Journal of Complex Networks (2021) **00**, 1–21
doi: 10.1093/comnet/cnaa047

Detecting the number of clusters in a network

GABRIEL BUDEL[†]
Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands
[†]Corresponding author. Email: G.J.A.Budel@tudelft.nl

AND

PIET VAN MIEGHEM
Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, Delft, The Netherlands

Edited by: Ernesto Estrada

[Received on 28 September 2020; editorial decision on 2 December 2020; accepted on 3 December 2020]

Many clustering algorithms for complex networks depend on the choice for the number of clusters and it is often unclear how to make this choice. The number of eigenvalues located outside a circle in the spectrum of the non-backtracking matrix was conjectured to be an estimator of the number of clusters in a graph. We compare the estimate of the number of clusters obtained from the spectrum of the non-backtracking matrix with three estimators based on the concept of modularity and evaluate the methods on several benchmark graphs. We find that the non-backtracking method detects the number of clusters better than the modularity-based methods for the graphs in our simulation study, especially when the clusters have slightly different sizes. The estimates of the non-backtracking method are narrowly distributed around the true number of clusters for all benchmark graphs considered. Additionally, for graphs without a clustering structure, the non-backtracking method detects exactly one cluster, which is a convenient property of an estimator of the number of clusters. However, the lack of a well-defined concept of a cluster prevents sharp conclusions.

Keywords: Complex Networks, Community Detection, Spectral Clustering, Number of Clusters, Non-backtracking Matrix.

1. Introduction

The detection of community structures in complex networks has been a popular topic in network science for many years [1]. Finding the number of communities or the number of clusters, however, receives comparably little attention. Many community detection algorithms require the number of communities as an input and their results depend on the chosen number of communities. The number of communities found by a given algorithm depends on the definition of ‘community’. The precise definition of a community is in turn driven by the motivation behind employing community detection: different motivations lead to different definitions, none of them is the best, but each of them is potentially useful for a specific goal [2]. Here, we adopt the clustering approach of finding groups of nodes that are ‘similar’ or ‘close’. In the context of complex networks, similarity or closeness between nodes is often described by the number of links or the weights on the links in weighted networks. A cluster is then a group of nodes that is densely

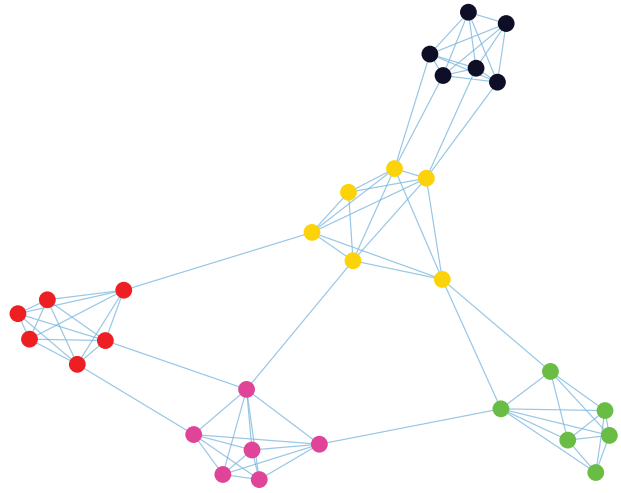


FIG. 1. A network with five densely connected clusters of nodes.

connected internally, while sparsely connected to the nodes of other groups. Figure 1 exemplifies the definition.

Initially, the research on clustering in complex networks focused on modularity optimization, initiated by Newman and Girvan [3–5]. The concept of modularity is naturally linked to the definition of the clustering problem: nodes that share more links than expected are considered to be part of one group, a cluster. While exact optimization of modularity is computationally intractable [6], many approximate algorithms were shown to achieve high modularity for some well-known real-world networks. Many heuristic modularity maximization algorithms do not require the number of clusters as prior knowledge; they discover the number of clusters in the network during the optimization.

Another popular and accurate type of clustering is spectral clustering. In spectral clustering nodes are assigned to clusters based on the values of their corresponding components in a subset of the eigenvectors of a matrix representation of the network, such as the Laplacian. However, most spectral clustering methods require the knowledge of the number of clusters. In real-world networks, the actual number of clusters is usually unknown and traditionally researchers therefore resorted to guessing or trying a range of different values. Recently, a spectral clustering algorithm based on the non-backtracking matrix was shown to achieve optimal clustering results for graphs generated by the stochastic block model [7]. Additionally, Krzakala *et al.* [7] conjectured that the number of real eigenvalues of the non-backtracking matrix H that are separated from the bulk of the eigenvalues is an estimate of the number of clusters in the network. Our contribution consists of the assessment of the accuracy of the non-backtracking method in comparison with three modularity-based methods for estimating the number of clusters in a network. We find that the non-backtracking outperforms the three other methods for all benchmark graphs considered in this work.

In Section 2, we give a short description of several clustering techniques for complex networks, focusing on modularity optimization methods and spectral clustering. We describe the literature related to this work in Section 3. In Section 4, we describe the non-backtracking method and three modularity-based methods for detecting the number of clusters in a network. We compare the four detection methods

on benchmark graphs in several simulation experiments and describe the experimental results in Section 5. We conclude our research and reflect upon our findings in Section 6.

2. Clustering in complex networks

2.1 *Modularity maximization*

The modularity of a graph is a measure for the quality of a given partition of a network based on the number of links between nodes belonging to the same cluster [6]. The modularity measure m proposed by Newman and Girvan [5] is defined as the difference between the actual number of intra-cluster links and the expected number of intra-cluster links if links were to be placed at random. The expected number of links between node i and node j if links are placed randomly is $\frac{d_i d_j}{2L}$. The modularity m is calculated as

$$m = \sum_{i=1}^{N} \sum_{j=1}^{N} \left(a_{ij} - \frac{d_i d_j}{2L} \right) 1_{\{i \text{ and } j \text{ belong to the same cluster}\}}, \tag{1}$$

where a_{ij} is an element of the $N \times N$ adjacency matrix A of a network with N nodes and L links, d_i is the degree of node i and $1_{\{x\}}$ is an indicator function that equals 1 if statement $\{x\}$ is true and equals 0 if it is false. A modularity close to 1 indicates a strong modular structure, while a modularity of 0 indicates that the partition is not better than random. Maximizing the modularity for a number of clusters c larger than two is equivalent to the maximum cut problem, which is NP-hard [3]. However, Van Mieghem *et al.* [8] find that it is possible to derive an upper bound on the modularity measure m for any graph given the true clustering:

$$m \leq 1 - \frac{1}{c} - \frac{L_{\text{inter}}}{L}, \tag{2}$$

with c the true number of clusters in the graph and L_{inter} the total number of inter-cluster links in the true clustering of the network. Additionally, the $N \times N$ modularity matrix M of a network is defined with elements $m_{ij} = a_{ij} - \frac{d_i d_j}{2L}$. If we define the $N \times c$ community matrix S with elements S_{ik} to be equal to 1 if node i is in cluster k and 0 otherwise, we can express the modularity [6] in terms of these matrices:

$$m = \frac{1}{2L} \sum_{k=1}^{c} \sum_{i=1}^{N} \sum_{j=1}^{N} S_{ik} m_{ij} S_{jk} = \frac{\text{trace}\left(S^T M S\right)}{2L}. \tag{3}$$

Several heuristic algorithms that approximately maximize modularity have been proposed. We consider the popular Louvain method [9] and Newman's iterative bisection algorithm [4], because they have been shown to achieve high modularity for several real-world networks.

2.2 *Spectral clustering*

In spectral clustering, nodes are assigned to clusters based on the values of their corresponding components in one or more of the eigenvectors of a matrix representation of the network. The Laplacian is the most popular matrix representation for spectral clustering [1]. Fiedler showed that the eigenvector corresponding to the second smallest eigenvalue μ_{N-1} of the Laplacian Q can be used to obtain a

bipartition of a graph into two equivalent parts [6]. Here, *equivalent* means equivalent with respect to the second smallest eigenvalue's eigenvector, the Fiedler eigenvector. If a disconnected network consists of c connected components, then the Laplacian Q will have c eigenvalues that are equal to zero. The eigenvectors corresponding to these eigenvalues are the trivial all-ones eigenvectors of the connected components, with entry 1 for nodes that are part of the corresponding component and entry 0 for nodes that are not part of the component. The c eigenvectors map the nodes of one connected component onto a single point on one of the axes in a c-dimensional space. The c eigenvectors can then be used to detect the component membership of the nodes. The idea of spectral clustering with the Laplacian Q is that if the graph consists of c weakly linked subgraphs (e.g. a network with community structure), the smallest $c-1$ non-zero eigenvalues will still be relatively close to zero. An eigenvector belonging to one of the c smallest eigenvalues no longer maps the nodes of one subgraph onto a single point on one of the axes, but rather to a small cloud of points that are still relatively close to each other [1]. Any clustering algorithm that can identify clusters shaped as clouds of points in metric space given the true number of clusters c, such as the *k-means* algorithm, can detect the original cluster memberships in the network. To what extent the clusters are considered weakly linked and to what extent spectral clustering works, we will discuss in the context of detecting the number of clusters with the maximum eigengap property in Section 2.3. For matrix representations other than the Laplacian Q, spectral clustering also works [1].

2.3 *Maximum Eigengap*

Consider the $N \times N$ adjacency matrix A of a graph G with c equally sized clusters of $N_g = N/c$ nodes. The nodes can always be rearranged such that the nodes are ordered according to the cluster memberships. The $N_g \times N_g$ adjacency matrices A_g of the cluster subgraphs are then located on the diagonal of the adjacency matrix A and contain only intra-cluster links:

$$A = \begin{bmatrix} A_1 & B_{12} & \dots & B_{1c} \\ B_{12}^T & A_2 & \dots & B_{2c} \\ \vdots & \vdots & \ddots & \vdots \\ B_{1c}^T & B_{2c}^T & \dots & A_c \end{bmatrix}. \tag{4}$$

We first consider the case where the c clusters are disconnected and the adjacency matrix A in (4) is block diagonal with all blocks $B_{ij} = O$. The eigenvalues of the matrix A are the union of the sets of eigenvalues of the blocks A_g and each block A_g has N_g eigenvalues $\lambda_1(A_g) \geq \lambda_2(A_g) \geq \cdots \geq \lambda_{N_g}(A_g)$. Let the degree distribution of the subgraph g have expectation $E\left[D_g\right]$ and variance $\mathrm{Var}\left[D_g\right]$. The Perron–Frobenius eigenvalue $\lambda_1(A_g)$ of each subgraph g must satisfy:

$$\lambda_1(A_g) \geq E\left[D_g\right]\sqrt{1 + \frac{\mathrm{Var}\left[D_g\right]}{(E\left[D_g\right])^2}}, \tag{5}$$

as derived in [6]. If the subgraph g has no community structure itself, then the largest eigenvalue $\lambda_1(A_g)$ is substantially larger than the other $N_g - 1$ eigenvalues [10, 11]. An exception to this statement would be when the expected degree $E\left[D_g\right]$ of nodes in the subgraph g is relatively low. However, in general, we have $\lambda_1(A_g) \gg \lambda_2(A_g)$ for subgraphs without a community structure [11]. If indeed $\lambda_1(A_g) \gg \lambda_2(A_g)$ and the degree distributions of the subgraphs g are similar, then the c largest eigenvalues of A are substantially larger than the other $N - c$ eigenvalues and we observe a large gap. Finding the largest gap among the sorted eigenvalues of A, the maximum eigengap, identifies the number of clusters c in this case.

Adding the inter-community links to the graph G can be considered as a perturbation of the adjacency matrix A, such that $A(\zeta) = A + \zeta B$ is the adjacency matrix with inter-community links for a value ζ that is small enough [11, 12]. The symmetric matrix ζB contains only the inter-community links in the blocks B_{ij} from (4) and has zeros O on the diagonal. When the perturbed eigenvalues $\lambda(\zeta)$ of $A(\zeta)$ are close to the eigenvalues λ of A, the maximum eigengap is still a good estimator of the number of clusters c. Similarly, when the perturbed eigenvectors $x(\zeta)$ are close to the eigenvectors x of A, spectral clustering with these eigenvectors still works. If ζ is sufficiently small, then the ith perturbed eigenvalue $\lambda_i(\zeta)$ can be approximated by the first-order approximation of $\lambda_i(\zeta)$,

$$\lambda_i(\zeta) \approx \lambda_i + \zeta x_i^T B x_i, \tag{6}$$

and also for the corresponding eigenvector $x_i(\zeta)$,

$$x_i(\zeta) \approx x_i + \zeta \sum_{j \neq i}^{N} \frac{x_j^T B x_i}{\lambda_i - \lambda_j} x_j, \tag{7}$$

as derived in [13]. In the clustering problem, we assume $\lambda_i \gg \lambda_j$ for the c largest eigenvalues $\lambda_1(A_g)$ of A. Wu *et al.* [12] therefore propose to write the approximations for the eigenvectors $x_i(\zeta)$ of the c largest eigenvalues as

$$x_i(\zeta) \approx x_i + \zeta \sum_{j \neq i}^{c} \frac{x_j^T B x_i}{\lambda_i - \lambda_j} x_j + \frac{\zeta}{\lambda_i} B x_i, \tag{8}$$

where the last term of the approximation error is smaller for large λ_i. Denote by $X_{\setminus\{i\}}$ the $N \times (N-1)$ matrix of all eigenvectors x_j except x_i. The approximations are close to the actual $\lambda_i(\zeta)$ and $x_i(\zeta)$ when

$$|\lambda_i - \lambda_{i+1}| - |\zeta| \left\| x_i^T B x_i \right\|_2 - |\zeta| \left\| X_{\setminus\{i\}}^T B X_{\setminus\{i\}} \right\|_2 > 0, \tag{9}$$

and

$$|\zeta| \left\| X_{\setminus\{i\}}^T B x_i \right\|_2 < \frac{1}{2} \left(|\lambda_i - \lambda_{i+1}| - |\zeta| \left\| x_i^T B x_i \right\|_2 - |\zeta| \left\| X_{\setminus\{i\}}^T B X_{\setminus\{i\}} \right\|_2 \right), \tag{10}$$

as described in [12, 14].

2.4 *Stochastic block model*

The stochastic block model (SBM) is a generative random graph model in which the link probabilities depend on cluster memberships. A realization of the model therefore results in a graph with a community structure. Clustering algorithms are often evaluated on SBM graphs. The hardest case for cluster detection is the symmetric SBM (SSBM), where there are only two different link probabilities: p_{in} for two nodes that belong to the same cluster, p_{out} for two nodes that belong to different clusters. When the intra-cluster link probability is larger than the inter-cluster probability, $p_{\text{in}} > p_{\text{out}}$, we obtain graphs that look similar to the graph in Fig. 1. In SSBM graphs with clusters of equal size, the clusters are not detectable based on the node degrees alone, because the expected degree is the same for each node in the graph, irrespective of

its cluster membership. We consider sparse, assortative SSBMs with c clusters of equal size. The SSBM is said to be sparse and assortative if $p_{\text{in}} = b_{\text{in}}/N$ and $p_{\text{out}} = b_{\text{out}}/N$ for two constants $b_{\text{in}} > b_{\text{out}} > 0$ that do not depend on the number of nodes N. In case of equally sized clusters, the expected average degree $E[D]$ is a weighted average of the constants:

$$E[D] = \frac{b_{\text{in}} + (c-1)b_{\text{out}}}{c}. \tag{11}$$

Decelle *et al.* [15] show that there is a regime where no algorithm can detect the clusters because the block structure is not apparent enough in the limit $N \to \infty$. When the difference $|b_{\text{in}} - b_{\text{out}}|$ is larger than the detectability threshold,

$$|b_{\text{in}} - b_{\text{out}}| > c\sqrt{E[D]}, \tag{12}$$

it is theoretically possible to detect the clusters by some algorithm. However, for most algorithms the difference between b_{in} and b_{out} must much be larger than the detectability threshold $c\sqrt{E[D]}$ to detect the clusters in the sparse case [7]. Decelle *et al.* [15] also show that the detectability limit marks a phase transition from the undetectable state to the theoretically detectable state.

3. Related work

There is not yet a single, generally accepted method that determines the number of clusters in any given network [16]. In absence of a golden standard, there have been a few works that—at least partly—address the problem of detecting the number of clusters. In many of the papers, the clustering methods are evaluated on real-world networks by comparing the results with the presumed underlying clustering or with a benchmark random graph model.

Shen and Cheng [17] discuss the detection of the number of clusters for spectral clustering using the maximum eigengap property based on several different matrix representations of the network and, additionally, based on the covariance matrix of the node degrees. They evaluate the estimators on a random graph model proposed by Lancichinetti *et al.* [18] with clusters of different sizes and heterogeneous node degrees. Shen and Cheng [17] conclude that in graphs with heterogeneous cluster sizes and heterogeneous node distributions the maximum eigengap property of the normalized Laplacian and the covariance matrix estimates the number of clusters best, because the two matrices both correct for heterogeneous node degrees. At the time of their analysis [17], the non-backtracking matrix had not been introduced for spectral clustering yet. Shea and Macker [19] try and formalize the selection of the number of clusters based on the eigengap property of the normalized Laplacian by combining the eigengap property with statistics of random cuts of the graph.

Another approach to the detection of the number of clusters consists of fitting a statistical parametric model to the graph and including the number of clusters as one of the parameters to be estimated. The stochastic block model is often explicitly assumed as the true underlying model of the graph and then the most likely number of clusters is estimated using maximum likelihood [16, 20, 21]. Given that the true clustering is unobserved, multiple values for the number of clusters are evaluated based on some criterion, before the estimated number of clusters is found. Alternatively, the problem of finding the number of clusters can also be cast in a Bayesian framework [22, 23], in which one formulates a prior distribution on both the number of clusters and the cluster memberships and then finds the *a posteriori* most likely number of clusters after learning from some network data. The disadvantage of Bayesian approaches is the

computational burden of the required data sampling, which Decelle *et al.* [15] overcome by proposing a belief propagation algorithm that runs in polynomial time. They argue that their method is asymptotically exact for the stochastic block model, also in the sparse case.

Krzakala *et al.* [7] introduced the non-backtracking matrix of Hashimoto [24] in a spectral clustering algorithm for networks. They conjectured that a spectral clustering algorithm based on the non-backtracking matrix can detect clusters all the way down to the detectability limit of the stochastic block model in the sparse case. Krzakala *et al.* [7] remark that the number of eigenvalues located outside a circle around the origin in the complex plane seems to be a good estimator of the number of clusters in the graph. Given that the bulk of the eigenvalues of the non-backtracking matrix are contained within a circle around the origin of the complex plane, the number of eigenvalues outside that circle yields a clear and unambiguous decision rule for the choice of the number of clusters.

4. Detecting the number of clusters

We compare four different methods to detect the number of clusters in a network. We compare the estimate obtained by counting the eigenvalues outside the circle in the spectrum of the non-backtracking matrix H with three different methods based on the concept of modularity.

4.1 *Detection methods*

4.1.1 *Spectrum of the non-backtracking matrix H* The non-backtracking matrix H is based on the idea of a non-backtracking walk: a walk that does not turn around and goes back to its starting point immediately after the first step. The non-backtracking matrix H of an undirected graph G is a $2L \times 2L$ matrix with elements $h_{el} = 1$ if the pair of links (e_e, e_l) in G is non-backtracking, otherwise $h_{el} = 0$. Two consecutive links $e_e = i \to j$ and $e_l = j \to k$ are non-backtracking if $i \neq k$. If $i = k$, then the link e_e connects the same two nodes as the link e_l but in opposite direction and the link pair (e_e, e_l) is backtracking. For undirected graphs, each undirected link $\{i, j\}$ is considered twice, once for the direction $i \to j$ and once for the direction $j \to i$, resulting in $2L$ bi-directional links. The non-backtracking matrix H then is a $2L \times 2L$ matrix with elements

$$h_{el} = \begin{cases} 1 & \text{if } e_e = i \to j,\ e_l = j \to k \text{ and } i \neq k \\ 0 & \text{otherwise} \end{cases}, \tag{13}$$

where $e, l \in \{1, \ldots, 2L\}$ the set of all bi-directional links and $i, j, k \in \{1, \ldots, N\}$. A non-backtracking walk is a walk in which every pair of consecutive links is non-backtracking. Similar to the adjacency matrix A, raising the non-backtracking matrix H to the kth power counts the number of non-backtracking walks of length $k + 1$ on the graph. We describe the non-backtracking matrix H in more detail in the Appendix A.

Krzakala *et al.* [7] conjectured that the number of real eigenvalues of the non-backtracking matrix H that are separated from the bulk of the eigenvalues indicates the number of clusters in the network. The bulk of the eigenvalues are located in a circle around the origin of the complex plane with radius the square root of the largest eigenvalue, therefore the number of eigenvalues located outside of the circle indicates the number of clusters. Figure 2 shows the eigenvalues of the matrix H in the complex plane for a graph on $N = 1,000$ nodes generated by an SSBM with three clusters. The largest eigenvalue is approximately equal to the expected degree of 7 and there are exactly three real eigenvalues located outside the circle. The spectrum of the matrix H indeed shows that the number of clusters $c = 3$, in agreement with the conjecture.

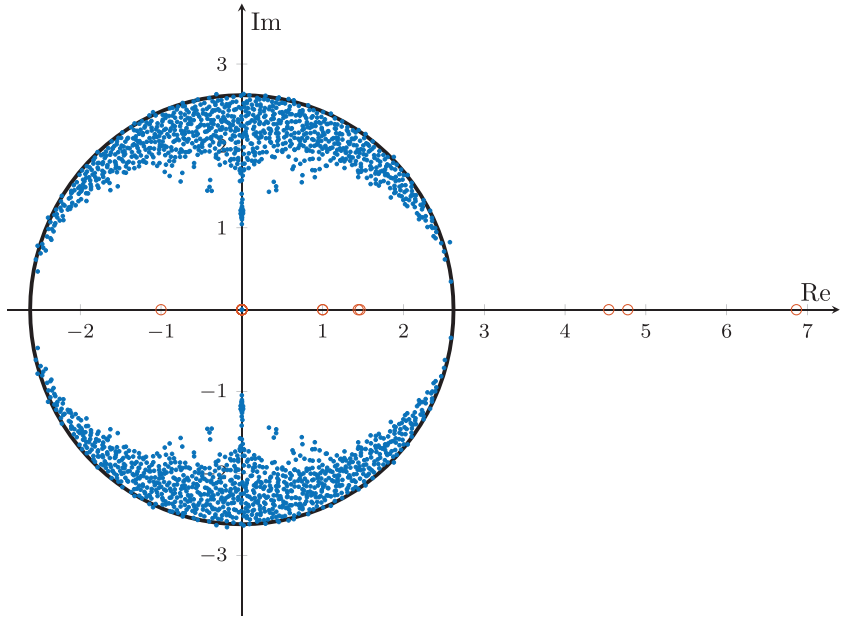


FIG. 2. The spectrum of the non-backtracking matrix in the complex plane for a graph with $N = 1,000$ nodes generated by an SSBM with $c = 3$ clusters of approximately equal size. The real eigenvalues are indicated by small circles and the eigenvalues with a non-zero imaginary part are indicated by dots. The circle containing the bulk of the eigenvalues is centred around the origin and has radius equal to the square root of the largest real eigenvalue. There are three real eigenvalues located outside the circle.

Let $|\lambda_1(H)| \geq \cdots \geq |\lambda_{2L}(H)|$ denote the eigenvalues of the non-backtracking matrix H, sorted in descending order according to the modulus. The eigenvalues $\lambda_l(H) \in \mathbb{C}$ are the solutions to the characteristic polynomial (A.1) in Appendix A.3. The largest eigenvalue $\lambda_1(H)$ is a real, non-negative number [6]. The estimate c^* of the number of clusters is the number of real eigenvalues $\lambda_l(H)$ larger than the square root $\sqrt{\lambda_1(H)}$ of the largest eigenvalue including the largest eigenvalue $\lambda_1(H)$ itself:

$$c^* = \sum_{l=1}^{2L} 1_{\left\{\mathrm{Re}\left(\lambda_l(H)\right) > \sqrt{\lambda_1(H)} \wedge \mathrm{Im}\left(\lambda_l(H)\right) = 0\right\}}. \tag{14}$$

For finding the estimate c^* of the number of clusters, not all $2L$ eigenvalues have to be computed. One first has to find the largest root $\lambda_1(H)$ of the characteristic polynomial and compute its square root $\sqrt{\lambda_1(H)}$. Next, find all real roots larger than $\sqrt{\lambda_1(H)}$ with a numerical procedure. The number of such roots is the estimate c^*. Moreover, it is even more efficient to obtain the estimate c^* by applying the above iterative procedure to the $2N \times 2N$ block matrix H^* specified in (A.2), Appendix A.3. The size of the square non-symmetric matrix H^* is of order $O(N^2)$. We discuss the computational complexity of the non-backtracking method in Appendix A.5.

4.1.2 *Louvain method* The Louvain method is a popular heuristic method aimed at maximizing the modularity of a graph [9]. The method starts out with N clusters, one for every node. The clusters are then iteratively merged in a two-stage procedure such that the modularity m increases in every iteration, until

the modularity cannot be improved upon anymore. However, there is no guarantee that the final result is a global modularity maximum. An estimate of the number of clusters is obtained from the clustering results.

One iteration in the Louvain method consists of two stages. In stage one, each node i is considered sequentially and possibly multiple times. Blondel *et al.* [9] compute the resulting gain in modularity from moving node i to the cluster g of some neighbouring node j as:

$$\Delta m = \left[\frac{\sum_{\text{in}} + 2\sum_{l:S_{lg}=1} W_{il}}{2L} - \left(\frac{\sum_{\text{tot}} + d_i}{2L}\right)^2\right] - \left[\frac{\sum_{\text{in}}}{2L} - \left(\frac{\sum_{\text{tot}}}{2L}\right)^2 - \left(\frac{d_i}{2L}\right)^2\right], \tag{15}$$

where W is the weighted adjacency matrix of the graph, $2L = \sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij}$ is the sum of the weights of all links in the graph, $d_i = \sum_{l=1}^{N} W_{il}$ is the sum of the weights of the links incident to node i, $\sum_{\text{in}}$ is the sum of the weights of the intra-cluster links in cluster g, $\sum_{\text{tot}}$ is the sum of the weights of all links incident to one of the nodes in cluster g. Node i is moved to the cluster g of the neighbouring node j for which the modularity gain is most positive. If the resulting gain in modularity is not larger than some small threshold or even negative for all neighbouring nodes j, then the node i remains in its original cluster. In the first iteration, the weighted adjacency matrix W is simply the unweighted adjacency matrix A, from the second iteration onwards, we have the weighted adjacency matrix W constructed in the second stage of the previous iteration. The cluster re-assignment procedure is repeated until there is no node anymore for which there is a positive gain in the modularity m achievable.

In the second stage of an iteration, a new weighted graph is constructed in which each node g represents a cluster g resulting from the first stage. In the new graph, the weight on the link from node g to node h is the sum of the weights of all inter-cluster links between cluster g and cluster h in the graph of Stage 1. The intra-cluster links in the graph of Stage 1 lead to self-loops in the new graph, such that the new graph has the same modularity m as the graph in Stage 1. The newly constructed graph in Stage 2 is the input for Stage 1 of the next iteration. The described iterations are repeated until there is no positive gain in the modularity m achievable anymore.

4.1.3 *Newman's iterated bisection* To maximize the modularity m, Newman [4] proposes to make recursive splits according to the leading eigenvector of the modularity matrix M, which is inspired by the approach of Fiedler [6]. In the case of $c = 2$ clusters, clustering is equivalent to choosing a vector y with elements $+1$ and -1 that indicate the cluster membership. The vector y can be written as a linear combination of the orthogonal eigenvectors $w_1, w_2, \ldots, w_N$ of the modularity matrix M, $y = \sum_{j=1}^{N} \beta_j w_j$, with coefficients $\beta_j = y^T w_j$. Invoking the orthogonality of the eigenvectors, the modularity m is written as:

$$m = \frac{1}{4L} y^T M y = \frac{1}{4L} \sum_{j=1}^{N} \beta_j^2 \lambda_j(M). \tag{16}$$

Maximizing the modularity m is equivalent to choosing the vector y with cluster memberships proportional to the eigenvectors corresponding to a few of the largest eigenvalues. Newman [4] proposes to perform the bisection by maximizing the term for the most positive eigenvalue: $\beta_1 = y^T w_1$. Since the elements of y only take two possible values, the coefficient β_1 is maximized for $y_j = -1$ if $(w_1)_j < 0$ and $y_j = +1$ if $(w_1)_j \geq 0$.

For the case of $c > 2$ clusters, the network is first split in two and then the procedure is repeated on each of the resulting subgraphs separately. However, simply applying the same procedure on the block components of the modularity matrix M corresponding to the subgraphs is not correct, because the block components do not contain inter-cluster links and the modularity m would change if the inter-cluster links are disregarded. Instead, Newman [4] proposes to write the modularity matrix M_g of a cluster g as:

$$M_g = m_{ij} - \left(\sum_{k \in g} m_{ik} \right) \delta_{ij}, \tag{17}$$

with Kronecker delta δ_{ij} equal to 1 if $i = j$ and 0 otherwise. In the iterated bisection algorithm of Newman [4], the subgraphs are iteratively split in two until the modularity cannot be improved anymore, indicated by an absence of positive eigenvalues of the modularity matrix M_g of the subgraph. The stopping criterion is evaluated for each subgraph separately, therefore the resulting number of clusters can be any number greater than or equal to 1 and is not necessarily a multiple of 2. The algorithm provides an estimate of the number of clusters and the corresponding clustering results too.

4.1.4 *Eigengap modularity matrix M* We estimate the number of clusters c by the maximum eigengap of the modularity matrix M as described for the adjacency matrix A in Section 2.3. Let the eigenvalues of the modularity matrix M be sorted in descending order: $\lambda_1(M) \geq \lambda_2(M) \geq \cdots \geq \lambda_N(M)$. The eigenvalues of the modularity matrix M and the adjacency matrix A are interlaced:

$$\lambda_1(A) \geq \lambda_1(M) \geq \lambda_2(A) \geq \lambda_2(M) \geq \cdots \geq \lambda_N(A) \geq \lambda_N(M), \tag{18}$$

as described in [6]. The maximum eigengap property then maximizes the difference $\lambda_{i-1}(M) - \lambda_i(M)$ in the sequence of N eigenvalues as

$$c^* = \arg\max_i \left(\lambda_{i-1}(M) - \lambda_i(M)\right), \qquad i = 2, \ldots, N, \tag{19}$$

with c^* the resulting estimate of the number of clusters c.

5. Results

5.1 *Erdős–Rényi graphs*

We evaluate the cluster detection methods on Erdős–Rényi (ER) random graphs with different link densities in Fig. 3. An ER random graph features no clustering structure in expectation, but in individual realizations some clustering structure might be present due to randomness. The non-backtracking method finds on average one single cluster in the ER graph for each of the link density values. The Louvain method and the modularity eigengap method find a higher, constant number of clusters on average, while Newman's iterated bisection method finds a lower number of clusters as the link density increases. The non-backtracking method detects 1 cluster in absence of a clustering structure, which is a convenient property for an estimator of the number of clusters.

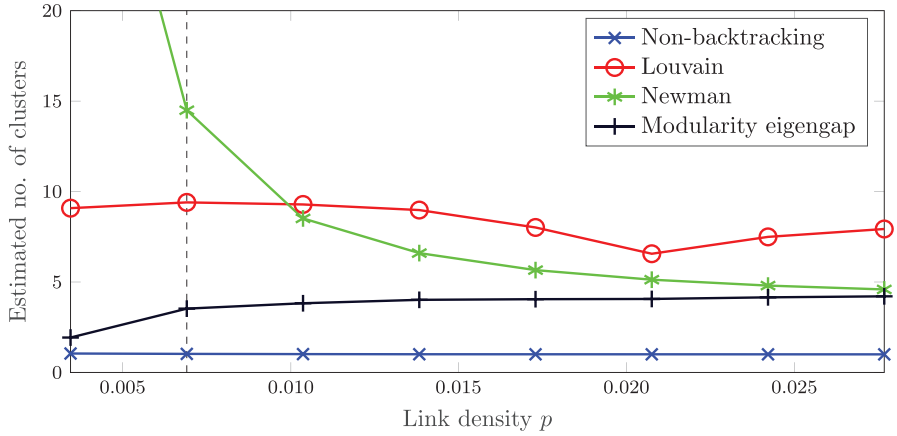


FIG. 3. The estimated number of clusters of the largest connected component for ER graphs $G_p(N)$ with $N = 1,000$ nodes and different link densities p. Each point in the plots represents an average over 10^4 realizations. The vertical dashed line indicates the connectivity threshold p_c for an ER graph with $N = 1,000$ nodes.

5.2 *Ring of cliques*

We evaluate the cluster detection methods on a ring of c cliques, similar to the evaluation method of Fortunato and Barthélemy [25]. Each clique is the complete graph K_{N_g} with N_g nodes and we connect neighbouring cliques by a single link. The set-up is a seemingly easy clustering problem, but Fortunato and Barthélemy [25] illustrate the resolution limit of modularity optimization methods by showing that for a ring of $c > \sqrt{L}$ cliques, the modularity m is maximized for $c/2$ clusters that consist of 2 cliques each. We choose a set-up where each cluster has relatively many nodes and links, therefore we have the opposite: $c < \sqrt{L}$. The complete graph K_{N_g} has the largest possible spectral gap between the two largest eigenvalues.

Figure 4 shows the estimated number of clusters for our experiment. Intuitively, a cluster detection method is expected to find as many clusters as there are cliques. The non-backtracking method estimates the correct number of clusters (cliques) in almost every instance, while Newman's iterated bisection and the modularity eigengap method are close on average. Surprisingly, the Louvain method finds exactly twice the number of cliques in every instance. Inspecting the Louvain clustering results reveals that the method subdivides each clique into two smaller clusters. The clustering results of the Louvain method are therefore almost equivalent to the partition where each clique is a cluster, but the detected number of clusters is not the intuitively expected number.

5.3 *The number of clusters in SSBMs*

5.3.1 *Clustering in the eigenvectors of the modularity matrix M* First, we inspect if and how the clustering pattern appears in the eigenvector components of the modularity matrix for a network generated by an SSBM with $N = 1,000$ nodes and average degree $E[D] = 7$. Figure 5 shows the components of the first eigenvector w_1 versus the components of the second eigenvector w_2 for two graphs generated by SSBMs with $c = 3$ clusters. For the network in Fig. 5a, the difference $b_{\text{in}} - b_{\text{out}} = 19$ is well above the detectability threshold in (12) of 7.94. The objects corresponding to the nodes of a single cluster are clearly cluttered and they are well separated from the objects corresponding to the nodes of other

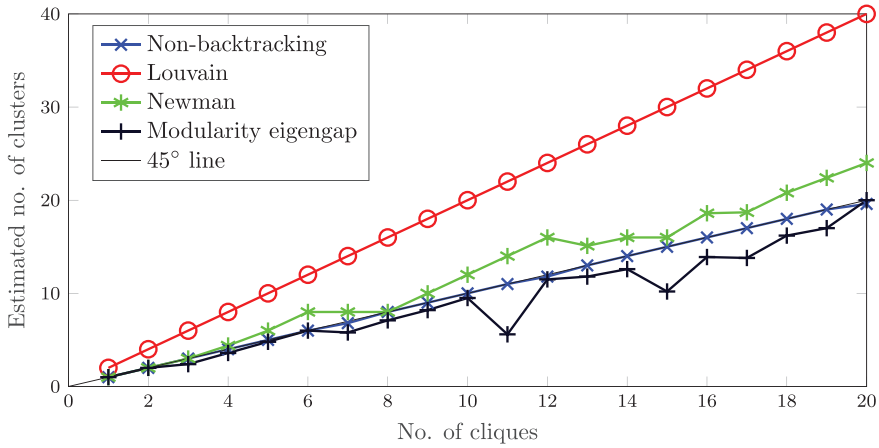
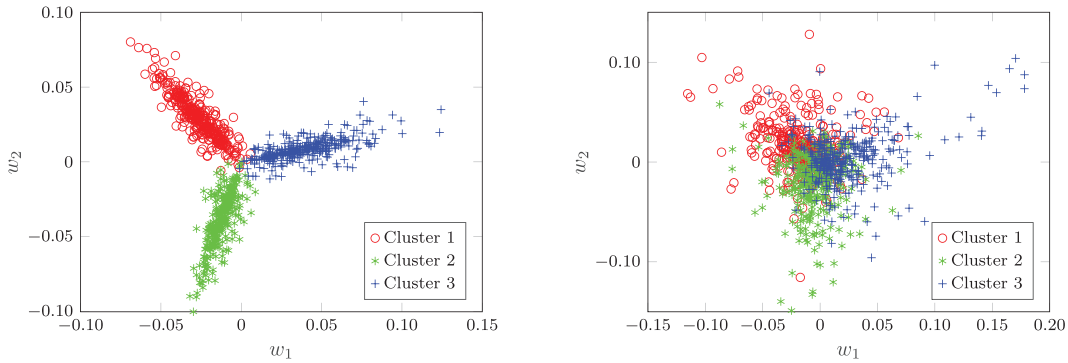


FIG. 4. The estimated number of clusters for a ring of c cliques, where each clique is the complete graph K_{100} with $N_g = 100$ nodes. The entire graph has $N = N_g \cdot c = 100c$ nodes and two neighbouring cliques in the ring are connected by a single link. The 45° line maps the true number of clusters (cliques) to the estimated number of clusters on a 1:1 scale, the estimates of an ideal estimator would be positioned close to the 45° line. For each point in the graph, we randomize the order of the nodes 20 times and average over these 20 estimations.



(a) SSBM with $b_{\text{in}} = 19\frac{2}{3}$, $b_{\text{out}} = \frac{2}{3}$ (b) SSBM with $b_{\text{in}} = 14\frac{1}{3}$, $b_{\text{out}} = 3\frac{1}{3}$

FIG. 5. Scatter plots of the second eigenvector w_2 versus the first eigenvector w_1 of the modularity matrix for two different graphs, both with $N = 1,000$ nodes and generated by SSBMs with $c = 3$ clusters. The parameters b_{in} and b_{out} are chosen such that the expected average degree $E[D] = 7$ in both networks and that they only differ through the difference $b_{\text{in}} - b_{\text{out}}$, for which the theoretical detectability threshold in (12) is 7.94. The colours and shapes of the objects indicate the true cluster memberships.

clusters. For the graph in Fig. 5a, a spectral clustering algorithm based on the first two eigenvectors would successfully detect the cluster memberships of the majority of the nodes. For the graph in Fig. 5b, the difference $b_{\text{in}} - b_{\text{out}} = 11$ is closer to the detectability threshold and the objects corresponding to the nodes of the different clusters now show significant overlap. Even though a spectral clustering algorithm would now classify less nodes correctly, there are still three (overlapping) clouds of points visible. The detection of the number of clusters through the maximum eigengap is potentially still possible.

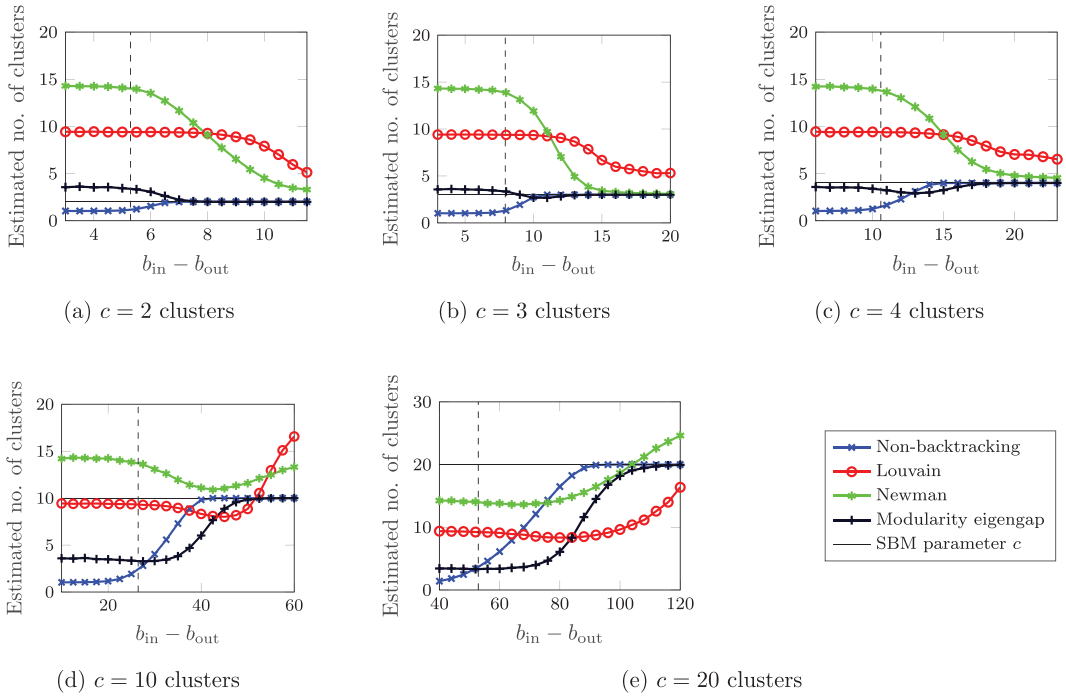


FIG. 6. The estimated of number of clusters for graphs generated by SSBMs of $N = 1,000$ nodes and c clusters. For an SSBM with a given number of clusters c, we vary the difference $b_{in} - b_{out}$ while keeping the expected average degree constant at $E[D] = 7$. Each point in the plots represents an average over 10^4 realizations. The vertical dashed line indicates the theoretical detectability limit of the SSBM.

5.3.2 *Clusters of equal size* Figure 6 shows the estimated number of clusters for networks generated by SSBMs with 2, 3, 4, 10 and 20 equally sized clusters. In each subfigure, from left to right the number of intra-cluster links increases with respect to the number of inter-cluster links, while keeping the average degree constant. The contrast of the clusters becomes stronger as the difference $b_{in} - b_{out}$ increases and the detection of clusters becomes easier. On the left side of the detectability threshold, all methods appear to have their own default guess for the number of clusters. The non-backtracking method and the modularity eigengap find the correct number of clusters already slightly above the detectability threshold for graphs with a lower number of clusters in Fig. 6a–c. Newman's iterated bisection method finds the correct number of clusters, but much higher above the detectability threshold than the first two methods. The Louvain method does not find the correct number of clusters. However, inspection of the actual clustering results from the Louvain method reveals that the true clusters are found, but they are subdivided into two or more clusters, similar to the ring of cliques in Fig. 4. For the SSBMs with a higher number of clusters c in Fig. 6d and e, the results are similar to the results for the low number of clusters. The difference is that Newman's iterated bisection does not find the correct number of clusters and the modularity eigengap needs the difference $b_{in} - b_{out}$ to be larger than the non-backtracking method before it detects the correct number of clusters c.

Figure 7 shows the simulated densities of the estimators for the case in Fig. 6c where the number of clusters $c = 4$. The three different values $b_{in} - b_{out} = \{6, 14, 23\}$ correspond to the left, centre and

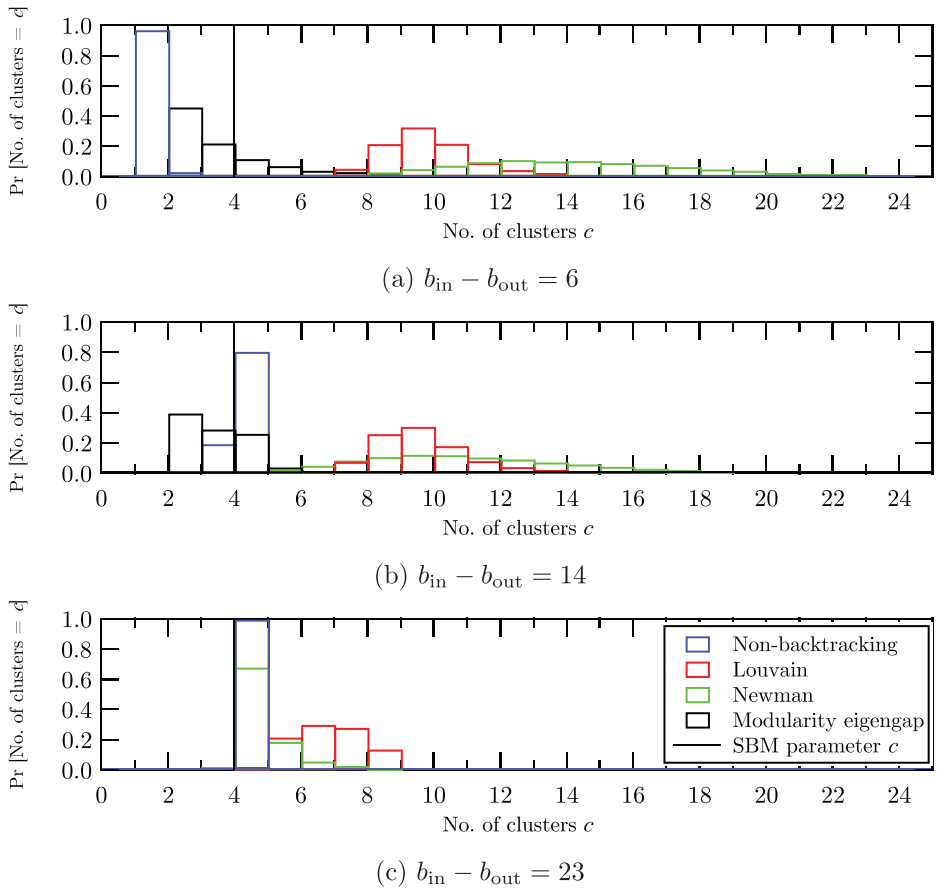


FIG. 7. Histograms of the estimates of the number of clusters c by the four different methods for graphs generated by SSBMs with $c = 4$ clusters. The values $b_{in} - b_{out} = \{6, 14, 23\}$ correspond to the left (a), centre (b) and right (c) of the interval of the experiment in Fig. 6c. The estimated probabilities are obtained by evaluating the methods on 10^4 simulated SSBM graphs with $N = 1,000$ nodes. In panel (a), none of the methods performs well since the difference $b_{in} - b_{out}$ is below the theoretical detectability limit in this case. In panel (c), both the non-backtracking method and the modularity eigengap detect the right number of clusters for each of the 10^4 networks, therefore the blue and black bars overlap.

right of the interval in Fig. 6c, respectively. The estimates of the non-backtracking method are narrowly distributed around the correct number of clusters for the two cases with the highest contrast of the clusters. Although the modularity eigengap method on average finds a number close to the true number of clusters for the case $b_{in} - b_{out} = 14$, most estimates deviate significantly from the true number, and it seems a coincidence that the average estimated number is close to the true number of clusters. Combining the three cases in Fig. 7, the distribution of the non-backtracking method is overall the most accurate.

5.3.3 *Imbalanced clusters* Figure 8 shows the estimated number of clusters for a similar experiment as in Fig. 6c, but with 4 clusters of heterogeneous size. The size of one cluster, cluster 1, is set to deviate from the sizes of the other clusters in order to assess the impact of imbalancedness. For example, in Fig. 8d

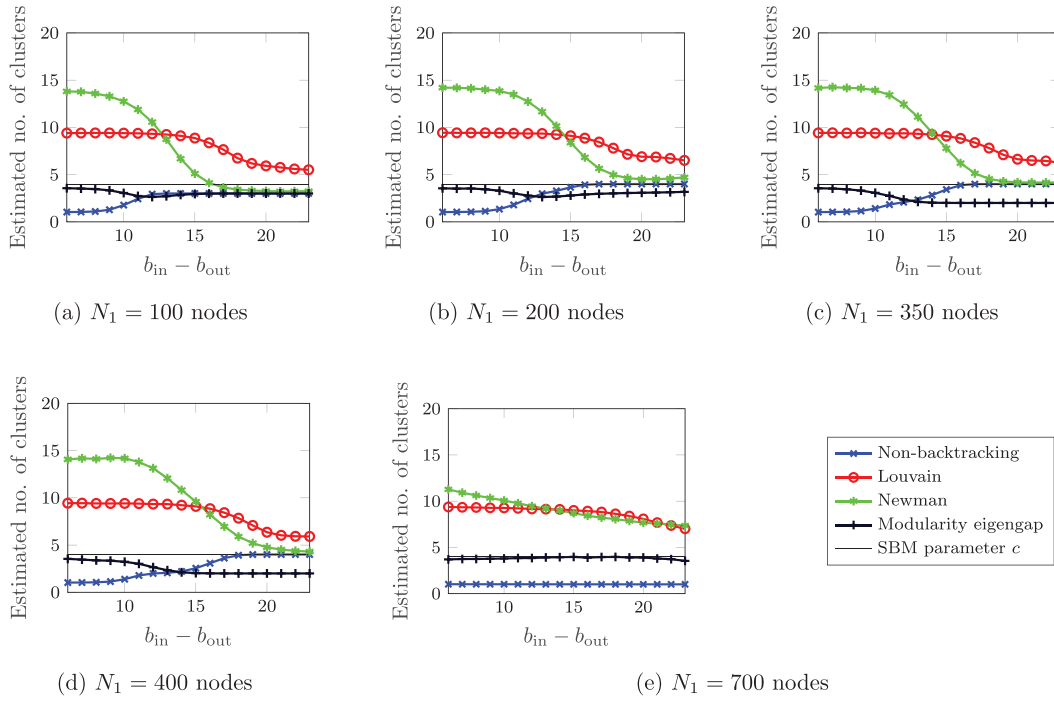


FIG. 8. The estimated number of clusters for graphs generated by SSBMs of $N = 1,000$ nodes subdivided into four imbalanced clusters. The size N_1 of cluster 1 is set to deviate from the sizes of the other three approximately equally sized clusters. The deviation of the size N_1 from the value 250 indicates the degree of imbalancedness. We vary the difference $b_{\text{in}} - b_{\text{out}}$ exactly like in Fig. 6c for comparability, but because of the imbalanced cluster sizes the average degree $E[D]$ is not equal to 7 anymore. Each point in the plots represents an average over 10^4 realizations.

the first cluster contains $N_1 = 400$ nodes and the other three clusters contain $N_2 = N_3 = N_4 = 200$ nodes each. In Fig. 8d, cluster 1 and the other three clusters have 250 nodes each. How far the size N_1 deviates from the value 250 indicates the degree of imbalancedness. Already when the clusters are slightly imbalanced (Fig. 8b and c), the modularity eigengap fails to detect the right number of clusters c. The graphs in Fig. 8a are recognized as graphs with three clusters by the detection methods and the imbalancedness appears to be too strong. In Fig. 8e, the modularity eigengap method seemingly detects the number of clusters perfectly. However, when comparing Fig. 8e with the other figures, it seems more likely that the imbalancedness is too strong and that none of the methods is able to detect the clustering structure. For the graphs generated by the SSBMs of Fig. 8a and e one could also argue that the graphs actually have three clusters and one cluster, respectively. A hard definition of the concept of a cluster would be required to make stronger statements.

5.4 *Evaluation on real-world networks*

We evaluate the four methods for detecting the number of clusters c on several real-world networks commonly used in community detection. Table 1 shows the detected number of clusters for Zachary's

TABLE 1 *Detected number of clusters for five real-world networks.*

			Detected number of clusters c			
Network	Nodes N	Links L	Non-backtracking	Louvain	Newman	Modularity eigengap
Karate	34	78	2	6	4	1
Political books	105	441	3	4	4	2
Facebook	347	2,519	8	26	18	2
Co-authorship	1,589	2,742	23	401	300	2
ArXiv	9,877	25,988	83	446	68	2

karate club network [26], a network of political books sold by Amazon[1], a social circle network from Facebook [27], a co-authorship network for publications in network science [28] and a collaboration network of the Arxiv High Energy Physics Theory category [29]. The detected number of clusters c differs significantly across the four different methods. The definition of a cluster indeed appears to depend on the method. It is difficult to make statements about the validity of the methods based on these results, since the ground truth is not known. The results of the modularity eigengap are not useful since the detected number of clusters does not seem to depend much on the network structure. Potentially the modularity eigengap method fails because the clusters in real networks are often not of equal sizes, since we know from the simulation experiment in Section 5.3.3 that the modularity eigengap does not work well for clusters of different sizes. The non-backtracking method finds a lower number of clusters c than the Louvain method and Newman's iterated bisection for most networks.

6. Conclusion

This work considers the detection of the number of clusters in a graph. Many clustering methods require the number of clusters as an input and their results depend on the chosen number of clusters. Partly also due to the lack of a clear definition of the concept of a cluster, the precise number of clusters in a graph is debatable.

We have compared the estimates based on the non-backtracking matrix with several estimators based on the concept of modularity. We find that the number of eigenvalues of the non-backtracking matrix located outside a circle in the complex plane is an excellent estimator of the number of clusters in sparse graphs where the clusters are not distinguishable based on differences in the node degrees alone. For graphs without a clustering structure, the non-backtracking method detects one single cluster, which is a convenient property for an estimator of the number of clusters. We also find that the detection based on the maximum eigengap of the modularity matrix performs similarly to the non-backtracking method for equally sized clusters, but the performance of the modularity eigengap method breaks down already when the clusters are slightly imbalanced. The estimates of the non-backtracking method are narrowly distributed around the true number of clusters for the benchmark graphs. In conclusion, the method based on the eigenvalues of the non-backtracking matrix indeed yields a clear and unambiguous decision rule for the choice of the number of clusters.

[1] Unpublished, obtained from http://www-personal.umich.edu/~mejn/netdata/.

The non-backtracking matrix is, however, an asymmmetric matrix and many of its eigenvalues are complex, making the non-backtracking method spectrally and conceptually difficult. The computational complexity can be reduced by calculating the spectrum of H^* in (A.2) from a quadratic eigenvalue equation based on the adjacency matrix. Moreover, for obtaining the estimate of the number of clusters, only the few largest, real eigenvalues have to be computed, reducing the computational complexity further. The computation time of only a few large eigenvalues of the non-backtracking matrix scales approximately linearly with the number of nodes N in the network.

The problem of detecting the number of clusters in a graph remains a difficult problem because of the lack of a hard definition of the concept of a cluster. However, we find that when loosely defining a cluster as a group of densely connected nodes, the number of eigenvalues of the non-backtracking located outside a circle in the complex plane is a good estimator of the number of clusters.

Acknowledgements

This work is part of NExTWORKx, a collaboration between TU Delft and KPN on future telecommunication networks.

References

1. Fortunato, S. (2010) Community detection in graphs. *Phys. Rep.*, **486**, 75–174.
2. Schaub, M. T., Delvenne, J.-C., Rosvall, M. & Lambiotte, R. (2017) The many facets of community detection in complex networks. *Appl. Netw. Sci.*, **2**, 1–13.
3. Newman, M. E. J. (2006) Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, **74**, 036104.
4. Newman, M. E. J. (2006) Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA*, **103**, 8577–8582.
5. Newman, M. E. J. & Girvan, M. (2004) Finding and evaluating community structure in networks. *Phys. Rev. E*, **69**, 026113.
6. Van Mieghem, P. (2010) *Graph Spectra for Complex Networks*. Cambridge, UK: Cambridge University Press.
7. Krzakala, F., Moore, C., Mossel, E., Neeman, J., Sly, A., Zdeborová, L. & Zhang, P. (2013) Spectral redemption in clustering sparse networks. *Proc. Natl. Acad. Sci. USA*, **110**, 20935–20940.
8. Van Mieghem, P., Ge, X., Schumm, P., Trajanovski, S. & Wang, H. (2010) Spectral graph analysis of modularity and assortativity. *Phys. Rev. E*, **82**, 056113.
9. Blondel, V. D., Guillaume, J.-L., Lambiotte, R. & Lefebvre, E. (2008) Fast unfolding of communities in large networks. *J. Stat. Mech.*, **2008**, P10008.
10. Chauhan, S., Girvan, M. & Ott, E. (2009) Spectral properties of networks with community structure. *Phys. Rev. E*, **80**, 056114.
11. Chung, F., Lu, L. & Vu, V. (2003) Eigenvalues of random power law graphs. *Ann. Combin.*, **7**, 21–33.
12. Wu, L., Ying, X., Wu, X. & Zhou, Z.-H. (2011) Line orthogonality in adjacency eigenspace with application to community partition. *Twenty-Second International Joint Conference on Artificial Intelligence*. (T. Walsh ed.). Barcelona, Spain: AAAI Press.
13. Wilkinson, J. H. (1965) *The Algebraic Eigenvalue Problem*. Oxford, UK: Clarendon Press.
14. Stewart, G. & Sun, J.-G. (1990) *Matrix Perturbation Theory*. San Diego, CA, USA: Associated Press.
15. Decelle, A., Krzakala, F., Moore, C. & Zdeborová, L. (2011) Inference and phase transitions in the detection of modules in sparse networks. *Phys. Rev. Lett.*, **107**, 065701.
16. Newman, M. E. & Reinert, G. (2016) Estimating the number of communities in a network. *Phys. Rev. Lett.*, **117**, 078301.

17. SHEN, H.-W. & CHENG, X.-Q. (2010) Spectral methods for the detection of network community structure: a comparative analysis. *J. Stat. Mech.*, **2010**, P10020.

18. LANCICHINETTI, A., FORTUNATO, S. & RADICCHI, F. (2008) Benchmark graphs for testing community detection algorithms. *Physical Review E*, **78**, 046110.

19. SHEA, J. M. & MACKER, J. P. (2013) Automatic selection of number of clusters in networks using relative eigenvalue quality. *MILCOM 2013-2013 IEEE Military Communications Conference*. (T. S. El-Bawab ed.). Piscataway, NJ, USA: IEEE, pp. 131–136.

20. CHEN, K. & LEI, J. (2018) Network cross-validation for determining the number of communities in network data. *J. Am. Stat. Assoc.*, **113**, 241–251.

21. CÔME, E. & LATOUCHE, P. (2015) Model selection and clustering in stochastic block models based on the exact integrated complete data likelihood. *Stat. Model.*, **15**, 564–589.

22. RIOLO, M. A., CANTWELL, G. T., REINERT, G. & NEWMAN, M. E. J. (2017) Efficient method for estimating the number of communities in a network. *Phys. Rev. E*, **96**, 032310.

23. PEIXOTO, T. P. (2017) Nonparametric Bayesian inference of the microcanonical stochastic block model. *Phys. Rev. E*, **95**, 012317.

24. HASHIMOTO, K. (1989) Zeta functions of finite graphs and representations of p-adic groups. *Automorphic forms and geometry of arithmetic varieties*. (K. Hashimoto and Y. Namikawa eds), Cambridge, MA, USA: Academic Press, pp. 211–280.

25. FORTUNATO, S. & BARTHELEMY, M. (2007) Resolution limit in community detection. *Proc. Natl. Acad. Sci. USA*, **104**, 36–41.

26. ZACHARY, W. W. (1977) An information flow model for conflict and fission in small groups. *J. Anthropol. Res.*, **33**, 452–473.

27. LESKOVEC, J. & MCAULEY, J. J. (2012) Learning to discover social circles in ego networks. *Adv. Neural Inform. Process. Syst.*, **25**, 539–547.

28. NEWMAN, M. E. (2006) Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, **74**, 036104.

29. LESKOVEC, J., KLEINBERG, J. & FALOUTSOS, C. (2007) Graph evolution: densification and shrinking diameters. *ACM Trans. Knowl. Discov. Data*, **1**, 2–41.

30. BORDENAVE, C., LELARGE, M. & MASSOULIÉ, L. (2015) Non-backtracking spectrum of random graphs: community detection and non-regular Ramanujan graphs. *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*. (A. Leverrier, J. P. Tillich and G. Zémor eds). Piscataway, NJ, USA: IEEE, pp. 1347–1357.

31. STARK, H. M. & TERRAS, A. A. (1996) Zeta functions of finite graphs and coverings. *Adv. Math.*, **121**, 124–165.

32. ANGEL, O., FRIEDMAN, J. & HOORY, S. (2015) The non-backtracking spectrum of the universal cover of a graph. *Trans. Am. Math. Soc.*, **367**, 4287–4318.

33. GOLUB, G. H. & VAN LOAN, C. F. (1996) *Matrix Computations*. Baltimore, MD, USA: Johns Hopkins University Press.

34. LEHOUCQ, R., SORENSEN, D. & YANG, C. (1998) *ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*. Philadelphia, PA, USA: SIAM.

35. THE MATHWORKS, INC., NATICK, MASSACHUSETTS. (2017) *MATLAB version 9.5.0.944444 (R2018b)*, Natick, MA, USA.

A. The non-backtracking matrix H

A.1 *Non-backtracking walks*

Two directed links are backtracking if they both connect the same pair of nodes but in opposite directions. Two directed links are non-backtracking if one follows after the other and does not loop back to the starting node of the first link. For example, the two links $e_e = i \to j$ and $e_l = i \leftarrow j$ are backtracking, while the

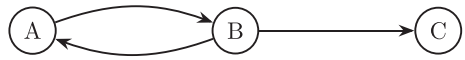


FIG. A.1. A directed graph with three nodes and three lexicographically ordered directed links $e_1 = A \to B$, $e_2 = A \leftarrow B$ and $e_3 = B \to C$.

FIG. A.2. Representing an undirected link in graph G by two bi-directional links.

two links $e_e = i \to j$ and $e_l = j \to k$ are non-backtracking for $k \neq i$. In a non-backtracking walk, no two consecutive links are backtracking.

For illustrating non-backtracking walks, consider the example of the directed graph in Fig. A.1. A walk over the links $e_1 = A \to B$ and $e_2 = A \leftarrow B$ immediately returns to its starting node and the walk is therefore backtracking. A walk over the links $e_1 = A \to B$ and $e_3 = B \to C$ does not immediately return to its beginning node and is therefore non-backtracking. The latter is a non-backtracking walk of length 2: a non-backtracking walk consisting of a succession of 2 links.

A.2 *Definition of the non-backtracking matrix H*

The non-backtracking matrix H is defined in (13). Starting from an undirected graph G, the non-backtracking matrix H is defined for each pair of bi-directional links in the bi-directional graph representation of G. Each of the L undirected links in G is represented by two bi-directional links, one in each direction, hence there are $2L$ links in the bi-directional representation (see Fig. A.2). The two bi-directional links that are created from a single undirected link in G are backtracking by construction.

Raising the matrix H to the kth power and taking the element (e, l) counts the number of non-backtracking walks of length $k+1$ from the link e_e to the link e_l. Similarly, raising the adjacency matrix A to the kth power and taking the element (i, j) counts all walks of length k from node i to node j. Krzakala *et al.* [7] and Bordenave *et al.* [30] have argued that counting only the non-backtracking walks rather than all possible walks is more informative on the structure of the graph in specific applications such as clustering.

A.3 *Spectrum of the non-backtracking matrix H*

The non-backtracking matrix was first mentioned by Hashimoto in the context of the Ihara-Bass zeta function [24]. The eigenvalues of the non-backtracking matrix H are the reciprocal of the poles of the Ihara-Bass zeta function. Stark and Terras [31] found and proved the surprising relation between the eigenvalues of the non-backtracking matrix H and the adjacency matrix A and the diagonal matrix matrix Δ with the degrees d_i on the diagonal. Following [32], the characteristic polynomial of the non-backtracking matrix H is:

$$\det(I_{2L} - zH) = (1 - z^2)^{(L-N)} \det(I_N - Az + (\Delta - I_N)z^2), \tag{A.1}$$

for complex values $z \in \mathbb{C}$ and where I_{2L} is the $2L \times 2L$ identity matrix and I_N the $N \times N$ identity matrix. The equation (A.1) has $2(L - N)$ simple zeros that are equal to $z = \pm 1$. The other $2N$ eigenvalues are

given by the quadratic eigenvalue equation on the right-hand side of (A.1). Angel *et al.* [32] remark that the $2N$ non-trivial eigenvalues can be obtained from the eigenvalue equation of a block matrix H^* constructed as follows

$$H^* = \begin{bmatrix} A & I - \Delta \\ I & O \end{bmatrix}. \tag{A.2}$$

Computing the $2N$ eigenvalues of the $2N \times 2N$ block matrix H^* rather than all $2L$ eigenvalues of the $2L \times 2L$ non-backtracking matrix H reduces the number of required computations significantly for most graphs.

A.4 *Symmetry*

The non-backtracking matrix H is an asymmetric matrix by definition: if a walk over two directed links e_e and e_l is non-backtracking, then the walk in opposite direction is not possible because of the directions of the links. If the element (e, l) in the matrix H equals 1, then the element (l, e) must always equal 0. Still, the non-backtracking matrix H features some kind of symmetry. The matrix H is defined for the pairs of bi-directional links which are created from the undirected links in the graph G. Bordenave *et al.* [30] note that if the link e_e followed by the link e_l is non-backtracking, then the same links but in opposite directions and in opposite order are also non-backtracking. It turns out that the transpose H^T of the matrix H is the non-backtracking matrix of the same graph G but with opposite directions in the bi-directional graph representation. After all, the chosen directions in the bi-directional graph representation are arbitrary (see Fig. A.2) and turning them around does not change the structure of the graph G. There exists a permutation matrix P that relabels the bi-directional links in the bi-directional graph representation such that all directions are reverted:

$$HP = PH^T, \tag{A.3}$$

where HP is a symmetric matrix. Bordenave *et al.* [30] find that while the eigenvalues of HP are strongly related to the node degrees in the graph G, the bulk of the eigenvalues of H are not.

A.5 *Complexity of the non-backtracking method*

We discuss the computational complexity of detecting the number of clusters c with the non-backtracking method as defined in (14). The size of the square non-symmetric matrix H^* in (A.2) is of an order $O(N^2)$ and calculating all $2N$ eigenvalues of H^* takes an order of $O(N^3)$ operations, for example with the QR algorithm for non-symmetric matrices [33]. For dense networks, calculating the eigenvalues of H^* offers a large reduction in computational complexity as compared to calculating all eigenvalues of H in an order of $O(L^3)$ operations. It is even more efficient to apply the iterative procedure described in Section 4.1.1 to the matrix H^*. Alternatively, it suffices to compute the k largest eigenvalues in absolute value for a small number $k << N$ if the number k is larger than the number of clusters c. An efficient way to approximate the $k << N$ largest eigenvalues of an $N \times N$ (or $2N \times 2N$) sparse matrix is with ARPACK, a collection of algorithms mainly based on the Arnoldi process [33]. ARPACK is used for eigenvalue computations in popular software packages. It is difficult to determine the exact complexity of ARPACK, but in some cases it is possible to obtain the k eigenvalues in an order of as low as $O(N)$ operations for a sparse matrix [34], which offers a great reduction in computational complexity for the non-backtracking

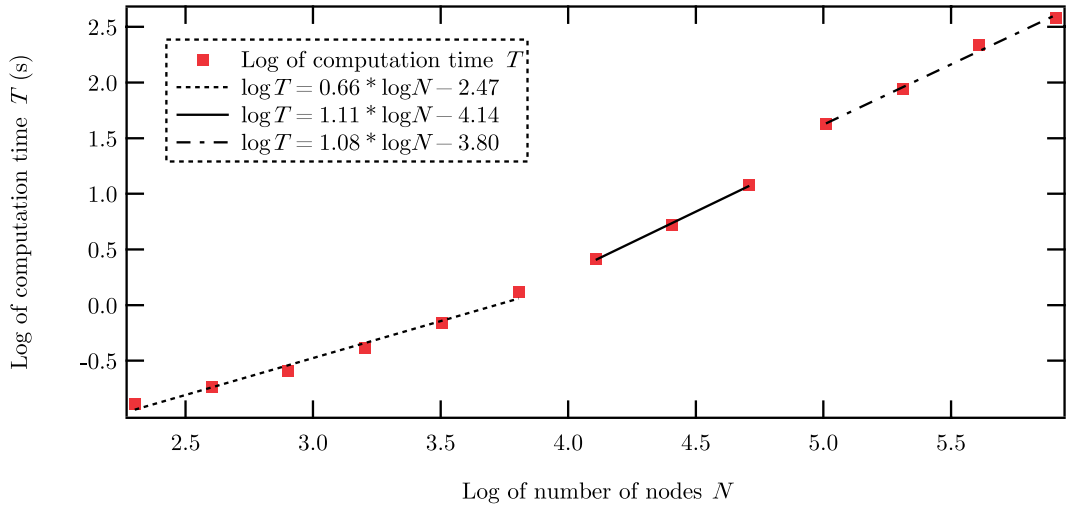


FIG. A.3. Measured computation times T in seconds for evaluating the non-backtracking method on graphs generated by an SSBM with an exponentially increasing number of nodes N, from $N = 200$ to $N = 200 \cdot 2^{12}$. The SSBM graphs are generated with the number of clusters $c = 3$ and parameters $b_{\text{in}} = 10$ and $b_{\text{out}} = 1$. The $k = 10$ largest eigenvalues are calculated from the $2N \times 2N$ matrix H^* (A.2) with ARPACK. For each network size N, we average the computation time over 20 instances. Regression lines are shown for three different sections.

method. Whether the condition $k > c$ holds has to be checked in retrospect, but it is safest to choose a number k that is much higher than the number of clusters c one expects to find.

Figure A.3 shows a numerical evaluation of the scaling of the computation time of the non-backtracking method. We measure the computation time T of estimating the number of clusters c by calculating the $k = 10$ largest eigenvalues with the ARPACK implementation of MATLAB [35]. The non-backtracking method is evaluated on graphs of exponentially increasing sizes generated by an SSBM with the number of clusters $c = 3$ and parameters $b_{\text{in}} = 10$ and $b_{\text{out}} = 1$, which is slightly above the detectability threshold in (12). Figure A.3 shows the logarithm of the computation time T plotted versus the logarithm of the number of nodes N. Regression lines are fitted to 3 different sections. In the leftmost section, the computation time T scales slower than linearly with the network size N. In the middle and rightmost sections, the computation time T scales slightly faster than linearly with the network size N. Overall, the scaling of the computation time T is close to linear. The scaling of the computation time of detecting the number of clusters with the non-backtracking method is therefore quite favourable.