Spectral Methods for Community Detection in Networks Based on Non-Backtracking Random Walks

Venelin Martinov[a,1]

[a] Department, University of Oxford, Oxford, England, United Kingdom

This manuscript was compiled on March 17, 2022

Spectral methods on matrices derived from networks have been widely and successfully used for community detection. They offer some advantages over most other methods, since they rely on essentially global information from the network. However, standard spectral methods suffer from some problems. They are very noisy on sparse networks and are disproportionately affected by small dense subgraphs. A very attractive potential solution to these problems is considering the non-backtracking matrix, which seems to address most of these issues. Several different extensions to this method have been proposed to improve performance and generalize it - the flow matrix, the reluctantly-backtracking matrix and its normalised counterpart, and the begrudgingly-backtracking matrix. In this work we compare the performance of the different extensions with the non-backtracking matrix and standard spectral methods and test results related to these methods. We show that the flow matrix, the normalised reluctantly-backtracking matrix and begrudgingly-backtracking matrix exhibit greater accuracy in their community detection on model networks and have more nicely behaved eigenvalues than either the non-backtracking or adjacency matrix in the example considered.

Non-backtracking | Random Walk | Community Detection

Many real-world networks exhibit clear community structures (1). The task of finding these structures from the network data is a well studied issue and many different methods have been employed for that (2), (3). One of the big classes of methods for finding communities is methods based on spectral properties of matrices derived from the network. The most commonly used ones are based on the adjacency matrix, the Laplacian or the normalised Laplacian. They rely on finding one or more of the eigenvalues of the matrix and then considering the corresponding eigenvectors to infer some network information from them. The main property used is that usually the bulk of the eigenvalues is grouped together and the information-carrying ones are well-separated, so easy to identify. Spectral methods rely on information global to the network, so are somewhat resistant to local anomalies. Another advantage is that there are very efficient algorithms for eigenvalue/ eigenvector computation in sparse matrices, which network matrices often are (4). In practice, these methods are often very effective, but suffer from reduced performance when used for especially sparse networks or ones with right-skewed degree distributions (5). The eigenvalues carrying the community information can get mixed with the bulk and become impractical to identify. One of extension to the spectral method, which addresses these issues is the non-backtracking matrix. It is less sensitive to high degree nodes, since it prohibits immediately returning to them and its construction makes the eigenvalues on sparse networks easier to work with. It still suffers from some problems and in this work we will consider further extensions to it, which mitigate its downsides.

Notation. In this work we have tried to keep to the notation used in (5) for directed networks, in contrast to (6) and (7). Throughout the work, the entry M_{ij} of the matrix $\mathbf{M}$, which is an operator on a network refers to the direction $j \to i$. So when the operator is considered on the directed edges of a network, the entry $M_{(i\to j),(k\to l)}$ refers to $(k \to l) \to (i \to j)$.

Stochastic Block Model

A fairly well-used model to test community algorithms in graphs is the Stochastic Block Model. In it several groups of nodes (blocks) are defined, as well as the probabilities of an edge existing between any two nodes of any given groups. All the edges exist independently of each other. In this work we use a slight variation on it, where the blocks are of equal size and the probabilities p_{in} of an edge within each block is the same. Moreover, the probabilities p_{out} of an edge between any two different groups is the same. We keep the notation of (6) and will refer to $c_{in} = p_{in}/n$, where n is the number of nodes in the network, as the average in-degree and we will refer to $c_{out} = p_{out}/n$ as the average out-degree. We will also denote the average degree $c = (c_{in} + c_{out})/2$.

The test usually performed is generating a random graph from the above model and then trying to infer the initial blocks from the graph generated. As shown in (8) the initial partition

Significance Statement

Methods for community detection in networks based on the eigenvectors of matrices derived from the network are quite effective for many uses. However, the standard methods fail on sparse networks (ones with very few connections) and on networks with a broad degree distribution. In this work we consider several extensions of the standard spectral methods which address these problems and compare their effectiveness. We also show that the eigenvalues of some of the matrices are nicer than the ones for the default methods when applied to a network with a heavy-tailed degree distribution inside its communities.

Venelin Martinov analysed the data and wrote the report

The author declares no conflict of interest

[1] To whom correspondence should be addressed. E-mail: venelin.v.martinov@gmail.com

is impossible to detect in the limit $n \to \infty$ if

$$c_{minus} = c_{in} - c_{out} < 2\sqrt{c},$$

since the structure of the blocks is lost in the noise.

The default spectral method for community detection is calculating the eigenvalue/ eigenvector pairs of the adjacency matrix. And in the case of two clusters, the sign of the node entries in the eigenvector for the second largest eigenvalue are considered and grouped together. A similar process can be performed with the corresponding Laplacian/ normalised Laplacian matrix, but the second smallest eigenvalue is considered. We will restrict our attention to 2 blocks in the networks, although most of the results are applicable to more, with some modifications (taking more eigenvectors and performing a clustering on their entries).

Non-Backtracking Random Walk and Matrix

The non-backtracking random walk on a graph is a random walk on the nodes of the graph, which is not allowed to return to the node it last visited. It is, however, more convenient to view this as a random walk on the directed edges of a network, because it makes the process easier to define as an actual random walk. It is only valid on graphs with no degree one nodes. The non-backtracking matrix $\mathbf{B}$, defined on the directed edges of the network is given by:

$$B_{(i\to j),(k\to l)} = \begin{cases} 1, & \text{if } l = i \text{ and } k \neq j \\ 0, & \text{otherwise} \end{cases}$$

Community detection using this matrix is then performed by calculating its second largest eigenvalue, taking the corresponding eigenvector and summing up the entries of the edges pointing to each node. Nodes are then grouped by the sign of the resulting vector.

The definition yields some useful results according to (6), including the fact that the spectral properties of this matrix are less sensitive to high degree nodes within the network and that the matrix is able to detect structure in the block model down to the theoretical limit in the sparse regime. Moreover, its largest eigenvalue is asymptotically the average degree of the network nodes as $n \to \infty$ in the stochastic block model. The matrix still has some downsides, since its suffers from somewhat low performance in some cases and completely ignores dangling trees in the network. Since its main use is in sparse networks, where trees are quite common, ignoring them is an undesirable property.

Flow Matrix and Begrudgingly-Backtracking Random Walk

The flow matrix $\mathbf{F}$ was introduced in (5) and is essentially a normalised version of the non-backtracking matrix. It is defined on the directed edges of the network and its elements are

$$F_{(i\to j),(k\to l)} = \begin{cases} 1/(d_i - 1), & \text{if } l = i \text{ and } k \neq j \\ 0, & \text{otherwise} \end{cases},$$

where d_i is the degree of node i. It exhibits some nice properties when compared to the normal non-backtracking matrix, since its spectrum seems to behave somewhat nicer and it is connected to the modularity metric, widely used for community detection algorithms (5). It is also well behaved on some networks with a broad degree distribution, where the non-backtracking matrix fails (5).

A slight variation of the flow matrix, which also considers degree one nodes is the begrudgingly-backtracking matrix, related to the begrudgingly-backtracking random walk. It was defined in (9). It enjoys many of the same properties as the flow matrix. In it the walker acts as the non-backtracking walker, except that when they reach a degree one node, they are allowed to come back. The entries of the begrudgingly-backtracking matrix $\hat{\mathbf{F}}$ are defined, accordingly, as

$$\hat{F}_{(i\to j),(k\to l)} = \begin{cases} 1/(d_i - 1), & \text{if } l = i \text{ and } k \neq j \\ 1, & \text{if } l = i, k = j \text{ and } d_i = 1 \\ 0, & \text{otherwise} \end{cases}.$$

Community detection with these matrices is performed identically to the non-backtracking matrix.

Reluctantly-Backtracking Matrix

The reluctantly-backtracking matrix and its normalised version, defined in (7) are another variation on the non-backtracking matrix. In the corresponding random walk, the walker is allowed to return with some probability, inversely proportional to the degree of the node it just left. This allows including degree one nodes in the graph and further mitigates the effect of network hubs on the spectrum of the matrix (7). The entries of the reluctantly-backtracking matrix $\mathbf{R}$ are

$$R_{(i\to j),(k\to l)} = \begin{cases} 1, & \text{if } l = i \text{ and } k \neq j \\ 1/d_k, & \text{if } l = i \text{ and } k = j \\ 0, & \text{otherwise} \end{cases}.$$

The normalised reluctant-backtracking matrix $\mathbf{P}$ is defined by

$$P_{(i\to j),(k\to l)} = \begin{cases} 1/(d_l - 1 + 1/d_k), & \text{if } l = i \text{ and } k \neq j \\ 1/d_k(d_l - 1 + 1/d_k), & \text{if } l = i \text{ and } k = j \\ 0, & \text{otherwise} \end{cases}.$$

Community detection, as defined in (7), with these matrices is performed by summing up the edge entries by the source node, in contrast to the other matrices considered here.

Variation of Information

In this work we have used the variation of information metric defined in (10) to compare different partitions of the same network. It is an information based metric which measures how much information is lost and gained in switching between two partitions. It was chosen as it exhibits some nice properties as detailed in (10) and is an actual metric in the mathematical sense. By the bound proven in the paper, for our case with only two clusters in each partition, the variation of information is bounded above by $2\log 2$. Hence, when comparing two different partitions of the same network, a variation of information of 0 tells us that the partitions are the same, and a variation of information of $2\log 2 \approx 1.386$ tells us that the partitions have essentially no information shared.

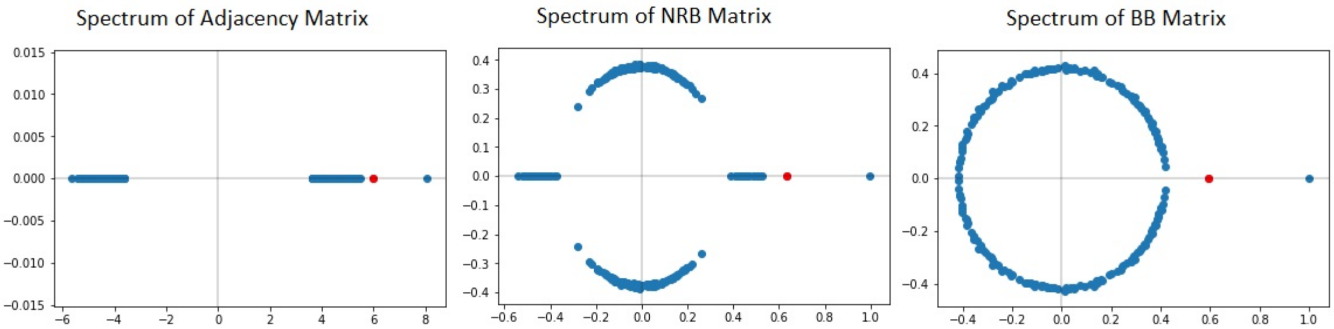

Fig. 1. Block model spectra of the three matrices. Only the largest 200 eigenvalues in magnitude are plotted. The blocks are of size 500 and an average degree of 3.5 with $c_{minus} = 2$ was used. In each spectrum, the second largest eigenvalue in magnitude is in red. The second largest eigenvalue of the adjacency matrix is very close to the bulk of the eigenvalues, while the eigenvalues of the other two matrices seem relatively well-separated.

Comparison Setup

Here we present the simulations for the performance of the different clustering algorithms. Each one was done on a range of values for c_{in} and c_{out}, and its variation of information with the planted partition was measured and averaged over several realisations. Two different simulations were performed for the different algorithms. The first one is done on the 2-core (the subgraph with nodes of degree at least two) of a random realisation of the stochastic block model for the non-backtracking ($\mathbf{B}$), flow ($\mathbf{F}$), reluctantly-backtracking ($\mathbf{R}$) and normalised reluctantly-backtracking ($\mathbf{P}$) matrices. They were compared to a clustering derived from the adjacency matrix and a random partition of the nodes into two groups. The second simulation is done on the largest connected component (the 1-core) of a random realisation of the block model for the begrudgingly-backtracking ($\hat{\mathbf{F}}$), reluctantly-backtracking ($\mathbf{R}$) and normalised reluctantly-backtracking ($\mathbf{P}$) matrices. They were again compared to a clustering derived from the adjacency matrix and a random partition of the nodes into two groups. Note that in both simulations parameters were chosen so that the largest connected component and the 2-core of the model realisations had almost as many nodes as the original.

We also compare the eigenvalue spectra of the adjacency, normalised reluctant-backtracking ($\mathbf{P}$) and begrudgingly-backtracking ($\hat{\mathbf{F}}$) matrices on two different random graph models. The first is as before - stochastic block model with equal sized partitions and symmetric communities. The second graph aims to model networks with heavy-tail degree distributions inside the communities. It was generated as a union of two configuration models with power-law degree distributions and a number of edges where randomly placed between the two communities.

We have also included a comparison of the methods on a real-world network - the Zachary Karate Club graph. While not exactly well-suited to the methods, the network shows some relevant information.

Results

In this section we present the results of our simulations as well as possible interpretations.

A. Comparison on 1-core. The comparison of the methods on the 1-core of the random graph (in the left column of 3) showed that in the sparse regime, the normalised reluctant-backtracking matrix ($\mathbf{P}$) and the begrudgingly-backtracking matrix ($\hat{\mathbf{F}}$) perform consistently better than the adjacency matrix or the non-normalised reluctant-backtracking matrix ($\mathbf{R}$). All the matrices perform comparably on the denser graphs. Random partitions perform as expected at around $2 \log 2$.

B. Comparison on 2-core. The comparison of the methods on the 2-core (in the left column of 3) yields similar results as the 1-core. The flow ($\mathbf{F}$) and normalised reluctant-backtracking ($\mathbf{P}$) matrix performed the best. The non-backtracking matrix ($\mathbf{B}$) seems to perform quite well in the less well-defined partitions. The random partitions again perform as expected at around $2 \log 2$.

C. Eigenvalues. We compare the spectra of the adjacency, normalised reluctant-backtracking ($\mathbf{P}$) and begrudgingly-backtracking ($\hat{\mathbf{F}}$) matrices on two different random graph. The first is a realisation of the planted partition model, while the second one is the heavy-tail degree distribution model defined above. The results can be seen on 1 and 2.

D. Real-World Network. The performance of the methods on the real-world network example is not very good and we have summarised the variation of information of the predicted clustering with the actual division in the club in table 1. The karate club is not a sparse graph and the usual spectral methods (adjacency matrix/ Laplcaian) work well. A large accuracy difference between the normalised matrices (flow and normalised reluctantly-backtracking) and the non-normalised ones (non-backtracking and reluctantly-backtracking) can be observed.

Discussion

In this work we have presented several extensions to the non-backtracking matrix and their uses in finding community structures in networks. We have compared the performance of the flow, begrudgingly-backtracking, reluctantly-backtracking and normalised reluctantly-backtracking matrices in detecting community structure with the performance of the adjacency matrix and the non-backtracking matrix. We have shown that on the simulated cases, the begrudgingly-backtracking/ flow matrix and the normalised reluctantly-backtracking matrix perform

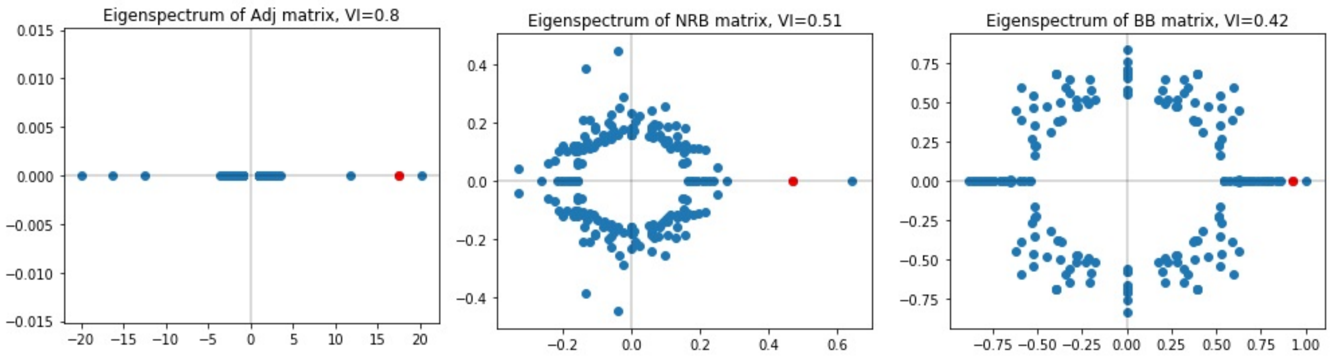

Fig. 2. Union of two configuration models with a -2.5 power-law degree distributions. The two graphs are of size 500 each and are joined by 50 randomly placed edges between them. In each plot, the second largest eigenvalue in size is in red. The second eigenvalue of the normalised reluctantly-backtracking matrix is well separated from the bulk and its values seem much less affected by the high-degree nodes than the ones in the adjacency matrix. The begrudgingly-backtracking eigenvalue is not well-separated, although the eigenvalues close to it are the ones arising from the "diagonal" entries (the ones allowing it to return), since they are not there in the flow matrix.

Table 1. Comparison of the variation of information in the different matrix methods' accuracy when used for community detection in the Karate Club graph. The VI column is the variation of information between the predicted communities and the observed split in the club.

Method	VI
1. Adjacency	0.28
2. Non-Backtracking	1.19
3. Flow	0.43
4. Reluctant	1.19
5. Normalised Reluctant	0.43

better than either the adjacency or the non-backtracking matrix. Moreover we have shown that their spectra behave more nicely than the spectrum of the adjacency matrix in the example generated.

A clear division in the utility of the different matrices is not very obvious. The normalised reluctantly-backtracking matrix seems especially well suited for heavy-tailed degree distributions, but it might also be the effect of noise. In the block model graphs, the flow/ begrudgingly-backtracking matrix seem optimal.

The work prompts some further analysis related to the comparison of these methods. The first one is extending the simulations to larger networks and ones composed of more clusters. A more rigorous analysis of the results from the network with heavy-tailed degree distribution is also in order, as well as using different models to simulate these kinds of networks.

Moreover, we did not manage to reproduce the results in (7) for community detection in a union of two trees connected by a single edge. In our simulations the two communities predicted had almost no relation to the planted communities.

We observed the approximate modularity optimisation of the flow matrix, since in most of the simulations performed, the flow matrix/ begrudgingly-backtracking matrix predicted better communities in terms of modularity than the planted ones. The results where incomplete and were not included here.

Materials and Methods

The simulations for the comparison of the matrix performances was averaged over 5 different realisations of each parameter choice. The sparsity of the matrices was heavily used throughout the work and algorithms for sparse matrices were especially useful. The power law distributions were generated using (11). Since the numbers generated by the algorithm were at least one, there were almost no nodes outside the largest connected component. The ends of the middle edges, connecting the two components were chosen uniformly at random, but since there was so few of them, the distribution was mostly unaffected.

1. Girvan M, E.J. Newman M (2001) Community structure in social and biological networks. *proc natl acad sci* 99:7821–7826.
2. Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008) Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment* 2008:10008.
3. Girvan M, E.J. Newman M (2001) Community structure in social and biological networks. *proc natl acad sci* 99:7821–7826.
4. Sorensen DC (1996) Implicitly restarted arnoldi/lanczos methods for large scale eigenvalue calculations.
5. Newman MEJ (2013) Spectral community detection in sparse networks. *arXiv e-prints* p. arXiv:1308.6494.
6. Krzakala F, et al. (2013) Spectral redemption in clustering sparse networks. *Proceedings of the National Academy of Science* 110:20935–20940.
7. Singh A, Humphries MD (2015) Finding communities in sparse networks. *Scientific Reports* 5:8828.
8. Mossel E, Neeman J, Sly A (2012) Stochastic Block Models and Reconstruction. *arXiv e-prints* p. arXiv:1202.1499.
9. Rappaport B, Gamage A, Aeron S (2017) Faster Clustering via Non-Backtracking Random Walks. *arXiv e-prints* p. arXiv:1708.07967.
10. Meila M (2007) Comparing clusterings – an information based distance. *Journal of Multivariate Analysis* 98:873–895.
11. Alstott J, Bullmore E, Plenz D (2014) powerlaw: A Python Package for Analysis of Heavy-Tailed Distributions. *PLoS ONE* 9:e85777.

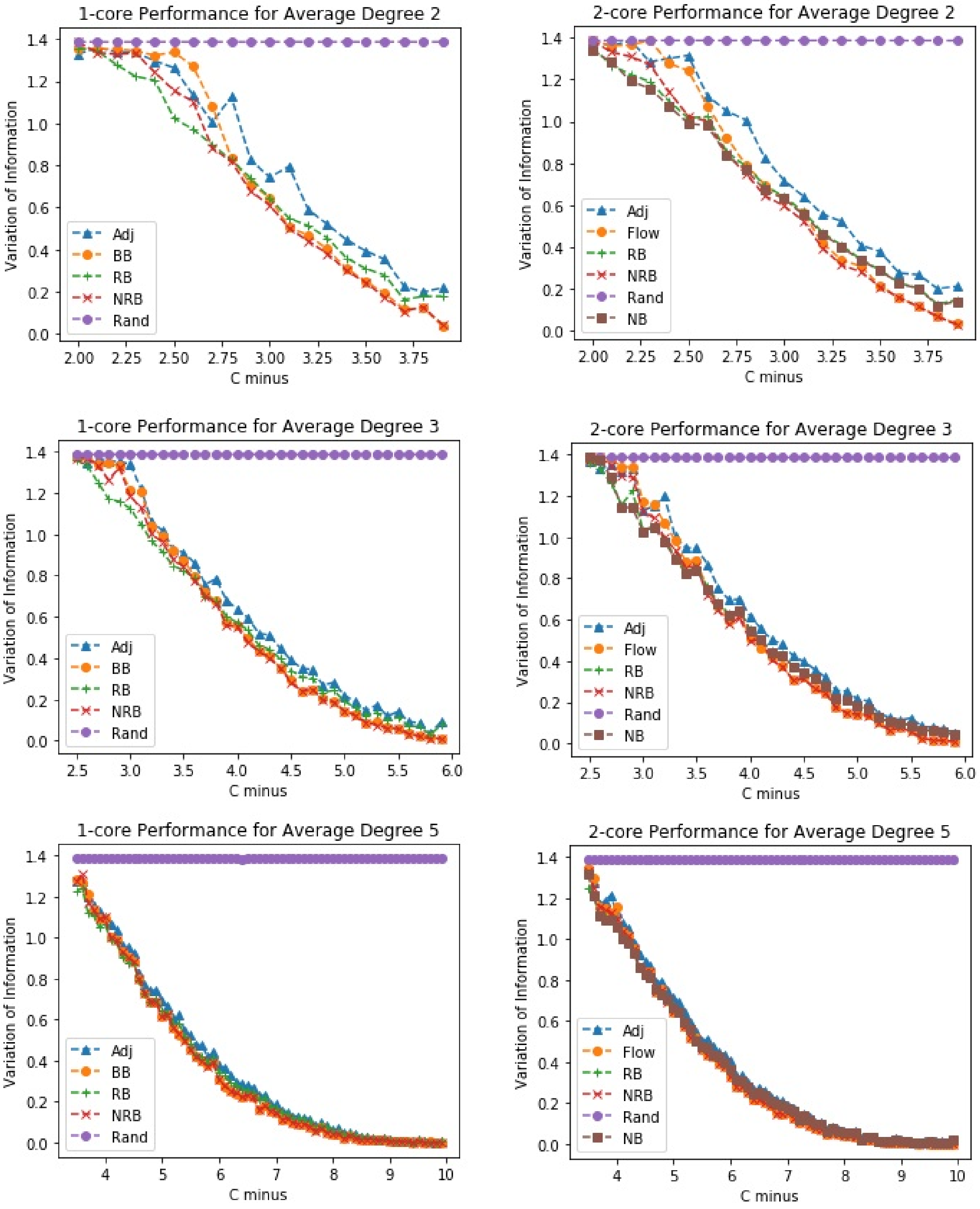

Fig. 3. Performance of the different matrix methods. The left columns shows the ones which allow degree one nodes, while the right one shows all the methods. The begrudgingly-backtracking matrix ($\hat{\mathbf{F}}$) is the same as the flow matrix ($\mathbf{F}$) on the 2-core of a graph. All the simulations are performed on two planted groups of size 1000 in a stochastic block model with varying average degree and c_{minus} value. c_{minus} is the difference between c_{in} and c_{out}. It shows how well separated the two groups are.