**Community Size Detection Algorithms**

Table of Contents

[1. Introduction 4](#_Toc26644853)

[2. Hypothesis 5](#_Toc26644854)

[3. Literature Review 6](#_Toc26644855)

[3.1. Erdös-Réyni (ER) Random Graph 6](#_Toc26644856)

[3.2. The Stochastic Block Model (SBM) 6](#_Toc26644857)

[3.3. The Degree Corrected Block Model (DCBM) 8](#_Toc26644858)

[3.4. The Bi-Partite Stochastic Block Model (biSBM) 8](#_Toc26644859)

[3.5. Dynamic Networks 10](#_Toc26644860)

[4. Survey of Algorithms 10](#_Toc26644861)

[4.1. Preliminaries 10](#_Toc26644862)

[4.2. Bethe Hessian (BH) 11](#_Toc26644863)

[4.2.1. The BH Algorithm 11](#_Toc26644864)

[4.2.2. Discussion 12](#_Toc26644865)

[4.3. Recursive Bipartitioning Algorithm (RB) 14](#_Toc26644866)

[4.3.1. Preliminaries 14](#_Toc26644867)

[4.3.2. The Algorithms 14](#_Toc26644868)

[4.4. Network Cross Validation (NCV) 16](#_Toc26644869)

[4.4.1. The Main NCV Algorithm 16](#_Toc26644870)

[4.4.2. Discussion of the NCV Algorithm 17](#_Toc26644871)

[4.5. Edge Cross-Validation (ECV) 18](#_Toc26644872)

[4.5.1. The Algorithm 18](#_Toc26644873)

[4.5.2. Discussion 19](#_Toc26644874)

[4.6. Minimum Description Length (MDL) 20](#_Toc26644875)

[4.6.1. Preliminaries 20](#_Toc26644876)

[4.6.2. Algorithm 21](#_Toc26644877)

[4.6.3. Discussion 21](#_Toc26644878)

[5. Data, Variables, and Statistical Methods 21](#_Toc26644879)

[6. Simulation Results and Discussion 24](#_Toc26644880)

[7. Dynamic Network Changepoint Detection Algorithm 31](#_Toc26644881)

[7.1. Algorithm 31](#_Toc26644882)

[7.2. Simulation Results 32](#_Toc26644883)

[8. Conclusion and Next Steps 34](#_Toc26644884)

[9. References 36](#_Toc26644885)

# Introduction

While network data abound in business, accessible theoretical and empirical tools to study such data are just developing. One example of network data in business settings could involve trade networks, where firms or countries are represented by nodes and trades between a given pair of firms or countries are represented by edges, with edge weights proportional with trade volumes.

The stochastic block models and their derivatives such as the degree corrected block models are powerful tools for representing network data with innate community structures. For this reason, the community detection problem for stochastic block models has received ample attention from the scientific research community in recent years in contrary to social sciences including business. The diversity of proposed methods in the scientific literature is matched only by the wide spectrum of analytical tools utilized, ranging from spectral clustering, maximum likelihood, and belief propagation.

However, regardless of the method choice, one important input is the number of clusters, which is simply assumed to be known in popular clustering algorithms. In business, the number of clusters itself is often a parameter of interest and has potential as a source of alpha generation in investment alpha models (Larcker, So, and Yang 2013; Van Peteghem, Bruynseels, and Gaeremynck 2018; Nguyen and Nielsen, 2010).

While there are several cutting-edge algorithms proposed in the last few years for detecting community sizes for networks that frequently occur in practice (Bickel and Sarkar 2016; Chen and Lei 2018; Levina, Li, and Zhu 2019), most of the work that has been done is largely theoretical that speaks to the theoretical efficacy and mathematical proofs about asymptotic consistency of algorithms. On few occasions, the authors have published open-source code that attempt to implement the algorithms in papers with varying levels of success in terms of efficiency and consistency. However, to the best of the author’s knowledge, just two such algorithms have been implemented: one in R (Levina, Li, and Zhu 2019) and the other in Python (Peixoto 2013). The rest (Bickel and Sarkar 2016; Chen and Lei 2018; Ma, Su, and Zhang 2019) are still in the theoretical, or early proof-of-concept, stage without a concrete implementation.

Further, most of the currently available algorithms are not directly applicable to one particular type of network data that commonly arises in business. This is because the algorithms in the literature assume a unipartite structure, while some of the important networks arising in business setting are bipartite.

This paper begins by reviewing the building-block models for representing network data, namely the stochastic block model and its variants. Then, it introduces several of the popular algorithms proposed in recent literature for determining , along with their main theoretical properties. It compares these algorithms for accuracy and efficiency, and selects the best-performing algorithm to be used in the dynamic networks portion. In the next part of the paper is a discussion of key observations on the behavior of network statistics around the changepoint, followed by an algorithm that encapsulates the main ideas presented. Lastly, the paper proposes a conjecture on the detectability threshold of the community size.

# Hypothesis

Of the five algorithms discussed in detail below, the Bethe Hessian algorithm is expected to have the smallest lower-bound on the edge density of the network, and hence outperform the other algorithms in inferring community sizes for sparse networks. For dense networks, it is expected to perform at least as well as the others. Its use of the robust symmetric adjacency matrices and efficient spectral decompositions is expected to yield the best space and time complexity of all the algorithms reviewed thus far. In addition, its theoretical guarantees for consistency are expected to carry for dynamic networks as well in not only inferring community sizes, but also community labels for individual nodes. In the dynamic setting, the community size estimated using the Bethe Hessian algorithm is expected to be an informative metric for detecting changepoints.

# Literature Review

## Erdös-Réyni (ER) Random Graph

In 1959, Erdös and Réyniintroduced a generative model for random graphs that formed the basis for much of the earlier work for clustering. A homogeneous ER random graph is one of the simplest models to generate network data, and is parameterized by the number of nodes, , and the probability of linkage, , for any pair of nodes. Given and , the ER model, denoted , constructs a random undirected graph by randomly connecting pairs of nodes, such that the probability of each linkage is and each ER random graph generated as such is equally likely. The parameters and are hence taken to be fixed and define an node linkage probability matrix , where and is a identity matrix. An empirical estimate of , denoted , can be expressed using , an estimate for parameter .

An inhomogeneous ER model generalizes the homogeneous ER model by assuming that edges between nodes and are drawn independently according to the linkage probability . The stochastic block model generalizes the homogeneous ER model by assuming a latent community structure among the nodes.

## The Stochastic Block Model (SBM)

In 1983, Holland et al. introduced the SBM, a popular method to model undirected network structure. Given the number of communities and nodes, SBM assigns a latent community label, , , to each node according to a multinomial distribution. This community membership can be represented by a matrix of dimension , such that row corresponding to node is a unit vector comprised of all zeroes except a single 1 whose column index indicates node ’s community. Hence, each row of sums to 1 and each column sums to , which can be interpreted as the size of community , such that .

Hence, SBM is a generative model with parameters and , where is a matrix of probabilities of edges between nodes in the graph, and modeling with SBM entails finding estimates of the parameters for and . A distinguishing feature of the SBM is that all edges in a given community have stochastic equivalence, that is, have identical probability of linkage. For any given pair of communities and b, the probability of linkage between the nodes belonging communities and is notated by . Hence, is symmetric and is often called the community probability matrix.

Given the stochastic equivalence of SBM models, for each pair of nodes, and , the edge connecting them is notated by , and is a realization of a Bernoulli random variable, independent of all other edges , for , with the probability of linkage depending only on the community memberships of the nodes. That is,

and hence,

with the main task of estimating from . The concatenation of these edges gives rise to a symmetric matrix , which is known as the adjacency matrix. By convention, self-linkages are not allowed in adjacency matrices, that is, for and linkage probabilities between communities take on positive values, that is,

While concise, SBM suffers from several criticisms based on empirical grounds. One such criticism is the stochastic equivalence assumption, which implies that SBM does not model within-community variations in edge probabilities, while real-world data often display such variations. This has led to a development of several variants of the SBM, such as the Degree Corrected Block Model (DCBM), the mixed membership model introduced by Airoldi et al. (2008), and the latent space model proposed by Hoff et al. (2002). The Degree Corrected Block Model is discussed below while providing the reference sources for the others for the interested reader.

## The Degree Corrected Block Model (DCBM)

Karrer and Newman (2011) introduced a generalization of the stochastic block model by allowing for a heterogeneity of node degrees within a community with an addition of degree parameters vector for each node . Given the membership vector , where the community membership of node is , and community probability matrix , the probability of an edge between nodes and is

Hence, the DCBM is parameterized by , with the identifiability condition

to ensure model identifiability. Both SBM and DCBM produce node linkage probability matrix of rank corresponding to the number of innate communities.

## The Bi-Partite Stochastic Block Model (biSBM)

Many of the network data in business are bi-partite. In bi-partite networks, there are two classes of nodes, and edges can be formed only between nodes of different classes. One example is the corporate director network where the corporate boards and directors comprise the two classes, edges are formed between directors that serve on the same board. Another example comprises suppliers to firms in an oligopsony that would form a link if they sell to the same buyer.

Larremore, et al. proposed in 2014 an algorithm specific to bipartite graph structure. A common approach to working with bipartite data is to derive a unipartite graph from a bipartite structure via a one-mode projection, which infers edges from overlapping cliques in the bipartite network and eliminates all of the nodes belonging to one of the classes. However, as noted by Larremore et al (2014), this approach has undesirable theoretical issues, such as loss of information due to the projection and emphasis on clique overlaps which are low-probability events.

Larremore et al. proposed a bipartite SBM by reformulating the adjacency and community probability matrices for bipartite networks. Given two groups and , let be the number of nodes in group and likewise for and let and be the number of communities in groups and , respectively. Finally, let be the bipartite matrix and be the community probability matrix for the two groups. Then, bipartite analogs of matrices and would be as follows:

and

From here, one can apply the same algorithms presented below that assume a unipartite graph structure.

## Dynamic Networks

The community detection algorithms discussed in this paper originally only considered static networks. In this paper, dynamic networks comprised of a series of static networks at discrete points in time are considered. In particular, the dynamic networks are considered in two forms. The first is a sum of the static networks over the time horizon. The second is a sum of static matrices whose elements are squared values of the original static network. The algorithms considered in this paper will be tested using simulations based on both static and dynamic networks.

# Survey of Algorithms

In this section, five algorithms from recent literature are discussed for estimating the number of communities in network data. For each algorithm, key terminologies and theoretical properties for main use cases are discussed before delving into the details of the algorithm. Then, the performance of the algorithms are compared and contrasted using simulated and real-world data.

## Preliminaries

The following review is based on Saade et al. (2014). Given the symmetric random adjacency matrix , the degree of node is given by and the expectation of is expressed as . The mean of expected degrees is . The th largest eigenvalue of a symmetric matrix is denoted by . The summed adjacency matrix is denoted and defined . The squared adjacency matrix is denoted and defined , and the summed squared adjacency matrix is denoted and defined .

Given a adjacency matrix , the non-backtracking operator is a non-symmetric matrix where each element is , for , where is the Kronecker delta function based on the value of , and the index refers to a directed edge from node to node . The Bethe Hessian (BH) matrix is expressed as , where is a parameter and is the diagonal matrix such that for .

## Bethe Hessian (BH)

BH matrices have traditionally been studied in physics and information sciences, until Le et al (2014) applied them to recover community structures in network data, and Saade et al. (2014) demonstrated their use in estimating the number of communities.

### The BH Algorithm

The usefulness of the non-backtracking operators comes from the fact that, for graphs constructed by SBMs, the number of communities, i.e., , is equal to the number of eigenvalues lying outside the radius . In particular, for disassortative SBM graphs, the number is given by where , and for assortative graphs, , for . However, one formidable challenge in estimating K this way is due to its asymmetric structure, which makes implementation computationally challenging, especially as grows large.

Fortunately, the usefulness of BH matrices comes from the one-to-one correspondence between the number of negative eigenvalues of , referred to as “informative eigenvalues” by Le et al., and the number of informative eigenvalues of . Several methods have been proposed in estimating the parameter . Saade et al. suggest , where and are the first and second moments of the distribution of , . Given the input adjacency matrix , the BH algorithm based on as an estimate for proceeds as follows:

|  |  |  |
| --- | --- | --- |
| **Algorithm 3.1.1 Bethe Hessian – Non-Backtracking Method** | | |
| 1. | **Procedure ()** | |
| 2. |  | Compute |
| 3. |  | Compute |
| 4. |  | Computethe vector of eigenvalues of BH, and let |
| 7. |  | Order from largest to smallest |
| 8. |  | Return for some |

In addition to , the authors also propose , especially for unbalanced networks for which appears to perform poorly. The BH algorithm below is based on :

|  |  |  |
| --- | --- | --- |
| **Algorithm 3.1.2 Bethe Hessian – Average Method** | | |
| 1. | **Procedure ()** | |
| 2. |  | Compute |
| 3. |  | Compute |
| 4. |  | Computethe vector of eigenvalues of BH, and let |
| 7. |  | Order from largest to smallest |
| 8. |  | Return for some |

### Discussion

Saade et al (2014) argued that the eigenvectors corresponding to the negative (“informative”) eigenvalues of encode the community structure. Le et al (2015) noted that the number of negative eigenvalues directly denotes the number of communities. In choosing the value of , the method estimates the parameter as , where is an approximation of the spectral norm of the non-backtracking matrix. Under the method proposed by Saade et al, is taken to be . Regardless of the specific method used, the authors note that BHc and BHa both tend to underestimate . To compensate for this, one computes the upper bound of K, , to which K converges to asymptotically:

where are eigenvalues of ordered from largest to smallest and is a hyper-parameter, which Le et al note works well for .

An important consideration in assessing community detection algorithms is the consistency property, which states whether the estimator is asymptotically equal to the parameter value. In general, when the number of edges scales with , the network is called a dense regime. Otherwise, it is called a sparse regime. Under mild conditions, both BH algorithms have been shown to be consistent. Bordenave et al. (2018) showed that for sparse regimes with , for some fixed symmetric and for all the number of eigenvalues of NB at least is equal to K as . For dense regimes with and , Le et al. (2015) showed that has exactly negative eigenvalues with high probability.

## Recursive Bipartitioning Algorithm (RB)

Bickel and Sarker (2013) showed that recursively testing the null hypothesis to determine for DCBMs works well under the condition that the average degree grows faster than as grows, which is a milder condition than the requirement that it grow faster than in Wang and Bickel (2016).

### Preliminaries

Following the notation from Bickel et al, given the normalized adjacency matrix and its eigenvalues in decreasing order , the test statistic converges weakly to the Tracy–Widom distribution with index 1 (TW1), where and are estimates for and , respectively, as follows:

, where is an estimate of

### The Algorithms

Bickel et al’s algorithm below incorporates a small sample correction for adjacency matrices since the latter have shown to converge to TW1 very slowly. The authors propose to compute the bootstrapped empirical distribution of based on a sample of ER simulations, then center and scale the test statistic to match the first two moments of the TW1 distribution.

|  |  |  |
| --- | --- | --- |
| **Algorithm 3.3.2a Hypothesis Test** | | |
|  | **Procedure ()** ▷ significance level, m sample size | |
| 1. |  |  |
| 2. |  | ▷ is the test statistic |
| 3. |  |  |
| 4. |  |  |
| 5. |  | for do ▷ By default, for bootstrap |
| 6. |  |  |
| 7. |  |  |
| 8. |  | mean ▷ for small sample correction |
| 9. |  | ▷ for small sample correction |
| 10. |  |  |
| 11. |  | pval |

The authors note that in step 5, increasing the value of m beyond the default 50 brings small marginal improvement in accuracy at the expense of adding overhead to the recursive call to the procedure.

Below is the algorithm that recursively tests the null hypothesis versus the alternative in Steps 5 and 6.

|  |  |  |
| --- | --- | --- |
| **Algorithm 3.3.2b Recursive Bipartitioning** | | |
| 1. | **Procedure biPartition()** ▷ = significance level | |
| 2. |  | **HypothesisTest(A)** |
| 3. |  | **If**  then |
| 4. |  | ▷ Using regularized spectral clustering |
| 5. |  |  |
| 6. |  |  |

## Network Cross Validation (NCV)

NCV was presented by Chen et al. (2017) as a model selection methodology under the assumption that networks are generated by SBM and DCBM, and is based on a block-wise node-pair splitting technique. A traditional approach to network analysis involving splitting nodes in a network received much criticism for removing edges incident to the split boundaries and modifying the original network structure. Contrary to the node splitting technique, NCV randomly splits nodes into and , after which one uses the edges as the training set, and the edges as the test set. An advantage that results from this procedure is that it incorporates all the information from the edges . Another would be that all model parameters, such as the membership vector, can be accurately estimated from the training set.

A noteworthy assumption in NCV is that the community membership vector for nodes is assumed to be parameters, rather than unknown variables to be estimated, simplifying the relationship between the edge distributions in training and test sets. A practical advantage of the NCV method is that, in addition to estimating the number of communities , it also does a model selection, for instance, the stochastic block model versus the degree corrected block model.

### The Main NCV Algorithm

In the below V-fold NCV algorithm, the input is comprised of Adjacency matrix , set of values , and the number of folds .

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm 3.2.1 V-fold NCV** | | | |
| 1. | **Procedure NCV()** | | |
| 2. |  | **Split** nodes into equal-sized | |
| 3. |  | **Split** into equal-sized | |
| 4. |  | **for** and **do** | |
| 5. |  |  | **Estimate** using for DCBM) |
| 6. |  |  | **Validate** with loss |
| 7. |  | **Let** | |
| 8. |  | **Return** | |

### Discussion of the NCV Algorithm

Steps (2) and (3) implement the block-wise node-pair splitting, where letting be the training set and be the test set, we end up with the following block form:

Step 5 in the for loop implements the rectangular spectral clustering if the generative model is the SBM and rectangular spherical spectral clustering in the case of the DCBM.

For the SBM, the given input is comprised of , and the number of spectral components , which is assumed to be by default. Then, the parameter estimation proceeds as follows:

|  |  |  |
| --- | --- | --- |
| **Algorithm 3.2.2 Rectangular Spectral Clustering** | | |
| 1. | **Procedure RectSpectClust()** ▷ d = # right singular values of A | |
| 2. |  | **Let** be the matrix consisting of right singular vectors of |
| 3. |  | **Apply** k-means with clusters to rows of |
| 4. |  | **Return** |

also corresponds to the rank of , and its default choice is based on rank

With in hand, one obtains as follows. An accurate estimation of and requires a sufficient presence of all communities in the training set, i.e., a balanced training/test sets, which is met when is not too big.

## Edge Cross-Validation (ECV)

Similar to NCV, ECV is another algorithm based on cross-validation of network data for model selection and parameter tuning, and splits data at the node-pair level. However, unlike the authors of NCV, Li et al. (2019) do not assume a particular generative model; rather, the authors assume that the node probability matrix is of low rank, which is a widely-applicable assumption for network models. Once data is split, ECV applies low rank matrix completion to the low rank matrix before fitting a model. ECV can be applied to directed and undirected graphs, as well as binary and weighted networks.

### The Algorithm

Input: An adjacency matrix , a loss function , a set of candidate models or tuning parameter values, the training proportion , and the number of replications .

|  |  |  |  |
| --- | --- | --- | --- |
| **Algorithm 3.4.1 ECV** | | | |
| 1. | **Procedure ()** | | |
| 2. |  | **for** **do** | |
| 3. |  |  | Randomly choose a subset of node pairs with probability , selecting if is selected, for . |
|  |  |  | for do |
| 4. |  |  | Apply a low-rank matrix completion to to obtain with rank constraint K |
| 5. |  |  | Execute spectral clustering on to get SBM membership vector and |
|  |  |  | spherical spectral clustering to get DCBM membership vector . |
| 6. |  |  | Estimate probability matrices and using the membership vectors. |
| 7. |  |  | Compute the loss on held-out set and . |
| 8. |  | **Let** | |
| 9. |  | **Return** | |

### Discussion

The node-pair splitting procedure takes place in Step 3. In Step 4, while any of computationally efficient low-rank matrix completion procedure can be applied to to obtain a full matrix , the authors suggest the following singular value thresholding procedure:

where is rank truncated singular value decomposition of . The truncation is implemented by replacing the last of eigenvalues sorted in decreasing order with 0s.

In Step 9, of the returned values, is the best model that the algorithm has chosen with indicating SBM and indicating DCBM.

As long as the expected node degree is and , the estimator is consistent, that is, .

## Minimum Description Length (MDL)

### Preliminaries

The MDL principle states that the optimal model is one that requires the least amount of information to describe it, i.e., the information entropy. Rosvall et al. (2007) applied the MDL principle to detect communities, and Peixoto (2013) extended its use to estimate that can be arbitrarily large up to the detectability threshold of with agglomerative multilevel Markov chain Monte Carlo (MCMC) algorithm.

We follow notations as laid out in Peixoto (2013) below. The entropy can be expressed as

for SBM

for DCBM

where is the number of edges in the network, is the number of nodes in community , is the number of nodes whose degree is , and . Let be the entropy for either SBM or DCBM case. When selecting only on , the optimal solution is the trivial .

Hence, to avoid the overfitting, Peixoto (2013) incorporates the following notion of description length to be minimized in optimization problem of finding :

where for SBMs, and . For DCBMs, , where is the proportion of nodes that have degree .

Clearly, the simplest model is . Then, the tradeoff between complexity and description length as compared to the simplest case by

where , and , and and .

### Algorithm

|  |  |  |
| --- | --- | --- |
| **Algorithm 8: MDL** | | |
| 1. | **Procedure MDL-MCMC ()** ▷ B = initial value for | |
| 2. |  | for do |
| 3. |  | Initialize by randomly assigning nodes into B non-empty communities |
| 4. |  | for do▷ t = community of random neighbor of node |
| 5. |  | With probability , move node in community to |
| 6. |  | Compute using Fibonacci Search algorithm |
| 7. |  | If , Return 1; Otherwise Return |

### Discussion

When , the extent to which the extracted information the model with can represent the data is outweighed by the model complexity. Hence, in this case, we would choose the simple Erdös-Réyni model (). The detectability of is thus given by the condition

# Data, Variables, and Statistical Methods

The algorithms and its variants discussed in Section 4 were tested through extensive simulations to evaluate their performance with regard to accuracy and speed in estimating . Degree-corrected stochastic block models (DCSBM) were used as generative models for network data for use in simulations to maximize the extent to which the data realistically represent the real-world network data. Since each of the algorithms has been shown in respective papers to be effective when the data are dense (i.e., the average degree is in the order of the network size *n*), sparse network data (average degree is some multiple of a constant) were used instead to best tell apart the performances of the algorithms. Then, each of the algorithms discussed in Section 4 is tested on the data set, and various window lengths of adjacency matrices were tested to determine the optimal value for use in the dynamic changepoint detection algorithm.

In designing simulations, the following combinations of parameter values were used:

1. Number of nodes,
2. Number of iterations per simulation = 5
3. Community membership vector
4. The true number of communities
5. Number of discrete time snapshots

where

,

# Simulation Results and Discussion

As discussed in the previous section, the network data used in the simulations were sparse, with the average degree of less than 1 for a single adjacency matrix, and 1.5 for a cumulative sum. Figure 1 below shows the average degree by layer for the cumulative sum metric:

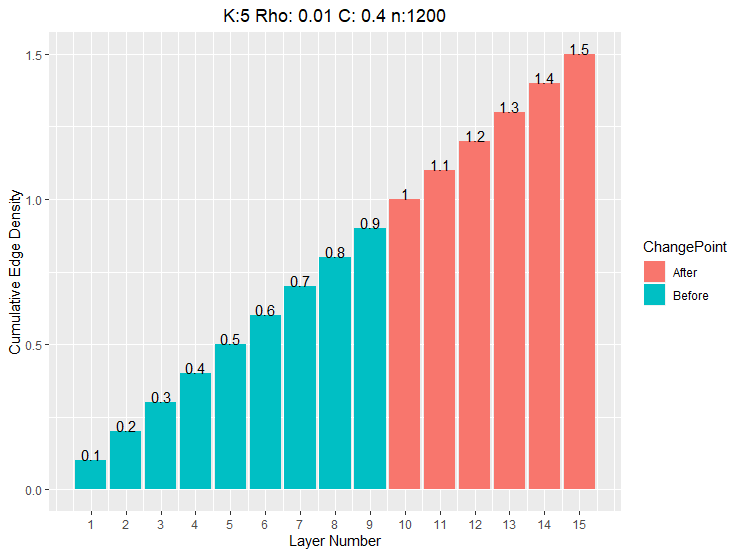


Figure 1

Each algorithm was tested for accuracy using cumulative sum of adjacency matrices. While all of the algorithms showed accuracy results with a delay of three layers, the Bethe Hessian operator displayed the most responsive metric, as shown below:

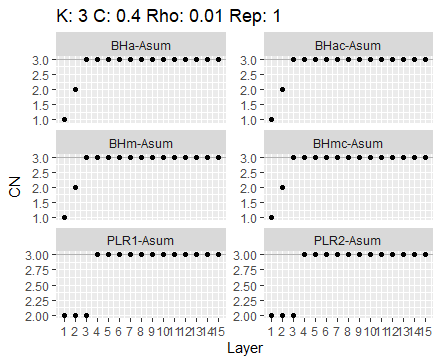


Figure 2

Then, changepoints were introduced into the time series of networks. In Figure 3, the changepoint occurred at Layer 9, indicated by a red vertical line. As is obvious in the charts, the cumulative sum method shows a significant delay in detecting the changepoint, since its estimate of the community size does not change until several time periods after changepoint occurs.

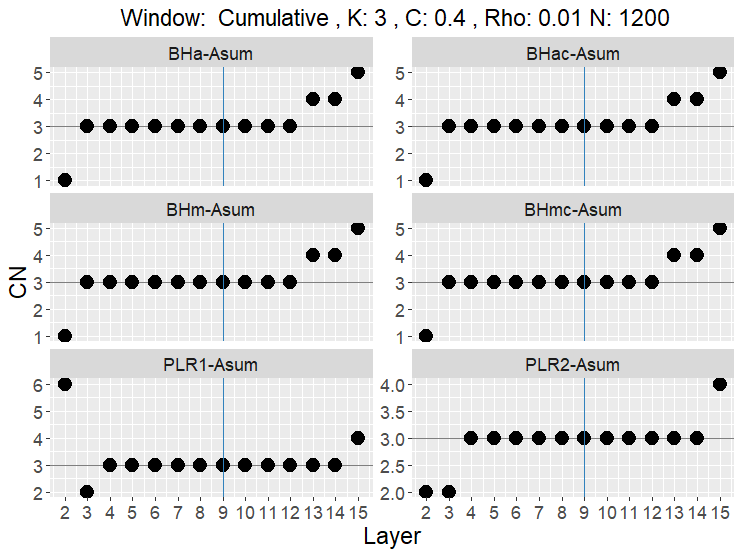


Figure 4

However, reducing the window length improves the algorithms’ ability to detect the changepoint much faster; in fact, with the window length of 4, there is a delay of just one period, as shown below:

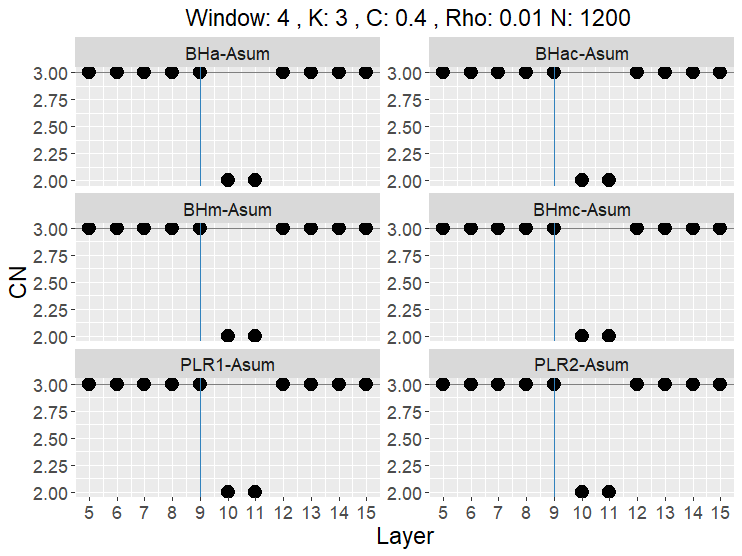


Figure 5

Reducing the window length further to three allows for even faster detection, but at the expense of more variability in the estimated community size.

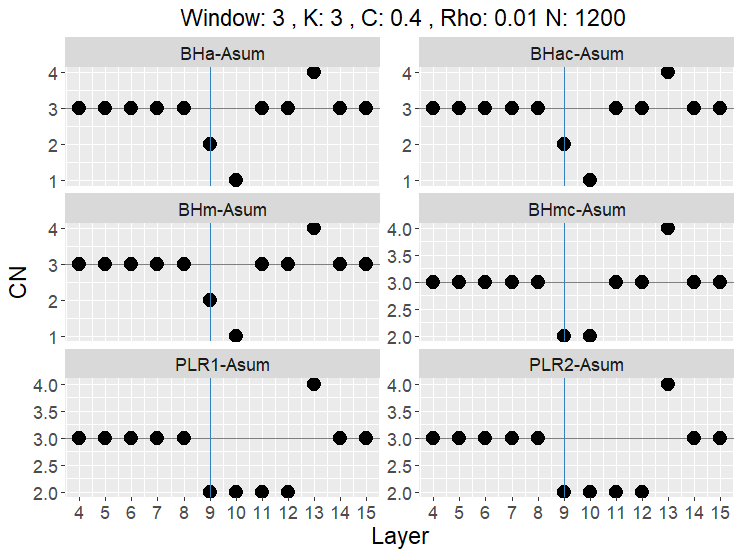


Figure 6

As expected per theoretical results in [Abbe et al 2014], when the sparsity of the graph dips below the detection threshold, the accuracy of the algorithms declines noticeably. Surprisingly, however, it is noteworthy that while the community size is not correctly estimated, the changepoint detection still occurs at the right place with the window length of four:

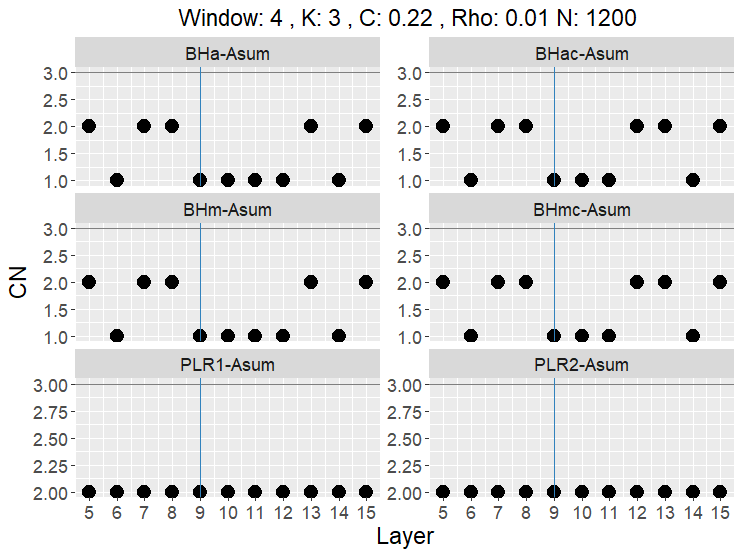


Figure 7

Based on these preliminary simulation results, the Bethe Hessian operator is chosen for the dynamic network changepoint detection phase of the project.

Next, we discuss the simulation results of dynamic network changepoint detection with 20 iterations per trial. First, it is noted that before the changepoint, the number of negative eigenvalues (which correspond to the number of communities in the Bethe Hessian operator) is three, with the smallest of the three in magnitude having a clear separation from zero. This is demonstrated in Figure 8 below:

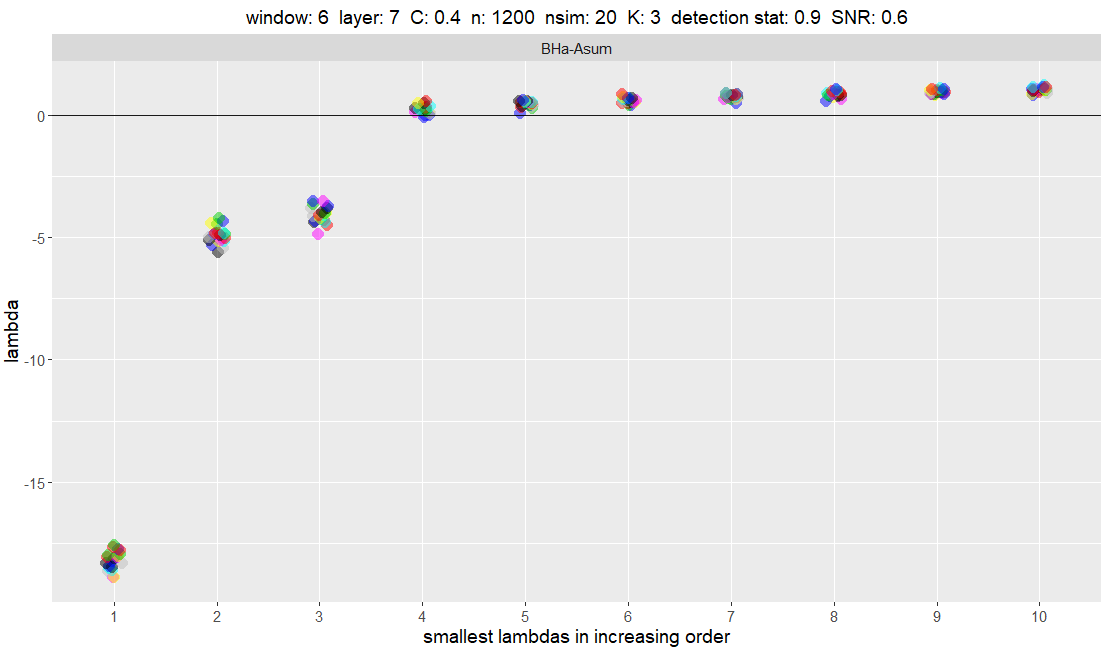


Figure 8

However, at the changepoint, the number of negative eigenvalues is still three, but the separation distance has decreased with the smallest of the three in magnitude much closer to zero than before.

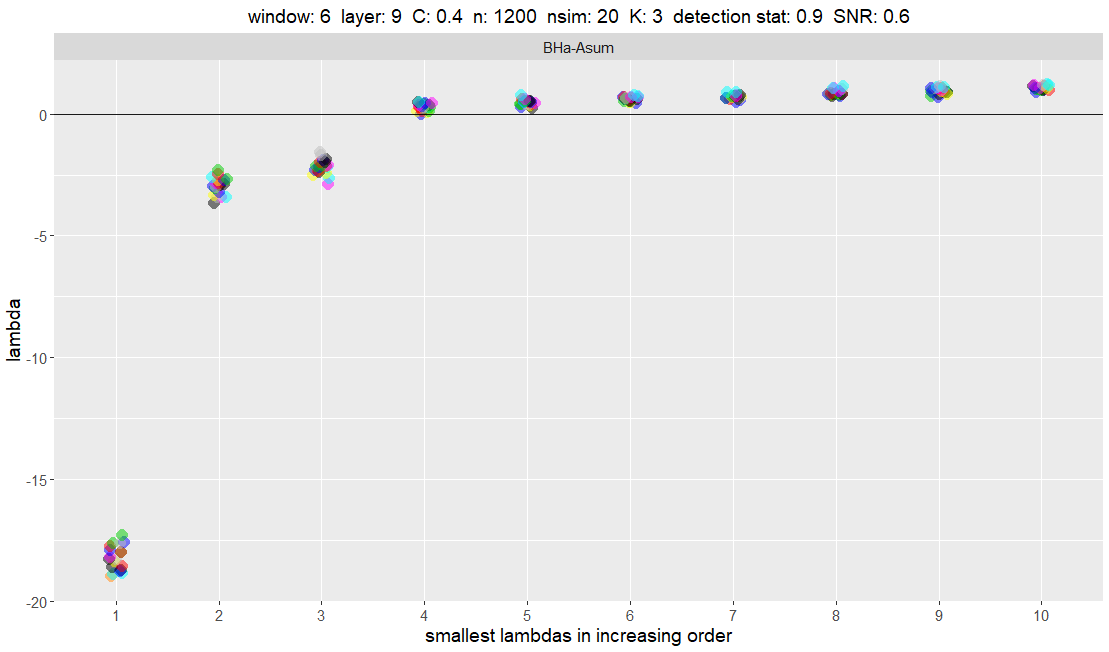


Figure 9

After the changepoint, the separation of the third largest eigenvalue in modulus is not noticeably different from zero, as shown below:

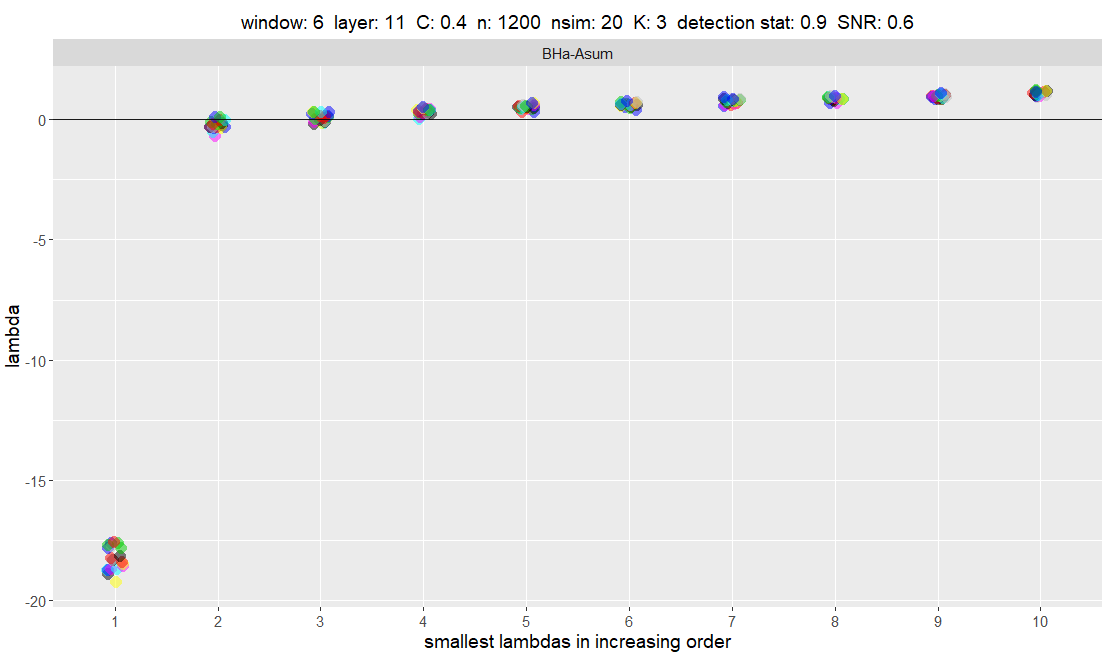


Figure 10

Using smaller window sizes appears to improve the speed at which the changepoint is detected, as the comparison chart below demonstrates:



Figure 11

These foregoing observations naturally lead to the formulation of the changepoint detection algorithm discussed in the next section.

# Dynamic Network Changepoint Detection Algorithm

## Algorithm

Below is the polynomial-time algorithm for detecting changepoint in dynamic networks. First, we outline the inputs and the expected outputs after each iteration, followed by the algorithm along with the time complexity for each step.

**Online Changepoint Detection Algorithm**

Input: (1) window length

(2) Adjacency matrices

(3) from the previous period

(4) tuning parameter

Output: (1)

(2) TRUE if T is a changepoint; FALSE otherwise

**-**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | |  |
| 1 | compute | |  |
| 2 | compute | |  |
| 3 | compute | |  |
| 4 | compute the eigenvalues of | |  |
| 5 | **sort** in non-decreasing order, | |  |
| 6 |  | |  |
| 7 | **if** | |  |
| 8 |  | **return** |  |
| 9 | **else** |  |  |
| 10 |  | **return** | 1 |

## Simulation Results

Next, we test the online detection algorithm outlined in the previous section. First, before the changepoint, we see that the detection threshold clearly holds, separating the three informative eigenvalues from the rest.

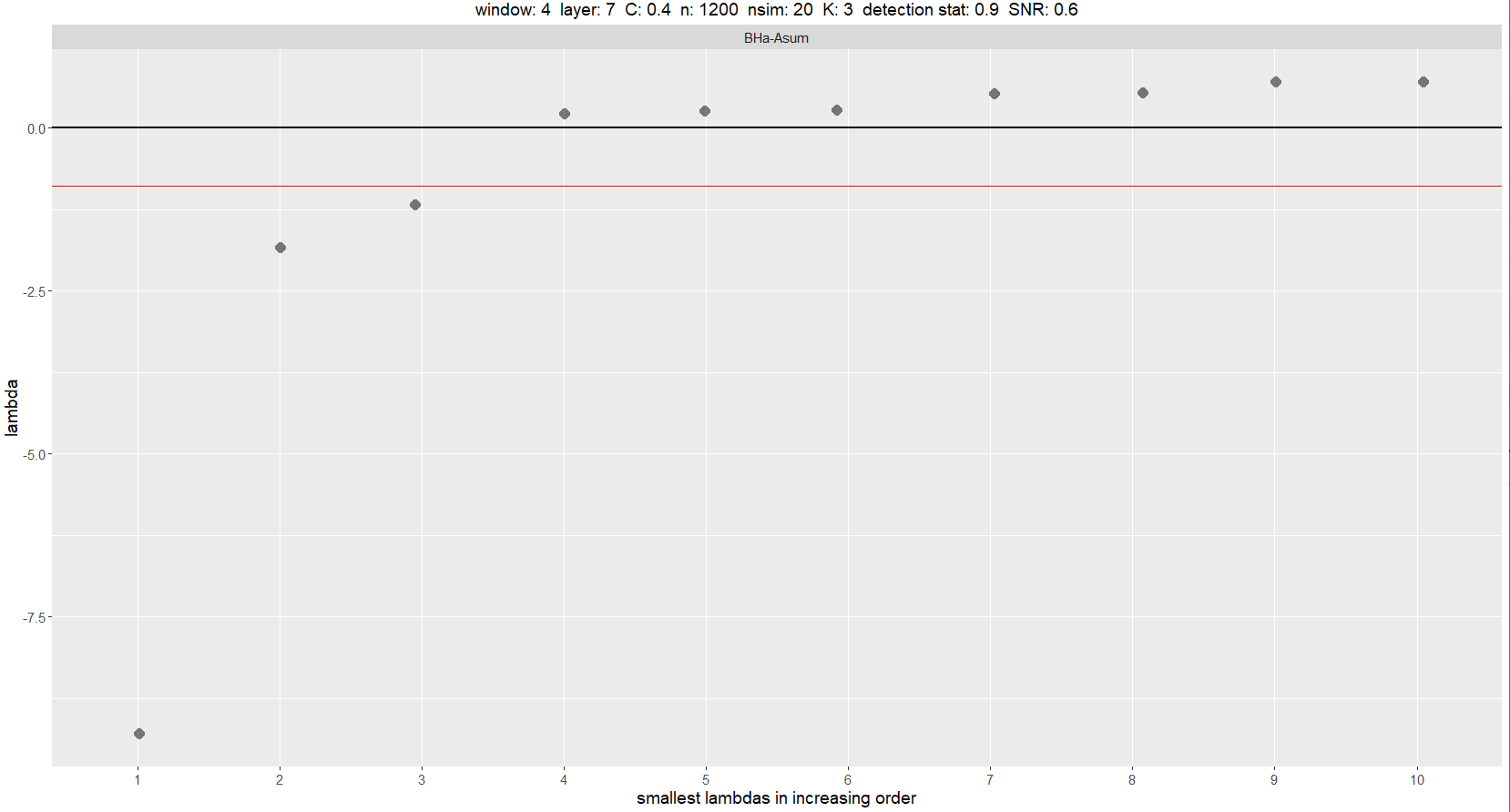


Figure 12

However, at the changepoint, we see that the two smaller eigenvalues have moved within the threshold, while still below the zero line, which would trigger the algorithm to return a TRUE indicating a changepoint.

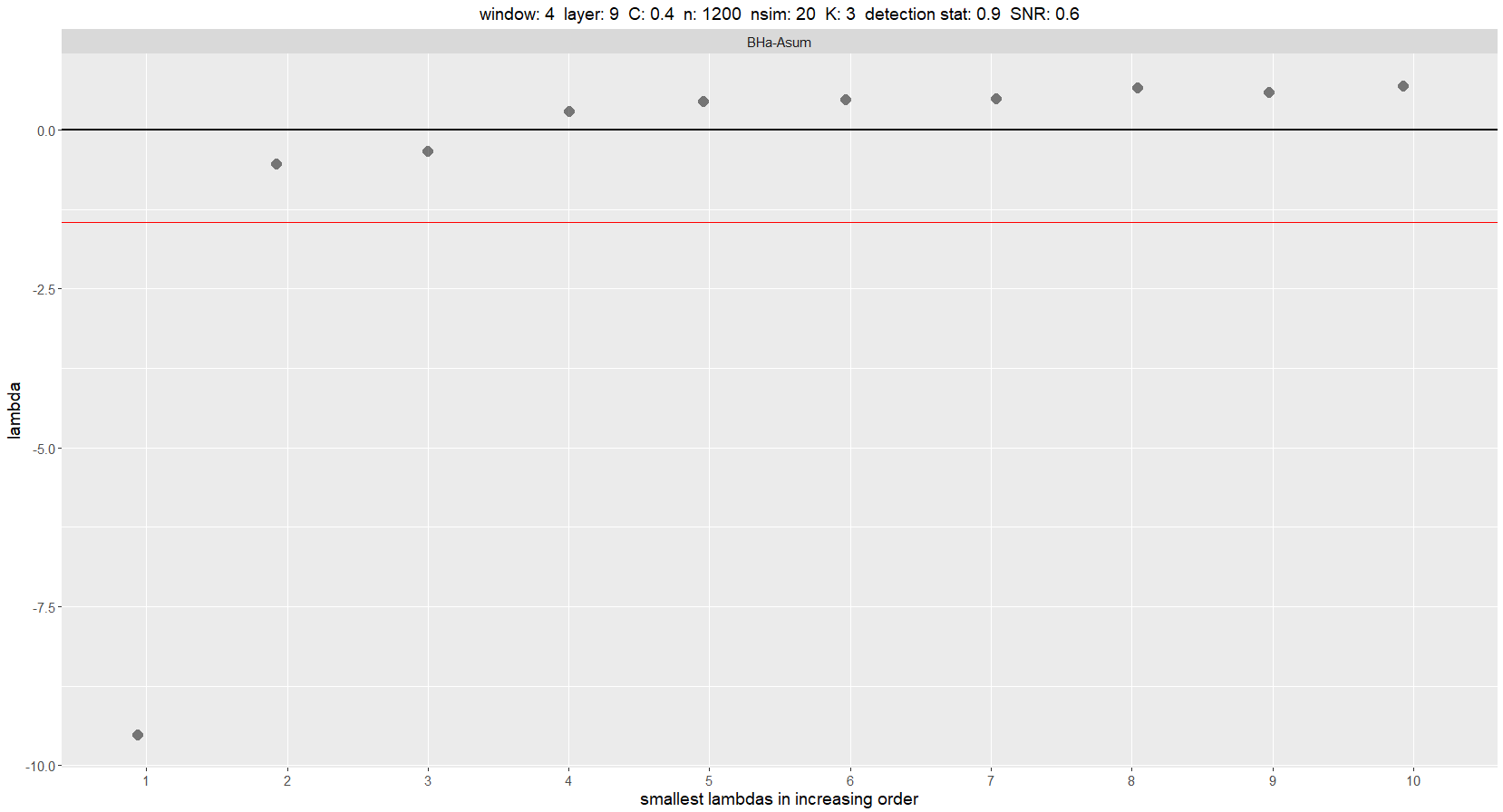
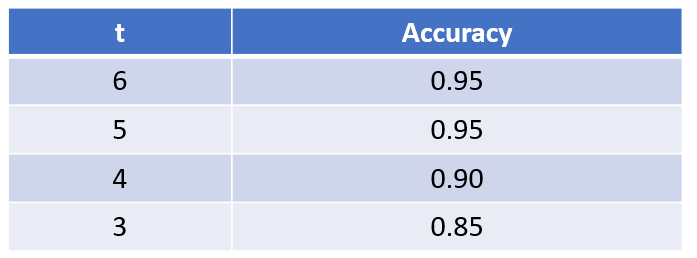


Figure 13

Compiling 20 simulation runs, each with 20 iterations, suggests that the optimal tuning parameter is 5.



# Conclusion and Next Steps

In this paper, we have studied various algorithms to detect the community size in networks generated by the degree-corrected stochastic block model and the changepoint in dynamic networks. In generating the simulation network data, we have assumed a regime where the community size stays constant while the node labels change over time, where the online detection algorithm based on the Bethe Hessian operator has been shown to be effective with minimal delay. For future research, it would be informative to study regimes where both the community size (5, for instance) and the labels change over time.

# References

Abbe, E. and Sandon, C. 2014. Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic BP, and the information-computation gap. 2016. *Proc. of ISIT.*

Bickel, P. and P. Sarkar. 2016. Hypothesis testing for automated community detection in networks. *Journal of Royal Statistical Society* 78, Part I, 253-273.

Bickel, P. and Y.X. Wang. 2017. Likelihood-based model selection for stochastic block models. *The Annals of Statistics* 45, 500-528.

Bordenave, C., M. Lelarge, and L. Massoulié. 2015. Non-backtracking spectrum of random graphs. *The Annals of Probability* 46, 1-71.

Chen, K. and J. Lei. 2018. Network Cross-Validation for Determining the Number of Communities in Network Data. *Journal of the American Statistical Association* 113, 241-251.

Erdös, P. and A. Réyni. “On Random Graphs. I.” *Publicationes Mathematicae*. 6: 290-297.

He, X., J.A. Pittman, O.M. Rui, and D. Wu. 2017. Do social ties between external auditors and audit committee members affect audit quality? *The Accounting Review* 92(5): 61-87.

Hoff, P.D., A. Raftery, and M. Handcock. 2002. Latent space approaches to social network analysis. *Journal of the American Statistical Association* 97: 1090-1098.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5(2):109–137.

Karrer, B. and M.E.J. Newman. 2011. Stochastic blockmodels and community structure in networks. *Physical Review E* 83: 016107.

Larcker, D. F., E.C. So, C.Y. Wang. 2013. Boardroom centrality and firm performance. *Journal of Accounting & Economics* 55(2, 3): 225–250.

Larremore, D., A. Clauset, and A. Jacobs. 2014. Efficiently inferring community structure in bipartite networks. *Physical Review* 90, 012805.

Le, C. and E. Levina. 2015. Estimating the number of communities in networks by spectral methods. Unpublished paper, retrieved June 14, 2019, from the arXiv database at <https://arxiv.org/abs/1507.00827>.pdf.

Levina, E., T. Li, and J. Zhu. 2019. Network cross-validation by edge sampling. Unpublished paper, retrieved June 14, 2019, from the arXiv database at <https://arxiv.org/pdf/1612.04717.pdf>.

Ma, S., L. Su, and Y. Zhang. 2019. Determining the Number of Communities in Degree-corrected Stochastic Block Models. Unpublished paper, retrieved June 14, 2019, from the arXiv database at <https://arxiv.org/pdf/1809.01028.pdf>.

Nguyen, B.D. and Nielsen, K.M. 2010. The Value of Independent Directors: Evidence from Sudden Deaths. *Journal of Financial Economics* 98, 550-567.

Peel, L., Larremore, D.B., and A. Clauset. 2017. The ground truth about metadata and community detection in networks. *Science Advances* 3(5) e1602548.

Peixoto, T.P. 2013. Parsimonious module inference in large networks. *Physical Review Letters* 110: 148701.

Rosvall, M. and C. T. Bergstrom. 2008. Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences of the United States of America*. 105 (4) 1118-1123.

Saade, A., F. Krzakala, and L. Zdeborova. 2014. Spectral Clustering of Graphs with the Bethe Hessian. *Advances in Neural Information Processing Systems*.

Van Peteghem, M., Bruynseels, and L., Gaeremynck, A. 2018. Beyond diversity: A Tale of Faultlines and Frictions in the Board of Directors. *The Accounting Review* 93, no. 2, 339- 367.