. \tag{3.3}$$

If for a node u, the raw scores for each sets are equal, up to floating point error (defined as the denominator of Eq. (3.3) being less than 10^{-10}), this implies an equal affinity to each set and thus we set $S_i^{Nrm}(j) = 0.25$.

(iii) For $i \in \{1, \ldots, 4\}$:

(a) Update S_i^{Raw}:

$$\begin{aligned} S_i^{Raw} = &\left(1 - \frac{m}{n^2}\right) \mathbf{A}\mathbf{S}^{Nrm}\mathbf{e}_i^T + \frac{m}{n^2}(1 - \mathbf{A})\mathbf{S}^{Nrm}(-\mathbf{e}_i^T) \\ &+ \left(1 - \frac{m}{n^2}\right) \mathbf{A}^T\mathbf{S}^{Nrm}\mathbf{d}_i + \frac{m}{n^2}(1 - \mathbf{A}^T)\mathbf{S}^{Nrm}(-\mathbf{d}_i). \end{aligned} \tag{3.4}$$

(b) Recompute S^{Nrm} using the procedure in step (ii).

(c) Measure and record the change in S_i^{Nrm}.

(iv) If the largest change observed in S_i^{Nrm} is greater than 10^{-8}; return to step (iii).

(v) Apply k-means++ to $\mathbf{S}^{Nrm}$ to partition the node set into four clusters.

(vi) Assign each of the clusters to a set based on the likelihood of each assignment under our stochastic block model formulation (see Section 2).

Remark 3.2. *(i) The first term in Eq. (3.4) rewards/penalises the outgoing edges, the second the missing outgoing edges, the third the incoming edges, and the fourth the missing incoming edges. The multiplicative constants are chosen to weigh edges in each direction evenly, and to fix the contribution of non-edges to be equal to that of edges.*

(ii) We envision the score to represent the affinity of a given node to each set. Thus, the normalisation step is included so that the scores of an individual node sum to one. We include $B(u)$ as the scores in Eq. (3.4) can be negative and thus we shift the values to be all positive (and rescale).

(iii) The general iteration can fail to converge within 1000 iterations. If the scheme has not converged after 1000 steps, we fall back to a scheme which updates the scores on each node in turn, which often empirically removes the convergence issue with the cost of additional computational complexity.

(c) Likelihood maximisation

Our third proposed method, MaxLike, maximises the likelihood of the directed core–periphery model Eq. (2.4), which is a stochastic block model with four blocks and our particular connection structure. To maximise the likelihood numerically we use a procedure from [42] which we call MaxLike, it updates the set assignment of the node that maximally increases/minimally

decreases the likelihood at each step, and then repeats the procedure with remaining non-updated nodes. The complete algorithm is given in SI C. For multimodal or shallow likelihood surfaces, the maximum likelihood algorithms may fail to detect the maximum, and instead find a local optimum. To alleviate this concern, we use a range of initial values for the algorithms.

In our preliminary analysis, we also employed a related faster, greedy likelihood maximisation algorithm. We found that MAXLIKE slightly outperformed the faster approach on accuracy, and hence do not present the fast greedy method here.

4. Numerical Experiments on Synthetic Data

In order to compare the performance of the methods from Section 3, we create three benchmarks using the synthetic model $DCP(p_1, p_2)$ from Section 2. Leveraging the fact that we have access to a ground truth partition (here, a planted partition), the purpose of these benchmarks is (1) to compare our approaches to other methods from the literature; and (2), to assess the effectiveness of the p-value and the $DCPM$ as indicators of core–periphery structure. We also use the benchmark to assess the run time of the algorithms. For the methods comparison, we compare HITS, ADVHITS and MAXLIKE to a naïve classifier (DEG.), which performs k-means++ [40], clustering solely on the in- and out- degree of each node; we also compare them against two well-known fast approaches for directed networks, namely SAPA from [34] and DISUM from [43]; implementation details and variants can be found in SI D. For brevity, we only include the best performing SAPA and DISUM variant, namely SAPA2, using degree-discounted symmetrisation, and DISUM3, a combined row and column clustering into four sets, using the concatenation of the left and right singular vectors. Both SAPA and DISUM perform degree normalisation which may limit their performance. Moreover, our methods are compared against the stochastic block modelling fitting approach GRAPHTOOL [44], based on [2,45], which minimises the minimum description length of the observed data. To make this a fair comparison, we do not use a degree corrected block model but instead a standard stochastic block model, and we fix the number of sets at four.

The second goal is to assess on synthetic networks whether our ranking of method performance based on p-value and $DCPM$ is qualitatively robust across measures that do not require knowledge of a ground truth partition. To this end, we compare these rankings to those obtained with measures that do leverage ground truth, namely the ARI.

(a) Results for the Benchmark Networks

Benchmark 1 We test our approaches using our 1-parameter SBM $DCP(1/2 + p, 1/2 - p)$, with equally-sized sets, and varying $p \in \{0.5, 0.49, 0.48, \ldots, 0.21\} \cup \{0.195, 0.19, 0.185, \ldots, 0.005\}$, the finer discretisation step zooming in on the parameter regime which corresponds to the planted partition being weak. We average over 50 network samples for each value of p. Recall that, for $p = 0.5$, the planted partition corresponds to the idealised block structure in Eq. (2.2) and for $p = 0$ the planted partition corresponds to an Erdős-Rényi random graph with edge probability 0.5.

The performance results for sets of size 100 ($n = 400$) are shown in Table 1, giving the ARI for $p = 0.4$ and for values of p between 0.1 and 0.02 with step size 0.01, in decreasing order (with results for the full parameter sweep in SI D).

With regards to ARI, MAXLIKE performs best for p in the range of 0.1 to 0.03., with performance deteriorating when the noise approaches the signal. Above a certain threshold of p (roughly around $p = 0.25$, results shown in Fig. SI 1 in SI D), many approaches, including the degree-based one DEG., achieve optimal performance, indicating that in this region of the networks obtained with Benchmark 1, the degrees alone are sufficient to uncover the structure. For NMI and VOI, we observe similar qualitative results, see SI D.

The performance of GRAPHTOOL collapses as p gets close to 0 (similar behaviour is observed for $n = 1000$ see SI D). Further investigation indicated that for low values of p, GRAPHTOOL often places most nodes in a single set (see SI D for further details).

p	0.4	0.1	0.09	0.08	0.07	0.06	0.05	0.04	0.03	0.02
DEG	**1.0**	0.878	0.819	0.753	0.663	0.536	0.408	0.281	0.163	0.0767
DISUM	0.995	0.383	0.277	0.193	0.117	0.0506	0.0171	0.00651	0.0021	0.000614
SAPA	**1.0**	0.405	0.276	0.202	0.144	0.0811	0.0306	0.00809	0.00274	0.00085
GRAPHTOOL	**1.0**	0.996	0.985	0.968	0.921	0.655	0.0104	0.000119	2.08e-05	2.73e-05
HITS	**1.0**	0.909	0.852	0.78	0.692	0.562	0.423	0.275	0.152	0.071
ADVHITS	**1.0**	0.972	0.946	0.901	0.814	0.693	0.525	0.333	0.168	**0.0777**
MAXLIKE	**1.0**	**0.997**	**0.986**	**0.971**	**0.931**	**0.831**	**0.675**	**0.42**	**0.195**	0.0577

Table 1. Average ARI of the methods under comparison on Benchmark 1 ($DCP(1/2+p, 1/2-p)$) for different values of p, and with network size $n = 400$. The largest values for each column are given in boldface.

Benchmark 1 is also used to assess the run time of the algorithms. The slowest of our methods across all values of p is MAXLIKE. For small p, HITS is the fastest of our methods, whereas for larger p it can be overtaken by ADVHITS; both are faster than GRAPHTOOL. Within methods, the performance is relatively constant for HITS while it speeds up for decreasing p in ADVHITS and MAXLIKE. The detailed results can be found in SI D.

Benchmark 2 We use the model $DCP(p_1, p_2)$, again with all four sets of the same size $\frac{n}{4}$. In this model, the edge probabilities (p_1, p_2) vary the density and the strength of the core periphery structure independently. To this end, we vary p_1 and the ratio $0 \leq \frac{p_2}{p_1} < 1$. For a given p_1, $\frac{p_2}{p_1} = 0$ corresponds to the strongest structure, and $\frac{p_2}{p_1} = 1$ to the weakest structure. We generate 50 networks each with $p_1 \in \{0.025, 0.05, \ldots, 1.0\}$ and $\frac{p_2}{p_1} \in \{0, 0.05, \ldots, 0.95\}$, resulting in 820 parameter instances of $(p_1, \frac{p_2}{p_1})$. The contours corresponding to an average ARI of 0.75 and an average ARI of 0.9 for $n = 400$ and $n = 1000$ are shown in SI D.

Similarly as in Benchmark 1, the full likelihood approach MAXLIKE outperforms all other methods, with GRAPHTOOL also performing well, and the performance of ADVHITS coming close and outperforming GRAPHTOOL in certain regions.

Benchmark 3 Benchmark 3 assesses the sensitivity of our methods to different set sizes. We use the model $DCP(1/2+p, 1/2-p)$. We fix $p = 0.1$, as we observed in Table 1 that this value is sufficiently small to highlight variation in performance between our approaches, but sufficiently large that most of the methods can detect the underlying structure. We then consider the effect of size variation for each set in turn, by fixing the size of the remaining three sets. For example, to vary the size of $\mathcal{P}_{out}$, we fix $n_{\mathcal{C}_{in}} = n_{\mathcal{C}_{out}} = n_{\mathcal{P}_{in}} = n_1$ and test performance when we let $n_{\mathcal{P}_{out}} = n_2 \in \{2^{-3}n_1, 2^{-2}n_1, \ldots, 2^3 n_1\}$, with equivalent formulations for the other sets. Thus for $\frac{n_2}{n_1} = 1$ we have equal-sized sets, which is equivalent to the model in Benchmark 1, for $\frac{n_2}{n_1} > 1$ one set is larger than the remaining sets, and for $\frac{n_2}{n_1} < 1$, one set is smaller than the others.

Results are shown in SI D for $n_1 = 100$ ($\frac{n_2}{n_1} = 1$ implies a 400 node network). MAXLIKE slightly outperforms GRAPHTOOL, and is the overall best performer, appearing to be robust to set size changes. ADVHITS usually outperforms the other approaches, however, for larger sets, the ADVHITS is in some cases even outperformed by DEG.

(b) Performance of the p-value and $DCPM$ to capture ground truth

To investigate whether the p-values and $DCPM$ introduced in Section 2 are appropriate to assess partition quality, we test the relationship between our proposed quality measures and ARI on a set of benchmark networks. We create these networks using the synthetic model for Benchmark 1, i.e., $DCP(1/2+p, 1/2-p)$, with three values of p focusing on the region where the planted partition is detectable ($p = 0.1$); marginally detectable ($p = 0.04$); and (mostly) undetectable ($p = 0.02$). We note that, for large p, all of the methods will be able to uncover the exact partition and thus each partition would have an ARI of 1 (Table 1), with differences in DCPM driven by the strength of

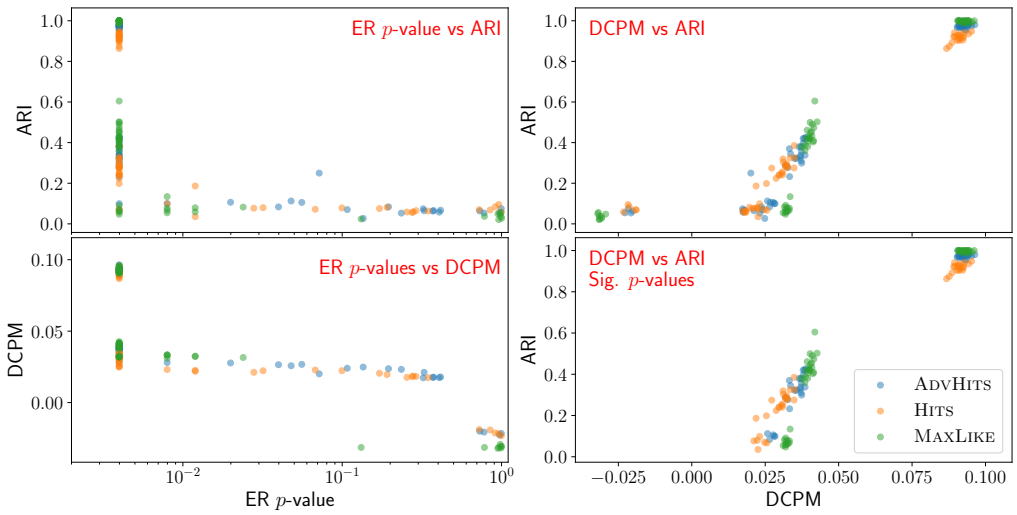


Figure 2. Scatter plots for p-value, $DCPM$ and ARI, using the partitions given by each of our methods on networks taken from $DCP(1/2+p, 1/2-p)$ with $p \in [0.015, 0.04, 0.1]$, with 20 networks for each p. Upper left panel: ER model p-value against ARI. Upper right panel: $DCPM$ against ARI. Lower left panel: ER model p-value against $DCPM$. Lower right panel: ARI against DCPM using only networks that are significant (p-value< 0.05) in both the ER model and the configuration model test. The colour of each of the points represents the method used.

the embedded structure. For computational reasons, we restrict the experiment to 20 networks for each p, and use 250 null replicates for each Monte Carlo test. Each of our three methods is applied to each network, and thus each network gives rise to three p-values and $DCPM$ values.

For good partitions, the ARI should be high, the p-value should be low, and the $DCPM$ value should be high. Hence ARI and p-value should be negatively correlated, p-value and $DCPM$ should be negatively correlated, while ARI and $DCPM$ should be positively correlated. For robustness, we assess correlation by Kendall's τ rank correlation coefficient. For both the Erdős-Rényi (ER) and configuration model p-values, we observe a moderate negative correlation with ARI (ER: -0.599, Configuration: -0.506, data for configuration model not shown). The correlation between $DCPM$ and ER p-value is -0.655, and the correlation between $DCPM$ and ARI is 0.774. The upper left panel of Fig. 2 illustrates selecting partitions with ER p-value less than 0.05 is successful at filtering out partitions with a low ARI, but struggles to separate partitions with mid range ARI from networks with high ARI. Focusing only on network partitions with a p-value of less than 0.05 in both the ER and the configuration model test, as shown in the lower right panel of Fig. 2, we note that $DCPM$ further differentiates the partitions with low p-value and gives a correlation of 0.774 with ARI. The direction of all of these correlations are as expected. If the observations were independent, then these correlations would be highly statistically significant. Thus, while not conclusive evidence, the level of correlation supports the use of our p-value test and $DCPM$ to identify partitions.

As further support for this claim, Table 2 presents the average ER and configuration p-value, average DCPM values, and average ARI, broken down by method and model parameter. As expected for good partitions, we observe low p-values for strong structures ($p = 0.1$, ARI> 0.9), higher p-values for weaker structures ($p = 0.04$, $0.25 <$ARI< 0.45), and non-significant p-values for very weak or non-existent structures ($p = 0.02$, ARI<0.1).[1] In particular, whenever average ARI ≥ 0.4 in Table 2, all p-values are significant. Thus, we find that both the p-value and the $DCPM$ can be used as proxy for the ARI, displaying a moderate correlation. The $DCPM$ is particularly useful to extract more detailed information for partitions which exhibit low p-values. In particular,

[1]For completeness, we display the sample standard deviation for all methods in SI D.

p	0.1				0.04				0.02			
	p-value				p-value				p-value			
	ER	Con.	$DCPM$	ARI	ER	Con.	$DCPM$	ARI	ER	Con.	$DCPM$	ARI
HITS	0.004	0.004	0.091	0.916	0.004	0.004	0.031	0.274	0.325	0.269	0.011	0.071
ADVHITS	0.004	0.004	0.093	0.974	0.007	0.008	0.035	0.340	0.327	0.412	0.014	0.074
MAXLIKE	0.004	0.004	0.093	0.997	0.004	0.004	0.040	0.439	0.344	0.4	0.007	0.059

Table 2. Average p-value (ER and configuration model), $DCPM$ and ARI, over 20 networks, with a breakdown by method and parameter in a $DCP(1/2+p, 1/2-p)$ model; p-values are rounded to 3 dp. The corresponding sample standard deviations are shown in Table 5 in SI D.

Table 2 and SI D indicate that using average $DCPM$ as an approach to rank methods, overall yields qualitatively similar results to ARI.

In Table 2, MAXLIKE and ADVHITS tend to have the highest average $DCPM$ and ARI. In SI D we show that this observation is robust across further values of p. Overall, our ranking of method performance based on average partition quality values is thus robust across $DCPM$ and ARI, for different values of p in $DCP(1/2+p, 1/2-p)$.

To illuminate the relationship between $DCPM$ and ARI further, for $p = 0.1$ we observe a Kendall correlation of 0.315 between them across methods; for $p = 0.04$ this correlation increases to 0.753, while for $p = 0.02$ the correlation decreases to 0.367 (all rounded to 3 dp). For $p = 0.1$ there is little noise and hence variation in $DCPM$, ranging between 0.0868 and 0.0964, nor in ARI, ranging from 0.863 to 1; the structure is so strong that much of it is picked up by the methods, and the noise which both methods pick up will be small and a Kendall correlation will mainly relate to this noise. For $p = 0.04$ there is a moderate signal; $DCPM$ ranges between 0.020 and 0.0427 while ARI ranges between 0.0186 and 0.605. Here the strong correlation between $DCPM$ and ARI supports the value of $DCPM$ as proxy for ARI in choosing partitions which resemble the ground truth. For $p = 0.02$ there is little signal in the data and hence $DCPM$ and ARI will be noisy; $DCPM$ here ranges between -0.032 and 0.033, while ARI ranges between 0.021 and 0.132. Due to the high level of noise, none of the methods will tend to give very good partitions, and the correlation between the measures will be relatively weak. Notably, in all cases the correlation is larger than 0.3, revealing a moderate correlation across the range.

(c) Procedure

Our procedure to select between methods and partitions in a systematic manner is as follows.

Procedure:

(i) Compute partitions using each computationally tractable method;

(ii) For each partition, use our Monte Carlo test to see if it deviates from random, both with respect to ER and to the directed configuration model, and exclude the partitions that are not significant;

(iii) Rank the selected significant partitions for further analysis using $DCPM$.

5. Application to real world data

In this section, we apply our methods to three real-world data sets, namely Faculty Hiring data (*Faculty*) from [46] (Section 5(a)), Trade data (*Trade*) from [47] (Section 5(b)), and Political Blogs (*Blogs*) from [48] (presented in the SI F for brevity). In each case, our methods find a division into four sets, and we explore the identified structure using known underlying attributes. We use the procedure which we validated on synthetic data in Section 4, using $DCPM$ to only rank partitions with significant p-values. We also assess the consistency of the partitions, both within and across each of the approaches, by computing the within-method ARI between the resultant partitions and the ARI between methods of different types.

	Faculty Hiring			World Trade			Political Blogs		
	p-value			p-value			p-value		
	ER	Config	DCPM	ER	Config	DCPM	ER	Config	DCPM
HITS	1.0	0.876	-0.409	1.0	1.0.	-0.60	0.004	0.960	0.384
ADVHITS	0.004	0.004	0.390	0.004	0.004	0.65	0.004	0.004	0.594
MAXLIKE	0.004	0.004	**0.507**	0.004	0.008	**0.72**	0.004	0.004	**0.652**

Figure 3. (**Top**) Performance of the methods on each of the real world data sets. The p-values are computed using our Monte Carlo test with 250 samples from the null distribution. The values have been rounded to 3dp. The largest values of $DCPM$ (from Section 2) for each data set are given in boldface. (**Bottom**) The ARI between the partitions uncovered by each method, in (**Left**) *Faculty*, (**Right**) *Trade*. Negative values are set to 0. For our methods we compare with 11 runs and show the average similarity between all pairs of partitions whereas for bow-tie, we use a single run (the algorithm is deterministic) and thus display a blank (white) square on the corresponding diagonal blocks. To compare to bow-tie, we compare both to the partition into 7 sets and the BOWTIEADJ partition formed by a subset of the nodes corresponding to the main three sets.

Moreover, we compare the partitions with the structure uncovered by bow-tie [20], as discussed in Section 2. As bow-tie allocates nodes to 7 sets, we consider the ARI between the partition into 7 sets (BOWTIE), and the partition induced only by the core set and the in- and out-periphery sets (BOWTIEADJ). When computing the ARI between the partition given by BOWTIEADJ with another partition $\mathcal{S}$, we consider the partition induced by $\mathcal{S}$ on the node-set in BOWTIEADJ (by construction, the ARI between BOWTIEADJ and BOWTIE is always 1).

Fig. 3 (top) shows a summary table for the three real-world data sets; the p-values correlate with the $DCPM$ measure on all three data sets, and the value of $DCPM$ is always highest for the likelihood approach. We thus focus our interpretation on the output partition obtained with MAXLIKE.

(a) Faculty Hiring

In the faculty hiring network from [46], nodes are academic institutions, and a directed edge from institution u to v indicates that an academic received their PhD at u and then became faculty at v. The data set is divided by gender, faculty position, and into three fields (Business, Computer Science, and History). For brevity, we only consider the overall connection pattern in Computer Science. This list includes 23 Canadian institutions in addition to 182 American institutions, The data were collected between May 2011 and March 2012. They include 5,032 faculty, of whom 2,400 are full professors, 1,772 associate professors, and 860 assistant professors; 87% of these faculty received doctorates which were granted by institutions within the sampled set. In [46], it is found that a large percentage of the faculty is trained by a small number of institutions, and it is suggested that there exists a core–periphery-like structure in the faculty hiring network.

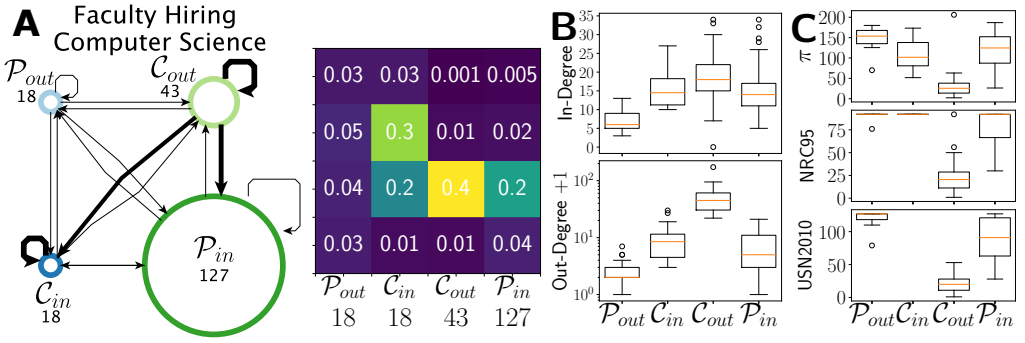


Figure 4. Structures in *Faculty*. Summary network diagram associated with the uncovered structure for MAXLIKE. The size of each of the nodes is proportional to the number of nodes in the corresponding set, and the width of the lines is given by the percentage of edges that are present between the sets. Partitions in *Faculty*. **A** - Boxplot of in- and out-degrees in each of the sets in MAXLIKE. **B** - Boxplot of in- and out-degrees in each of the sets in ADVHITS. To visualise the out-degrees on a log scale, we add 1 to the degrees. **C** - Boxplot of the ranking in [46], denoted π, ranking in NRC95 and the ranking in USN2010 in each of the sets in MAXLIKE. If a ranking is not reported for an institution, we exclude the institution from the boxplot.

We apply our procedure to this data set, and find that the results from the ADVHITS variants and the likelihood method MAXLIKE are significant at 5% under both random null models, whereas the other approaches are not (Fig. 3). Next, we consider the $DCPM$ score between the significant partitions (Fig. 3), and note that, MAXLIKE (0.507) yields a stronger structure than ADVHITS (0.390), and hence we focus on the MAXLIKE partition, which is shown in Fig. 4.

The results in Fig. 4 show a clear 'L'-shape structure, albeit with a weakly defined $\mathcal{P}_{out}$. To interpret these sets, we first compare them against several university rankings. In each of the sets found using MAXLIKE, Fig. 4C shows the University ranking π obtained by [46], and the two other University rankings used in [46], abbreviated NRC95 and USN2010. Here, the NRC95 ranking from 1995 was used because the computer science community rejected the 2010 NRC ranking for computer science as inaccurate. The NRC ranked only a subset of the institutions; all other institutions were assigned the highest NRC rank $+1 = 92$. The set $\mathcal{C}_{out}$ has considerably smaller ranks than the other sets, indicating that $\mathcal{C}_{out}$ is enriched for highly ranked institutions. Upon inspection, we find that $\mathcal{C}_{out}$ consists of institutions including Harvard, Stanford, MIT and also a node that represents institutions outside of the data set. The set $\mathcal{P}_{in}$ from MAXLIKE appears to represent a second tier of institutions who take academics from the schools in $\mathcal{C}_{out}$ (Fig. 4) but do not return them to the job market. This observation can again be validated by considering the rankings in [46] (Fig. 4C). The $\mathcal{C}_{in}$ set loosely fits the expected structure with a strong incoming link from $\mathcal{C}_{out}$ and a strong internal connection (Fig. 4), suggesting a different role to that of the institutions in $\mathcal{P}_{in}$. A visual inspection of the nodes in $\mathcal{C}_{in}$ reveals that 100% of the institutions in $\mathcal{C}_{in}$ are Canadian (also explaining the lack of ranking in USN2010 (Fig. 4C)). In contrast, the proportion of Canadian universities in $\mathcal{P}_{out}$ is 11.1%, in $\mathcal{C}_{out}$ it is 2.3%, and in $\mathcal{P}_{in}$ it is 0.79%. This finding suggests that Canadian universities tend to play a structurally different role to US universities, tending to recruit faculty from other Canadian universities, as well as from the top US schools. In [46], the insularity of Canada was already noted, but without a core–periphery interpretation. One possible interpretation of this grouping is salary. In 2012 it was found that Canadian public universities offered a better faculty pay on average compared to US public universities; see [49].

Finally, $\mathcal{P}_{out}$ is weakly connected both internally and to the remainder of the network and does not strongly match the 'L'-structure (Fig. 4). In each of the rankings (Fig. 4C), $\mathcal{P}_{out}$ has slightly lower average ranks than the other sets (with the exception of $\mathcal{C}_{in}$, due to the default/missing rankings of Canadian institutions). This could indicate that $\mathcal{P}_{out}$ consists of

lower ranked institutions which are not strong enough to attract faculty from the larger set of institutions. The in- and out-degree distributions, (Fig. 4B), show that $\mathcal{P}_{out}$ has lower in- and out-degree distributions than the other sets. Thus, an alternative hypothesis is that $\mathcal{P}_{out}$ consists of universities with smaller Computer Science departments which do not interact with the wider network. We leave addressing this interpretation to future work. In either case, the institutions in $\mathcal{P}_{out}$ do not appear to match the pattern observed in the remainder of the network and hence it is plausible to delegate them into one set.

Overall, in this real-world data set, we demonstrated the power of our method by uncovering an interesting structure that includes a $\mathcal{C}_{in}$ which captures Canadian Universities that appear to recruit faculty from top ranked US institutions, but also recruit from other Canadian institutions in $\mathcal{C}_{in}$.

(b) World Trade

The World Trade network from [47] has countries as nodes and directed edges between countries representing trade. For simplicity, we focus on data from the year 2000 and restrict our attention to the trade in *"Armoured fighting vehicles, war firearms, ammunition, parts"* (the SITC class 9510). We remove trades that do not correspond to a specific country, resulting in a total of 256 trades involving 101 countries, which leads to a network density of roughly 0.025.

Following our procedure, we first consider the p-values of our Monte Carlo test. ADVHITS and MAXLIKE show significant deviation from random when compared to the directed ER and directed configuration models (Fig. 3). When calculating the $DCPM$ for statistically significant partitions, we observe a similar ordering to that of the *Faculty* data set results, with MAXLIKE having the highest $DCPM$ (0.72), ADVHITS having the second highest $DCPM$ (0.65), and finally HITS with a $DCPM$ of -0.60.

The ARIs in Fig. 3 show considerable similarity between the MAXLIKE and ADVHITS, with a weaker similarity between HITS and the BOWTIE variants. Considering the similarity with BOWTIE, the connected component based BOWTIE performs better on this sparser data set, producing 4 sizable sets and 2 singleton sets (unlike in *Faculty* with 2 sizable sets and 1 singleton set). However, while there is some similarity with our partitions (as demonstrated by a larger value of ARI), the structures captured by each approach are distinct and complementary. For example, focusing on the structure with the highest DCPM (MAXLIKE), the BOWTIE 'core' combines our $\mathcal{P}_{out}$ and $\mathcal{C}_{out}$, capturing $\approx 93\%$ of the nodes in $\mathcal{P}_{out}$ (26) and $\approx 82\%$ of the nodes in $\mathcal{C}_{out}$ (9). Overall, this demonstrates that in this data set, BOWTIE does not distinguish between what we will demonstrate below are two distinct structural roles. Furthermore, BOWTIE splits our $\mathcal{P}_{in}$ set into two. A similar comparison of the division of the sets holds between BOWTIE and ADVHITS, indicating that the differences between BOWTIE and methods to which it is similar in Fig. 3 methods are robust.

Following our procedure, we now focus on the structure with the highest DCPM (MAXLIKE). It has the 'L' shaped structure (Fig. 5 top left panel), with smaller core sets and larger periphery sets. To support our interpretation of the structures, we also present summaries of some of their covariates for the year 2000, namely GDP per capita, research spend, and military spend, the latter two as a percentage of GDP. We obtain these covariates from the World Bank using the 'wbdata' package [50], using 'GDP per capita (current US$)' licensed under CC-BY 4.0 [51], 'Military expenditure' (% of GDP) from the Stockholm International Peace Research Institute [52], and 'Research and development expenditure (% of GDP)' from the UNESCO Institute for Statistics and licensed under CC BY-4.0 [53]. Not all country covariate pairs have the covariate data available. For completeness, in the last line of Fig. 5, we report the percentage of data points we have available, split by covariate and group.

From Fig. 5, key patterns emerge, with $\mathcal{C}_{out}$ consisting of somewhat wealthy countries, with a higher research spend as a percentage of GDP and a high density of export links. This set includes several European countries (France/Monaco, Germany, Italy, UK, Switzerland/Liechtenstein, the Czech Republic, and Slovakia), as well as Russia, China, Iran and South Africa.

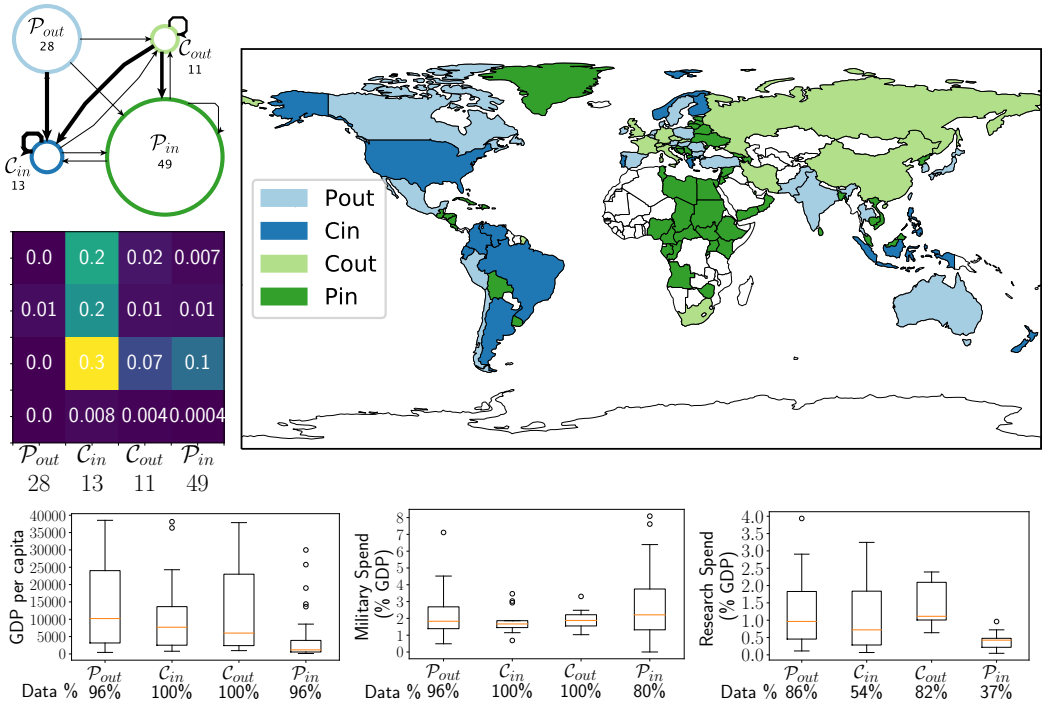


Figure 5. Structures in the *WorldTrade* data set. We show summary network diagrams associated with the uncovered structures for the MaxLike partition on *Trade* network, constructed using trades from the category *"Armoured fighting vehicles, war firearms, ammunition, parts"* category from the year 2000. In the **top left panel**, we show a summary of the uncovered structure. The size of each of the nodes is proportional to the number of nodes in each set, and the width of the lines is given by the percentage of edges that are present between the sets. In the **middle left panel**, we display the percentage of edges between each pair of blocks, allowing for a visualisation of the 'L'-structure. The **top right panel** visualises the partition on a World map with the colours corresponding to each of the uncovered sets. In the lower set of three panels, we display boxplots of three covariates of the uncovered groups, namely *GDP per capita*, *military spend* as a percentage of GDP, and *research spend* as a percentage of GDP. To render the covariates comparable with the partitions from the year 2000, we restrict the covariate data to be from the same year. We note that data from yer 2000 is not available for all country covariate pairs, and thus we present the percentage of countries with data in each group in the bottom row of each plot.

In contrast, the set $\mathcal{C}_{in}$ has a higher median GDP per capita but with a lower upper quartile, and on average, lower research spend than $\mathcal{C}_{out}$ (Fig. 5). It includes several South American countries, (Argentina, Brazil, Colombia, Ecuador and Venezuela), several European countries (Greece, Norway and Finland), and several countries in south-east Asia/Oceania (Philippines, Indonesia and New Zealand). A key player in the network appears to be the USA, with a very high in-degree of 45 (the country with the second-largest in-degree is Norway, also in $\mathcal{C}_{in}$, with an in-degree of 15) and a lower out-degree of 14 (11 of which are in $\mathcal{C}_{in}$); the country with the largest out-degree of 16 is the Czech Republic (6 of which are in $\mathcal{C}_{in}$). To assess the robustness of this allocation, we removed the USA and all its degree 1 neighbours (a total of four nodes); the resulting core–periphery structure is similar with 9 nodes changing sets.

The set $\mathcal{P}_{out}$ appears to consist of economies which are not large exporters, but support the countries in $\mathcal{C}_{in}$. The group consists of 14 European nations (e.g. Austria, Belgium, Netherlands and Spain), several nations from Asia, (India, Pakistan, Japan, South Korea, Singapore, Taiwan and Thailand), three Latin American countries (Chile, Mexico and Peru) and several additional countries which do not fit into a clear division. Finally, $\mathcal{P}_{in}$ consists of nations who buy from the main exporters, but do not export themselves. This group is large (49 nodes), and includes

17 African nations, representing most of the African nations in the data set. An additional set of seven nations were either part or closely aligned with the USSR (e.g. Estonia, Latvia, Lithuania and Ukraine). Finally, there is also a group of six Latin American countries and seven Middle Eastern countries, including Syria and Oman. The set $\mathcal{P}_{in}$ appears to have on average lower GDP per capita than other groups (Fig. 5), with a higher range of military spending as a proportion of GDP. For this group, data on the research spend as a percentage of GDP is only available for 37% of the countries. We observe that for these countries, it is (on average) much lower than the other groups.

In conclusion, our procedure uncovers four groups, each with a different structural role in the trade network. We have explored the roles that each of these groups might play in the global market, and while we cannot rule out data quality issues, the partition found does uncover latent strong patterns which we have validated by considering external covariates.

6. Conclusion and future work

We provide the first comprehensive treatment of a directed discrete core–periphery structure which is not a simple extension of the bow-tie structure. The structure we introduced consists of two core sets and two periphery sets defined in an edge-direction-dependent way, each with a unique connection profile.

In order to identify when this structure is statistically significant in real-world networks, and to rank partitions uncovered by different methods in a systematic manner, we introduce two quality measures: p-values from Monte Carlo tests and a modularity-like measure which we call $DCPM$. We validate both measures on synthetic benchmarks where ground truth is available.

To detect this structure algorithmically, we propose three methods, HITS, ADVHITS and MAXLIKE, each with a different trade-off between accuracy and scalability, and find that MAXLIKE tends to outperform ADVHITS, as well as the standard methods from the literature against which we compared.

Using our quality functions to select and prioritise partitions, we explore the existence of our directed core–periphery structure in three real-world data sets, namely a faculty hiring network, a world trade network, and a political blog network. In each data set, we found at least one significant structure when comparing to random ER and configuration model graphs.

(i) In the faculty hiring data set, the MAXLIKE partition uncovers a new structure, namely Canadian universities which have a large number of links with the top US schools, but also appear to strongly recruit from their own schools, indicating a complementary structure to the one found in [46].

(ii) In the trade data, we uncover four sets of countries that play a structurally different role in the global arms trade, and we validate this structure using covariate data from the world bank.

(iii) In the political blogs data set, we uncover a $\mathcal{C}_{in}$ core, which we hypothesise to consist of authorities that are highly referenced, and a $\mathcal{C}_{out}$ core which links to a large number of other blogs. We support this hypothesis by noting that $\mathcal{C}_{in}$ has a much lower percentage of 'blogspot' sites than the other set, and that $\mathcal{C}_{in}$ contains all but 1 of the top blogs identified by [48].

In cases where one of our methods does not yield a statistically significant partition or yields a partition with a low value of DCPM (e.g., HITS with *Trade*), it can be important to inspect the output partition before disregarding it. We have observed that in certain cases this can occur because the assignment of clusters to the sets $\mathcal{P}_{out}, \mathcal{C}_{in}, \mathcal{C}_{out}$ and $\mathcal{P}_{in}$ with the highest likelihood in the final step of each method (see Section 3) has low density within the 'L' and high density outside of the 'L'. This may phenomenon occur because the stochastic blockmodel which assigns the group labels of recovered sets rewards homogeneity but does not penalise for sparseness within the 'L'. One could modify our implementation into a constrained likelihood optimisation where one would obtain partitions with potentially lower likelihood but a more pronounced 'L' structure.

Future research directions There are a number of interesting directions to explore in future work. We start with the specification of the core–periphery structure. The faculty data highlights that some nodes simply may not fit the core–periphery pattern, and thus following the formulation of bow-tie, it would be interesting to explore modifications to our approaches that would allow for not placing nodes if they do not match the pattern (for example, by introducing a separate set for outlier nodes). As detailed in SI A, other directed core–periphery patterns are possible. Some of our methods could be adapted to detect such core–periphery patterns. In principle, all possible core–periphery structures could be tested simultaneously, with an appropriate correction for multiple testing. Such a development should of course be motivated by a suitable data set which allows for interpretation of the results. More generally, meso-scale structures may change over time, and it would be fruitful to extend our structure and methods to include time series of networks.

Next, we propose some future directions regarding the methods for detecting core–periphery structure. The first direction concerns scalability. Depending on the size of the data set under investigation, a user of our methods may wish to compromise accuracy for scalability (e.g., by using HITS or ADVHITS instead of MAXLIKE). Another scalable method to potentially consider stems from the observation that the expected adjacency matrix (under a suitable directed stochastic block model) is a low-rank matrix. With this in mind, the observed adjacency matrix can be construed as a low-rank perturbation of a random matrix, and therefore, one could leverage the top singular vectors of the adjacency matrix to propose an algorithm for directed core-periphery detection. The advantage of this approach is that it is amenable to a theoretical analysis and one could provide guarantees on the recovered solution, by using tools from matrix perturbation and random matrix theory. In our preliminary numerical experiments, such an SVD-based approach outperforms the standard methods, and while outperformed by MAXLIKE and ADVHITS, it is considerable faster. More details can be found in the technical report [23]. Further future work could explore graph regularisation techniques, which may increase performance for sparse networks. Another direction for future work concerns $DCPM$. In this paper, we have used it as a quality function that is method-independent for assessing the directed core–periphery partition in Eq. (2.2) produced by different methods. It would be interesting to develop methods which optimize the $DCPM$ quality function directly.

Finally, in future work, it would be interesting to explore more data sets with complex structure. In studies of meso-scale structure (e.g. core–periphery and community structure), there are many possible methods for detecting a given partition structure. While our methods are designed to detect a specific core–periphery structure, empirical networks often contain more than one type of meso-scale structure at a time. Adapting our partition selection process to other types of meso-scale structures and combining different methods to explore a range of meso-scale structures may yield novel insights about empirical networks.

Acknowledgements. We thank Aaron Clauset for useful discussions and the authors of [26] for providing the code for the bow-tie structure. We also thank the anonymous referees and the board member for helpful suggestions which have much improved the paper.

Funding. This work was funded by EPSRC grant EP/N510129/1 at The Alan Turing Institute and Accenture Plc. In addition, we acknowledge support from COST Action CA15109.

References

1. Newman M. 2018 *Networks*. Oxford University Press 2nd edition.
2. Peixoto TP. 2014 Hierarchical Block Structures and High-Resolution Model Selection in Large Networks. *Phys. Rev. X* **4**, 011047.
3. Beguerisse-Díaz M, Garduno-Hernández G, Vangelov B, Yaliraki SN, Barahona M. 2014 Interest communities and flow roles in directed networks: the Twitter network of the UK riots. *J. Royal Soc. Interface* **11**, 20140940.
4. Kojaku S, Masuda N. 2017 Finding multiple core-periphery pairs in networks. *Phys. Rev. E* **96**, 052313.

5. Borgatti SP, Everett MG. 1999 Models of core/periphery structures. *Soc. Netw.* **21**, 375–395.
6. Everett MG, Borgatti SP. 2000 Peripheries of cohesive subsets. *Soc. Netw.* **21**, 397 – 407.
7. Holme P. 2005 Core-periphery organization of complex networks. *Phys. Rev. E* **72**, 046111.
8. Yang J, Leskovec J. 2012 Structure and overlaps of communities in networks. *arXiv preprint arXiv:1205.6228*.
9. Zhang X, Martin T, Newman ME. 2015 Identification of core-periphery structure in networks. *Phys. Rev. E* **91**, 032803.
10. Tudisco F, Higham DJ. 2019 A nonlinear spectral method for core–periphery detection in networks. *SIMODS* **1**, 269–292.
11. Mondragón RJ. 2016 Network partition via a bound of the spectral radius. *J. Complex Netw.* **5**, 513–526.
12. Cucuringu M, Rombach P, Lee SH, Porter MA. 2016 Detection of core–periphery structure in networks using spectral methods and geodesic paths. *Eur. J. Appl. Math.* **27**, 846–887.
13. Lee SH, Cucuringu M, Porter MA. 2014 Density-Based and Transport-Based Core-Periphery Structures in Networks. *Phys. Rev. E* **89**.
14. Tang W, Zhao L, Liu W, Liu Y, Yan B. 2019 Recent advance on detecting core-periphery structure: a survey. *CCF Transactions on Pervasive Computing and Interaction* pp. 1–15.
15. Azimi-Tafreshi N, Dorogovtsev SN, Mendes JFF. 2013 Core organization of directed complex networks. *Phys. Rev. E* **87**, 032815.
16. van Lidth de Jeude J, Caldarelli G, Squartini T. 2019 Detecting core-periphery structures by surprise. *EPL (Europhysics Letters)* **125**, 68001.
17. Boyd JP, Fitzgerald WJ, Mahutga MC, Smith DA. 2010 Computing continuous core/periphery structures for social relations data with MINRES/SVD. *Soc. Netw.* **32**, 125 – 137.
18. Kostoska O, Mitikj S, Jovanovski P, Kocarev L. 2020 Core-periphery structure in sectoral international trade networks: A new approach to an old theory. *PLOS ONE* **15**, 1–24.
19. Csermely P, London A, Wu LY, Uzzi B. 2013 Structure and dynamics of core/periphery networks.. *J. Complex Netw.* **1**, 93–123.
20. Broder A, Kumar R, Maghoul F, Raghavan P, Rajagopalan S, Stata R, Tomkins A, Wiener J. 2000 Graph structure in the Web. *Comput. Netw.* pp. 309–320.
21. Lu NP. 2016 Using eigenvectors of perturbed and collapsed adjacency matrices to explore bowtie structures in directed networks. *J Chin Inst Eng* **39**, 936–945.
22. Yang R, Zhuhadar L, Nasraoui O. 2011 Bow-tie decomposition in directed graphs. In *14th International Conference on Information Fusion* pp. 1–5.
23. Elliott A, Chiu A, Bazzi M, Reinert G, Cucuringu M. 2019 Core-Periphery Structure in Directed Networks. .
24. Cattani G, Ferriani S. 2008 A Core/Periphery Perspective on Individual Creative Performance: Social Networks and Cinematic Achievements in the Hollywood Film Industry. *Organization Science* **19**, 824–844.
25. Kleinberg JM. 1999 Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)* **46**, 604–632.
26. Lacasa L, van Lidth de Jeude J, Di Clemente R, Caldarelli G, Saracco F, Squartini T. 2019 Reconstructing Mesoscale Network Structures. *Complexity* **2019**, 5120581.
27. Ma HW, Zeng AP. 2003 The connectivity structure, giant strong component and centrality of metabolic networks. *Bioinformatics* **19**, 1423–1430.
28. Barucca P, Lillo F. 2016 Disentangling bipartite and core-periphery structure in financial networks. *Chaos Soliton Fract* **88**, 244 – 253.
29. Kojaku S, Masuda N. 2018 Core-periphery structure requires something else in the network. *New Journal of Physics* **20**, 043012.
30. Hubert L, Arabie P. 1985 Comparing partitions. *J Classif* **2**, 193–218.
31. Pedregosa, F. et al. 2011 Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830.
32. Meilă M, Shi J. 2001 A Random Walks View of Spectral Segmentation. In *International Workshop on Artificial Intelligence and Statistics (AISTATS)*.
33. Kvalseth TO. 1987 Entropy and correlation: Some comments. *IEEE Transactions on Systems, Man, and Cybernetics* **17**, 517–519.
34. Satuluri V, Parthasarathy S. 2011 Symmetrizations for clustering directed graphs. In *Proc. of the 14th International Conference on Extending Database Technology* pp. 343–354. ACM.
35. Borodin A, Robert GO, Rosenthal JS, Tsaparas P. 2005 Link Analysis Ranking: Algorithms, Theory, and Experiments. *ACM Transactions on Internet Technology*.

36. Hagberg AA, Schult DA, Swart PJ. 2008 Exploring network structure, dynamics, and function using NetworkX. In *Proc. of the 7th Python in Science Conference (SciPy2008)* pp. 11–15 CA USA.
37. Kleinberg JM. 1999 Authoritative Sources in a Hyperlinked Environment. *J. ACM* **46**, 604–632.
38. Cucuringu M, Koutis I, Chawla S, Miller GL, Peng R. 2016 Simple and Scalable Constrained Clustering: a Generalized Spectral Method. *AISTATS 2016* pp. 445–454.
39. Lee JR, Gharan SO, Trevisan L. 2014 Multiway spectral partitioning and higher-order Cheeger inequalities. *J. ACM* **61**, 37.
40. Arthur D, Vassilvitskii S. 2007 k-means++: the advantages of careful seeding. In *SODA '07: Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* pp. 1027–1035. Society for Industrial and Applied Mathematics.
41. MacQueen JB. 1967 Some Methods for Classification and Analysis of MultiVariate Observations. In Cam LML, Neyman J, editors, *Proceedings of the fifth Berkeley Symposium on Mathematical Statistics and Probability* vol. 1 pp. 281–297. University of California Press.
42. Karrer B, Newman ME. 2011 Stochastic blockmodels and community structure in networks. *Phys. Rev. E* **83**, 016107.
43. Rohe K, Qin T, Yu B. 2016 Co-clustering directed graphs to discover asymmetries and directional communities. *Proc. Natl. Acad. Sci. U.S.A.* **113**, 12679–12684.
44. Peixoto TP. 2014a The graph-tool python library. *Figshare*.
45. Peixoto TP. 2014b Efficient Monte Carlo and greedy heuristic for the inference of stochastic block models. *Phys. Rev. E* **89**, 012804.
46. Clauset A, Arbesman S, Larremore DB. 2015 Systematic inequality and hierarchy in faculty hiring networks. *Sci. Adv.* **1**, e1400005.
47. Feenstra RC, Lipsey RE, Deng H, Ma AC, Mo H. 2005 World Trade Flows: 1962-2000. Working Paper 11040 National Bureau of Economic Research.
48. Adamic LA, Glance N. 2005 The political blogosphere and the 2004 US election: divided they blog. In *Proc. of the 3rd international workshop on Link discovery* pp. 36–43. ACM.
49. Jaschik S. 2012 Faculty Pay, Around the World. Inside Higher Ed https://www.insidehighered.com/news/2012/03/22/new-study-analyzes-how-faculty-pay-compares-worldwide.
50. Sherouse O. 2014 Wbdata - Python Package. Github - https://github.com/skojaku/core-periphery-detection.
51. World Bank, OECD. 2020 GDP per capita (current US$) data. World Bank - https://data.worldbank.org/indicator/NY.GDP.PCAP.CD.
52. Stockholm International Peace Research Institute. 2020 Military expenditure (% of GDP). World Bank and SIPRI- https://data.worldbank.org/indicator/MS.MIL.XPND.GD.ZS/https://www.sipri.org/databases.
53. UNESCO Institute for Statistics . 2020 Research and development expenditure (% of GDP). World Bank and UNESCO - https://data.worldbank.org/indicator/GB.XPD.RSDV.GD.ZS.