Data Science and Engineering
https://doi.org/10.1007/s41019-019-0084-x

Regular Decomposition of Large Graphs: Foundation of a Sampling Approach to Stochastic Block Model Fitting

Hannu Reittu[1] · Ilkka Norros[2] · Tomi Räty[1] · Marianna Bolla[3] · Fülöp Bazsó[4]

Received: 12 March 2018 / Revised: 30 January 2019 / Accepted: 22 February 2019

Abstract

We analyze the performance of regular decomposition, a method for compression of large and dense graphs. This method is inspired by Szemerédi's regularity lemma (SRL), a generic structural result of large and dense graphs. In our method, stochastic block model (SBM) is used as a model in maximum likelihood fitting to find a regular structure similar to the one predicted by SRL. Another ingredient of our method is Rissanen's minimum description length principle (MDL). We consider scaling of algorithms to extremely large size of graphs by sampling a small subgraph. We continue our previous work on the subject by proving some experimentally found claims. Our theoretical setting does not assume that the graph is generated from a SBM. The task is to find a SBM that is optimal for modeling the given graph in the sense of MDL. This assumption matches with real-life situations when no random generative model is appropriate. Our aim is to show that regular decomposition is a viable and robust method for large graphs emerging, say, in Big Data area.

Keywords Community detection · Sampling · Consistency · Martingales

1 Introduction

In the conference paper [1] we conjectured the possibility of applying our regular decomposition algorithm [2] to very large graphs, for which the full adjacency information is not possible to process, using a sampling approach. In this paper we develop the corresponding theory. Using martingale techniques, we prove claims of the preceding paper and give precise conditions under which they are true. This method allows to abandon the customary assumption that the graph be generated by a SBM.

The so-called Big Data is a hot topic in science and applications. Revealing and understanding various relations embedded in such large data sets is of special interest. In mathematical terms, such relations form a huge graph. The size of the graph may be so big that sampling of subgraphs is the only practically feasible operation. Furthermore, part of the information may be inaccessible, faulty or missing. Our method suggests a way to overcome such hurdles in the case of dense data.

One example is the case of semantic relations between words in large corpora of natural language data. Relations can also exist between various data sets, forming large higher-order tensors, thus requiring integration of various data sets. In such a scenario, algorithms based on stochastic block models (SBMs, also known as generalized random graphs [3]; for

✉ Hannu Reittu
hannu.reittu@vtt.fi

Ilkka Norros
ilkka.norros@elisanet.fi

Tomi Räty
tomi.raty@vtt.fi

Marianna Bolla
marib@math.bme.hu

Fülöp Bazsó
bazso.fulop@wigner.mta.hu

[1] VTT Technical Research Centre of Finland Ltd., P.O. Box 02044, Espoo, Finland

[2] Department of Mathematics and Statistics, University of Helsinki, P.O. Box 64, 00014 Helsinki, Finland

[3] Department of Stochastics, Institute of Mathematics, Technical University of Budapest, P.O.Box 91, Budapest 1521, Hungary

[4] Department of Computational Sciences, Institute for Particle and Nuclear Physics, Wigner Research Centre for Physics, Hungarian Academy of Sciences, P.O. Box 49, Budapest 1525, Hungary

Published online: 07 March 2019

a review see, e.g., [4]) and their extensions are very attractive solutions, instead of simple clustering like k-means.

A strong result, less known among practitioners, is Szemerédi's regularity lemma (SRL) [5]. SRL in a way supports a SBM-based approach to large data analysis, because it proves the existence of a SBM-like structure for any large graph, and has generalizations to similar objects like hypergraphs. SRL has been a rich source in pure mathematics—it appears as a key element of important results in many fields. We think that in more practical research it is also good to be aware of the broad context of SRL as a guiding principle and a source of inspiration. To emphasize this point, we call this kind of approach to SBM and a concrete way to fit data to it regular decomposition (RD).

Traditionally, SBM has been used for the so-called community detection in networks, like partitioning a network into blocks that are internally well connected and almost isolated from each other. From the point of view of SRL, this is too restrictive. According to SRL, the relations between the groups are more informative than those inside the groups, traditionally emphasized in community detection. Regular decomposition takes both aspects into account on equal footage. In [6] a more general scope of spectral clustering is introduced: the discrepancy-based spectral clustering that favors regular structures.

The methodology of RD has been developed in our previous works, first introduced in [7] and later refined and extended in [1, 2, 8–10]. In [2] we used Rissanen's minimum description length principle (MDL) [11] in RD. In [10] we extend RD to the case of large and sparse graphs using graph distance matrix as a basis.

In the current paper, we restrict ourselves to the case of very large and dense simple graphs. Other cases like rectangular matrices with real entries can be probably treated in very similar manner using the approach described in [2]. We aim to show that large-scale block structures of such graphs can be learned from bounded size, usually quite small, samples. The block structure found from a sample can then be extended to the rest of the graph in just a linear time w.r.t. n, the number of nodes.

The block label of a given node can be found independently of all other nodes, provided a good enough sample was used. This allows parallel computations in RD algorithm. The labeling of a node requires only the information on links from that node to the nodes of the fixed sample. As a result, the labeling can be done independently in many processing cores sharing the same graph and the sample.

The revealed structure is useful for many applications like those that involve counting numbers of small subgraphs or finding motifs. It also helps in finding a comprehensive overall understanding of graphs when the graph size is far too large for complete mapping and visualization. We also introduce a new version of the RD algorithm for simple graphs that tolerates missing data and noise.

2 Related Work

Our theoretical description is partly overlapping with SBM literature using the apparatus of statistical physics, see, e.g., extensive works by T.P. Peixoto. In particular, the use of MDL to find the number of blocks has been studied earlier by Peixoto [12]. We think that such multitude of approaches is only beneficial since it brings in results and ideas of different fields and is understandable by a wider community of researchers.

Recently, impressive progress has been achieved in the theory of nonparametric statistical estimation of large and sparse graphs [13, 14]. Under certain conditions, latent models like the SBM can be found from samples of links, linear in number of nodes sizes. Our work could be seen as a special case when the target is to find a block structure from a large graph using a specific algorithm. We show that in the dense case, the sample size is just a finite constant, instead of a linear function of the number of nodes, needed in the sparse case. Moreover, the algorithm in [14] is much more complicated than ours and may be difficult to use in practice. In a recent mostly experimental work [10], we attempted to use regular decomposition in a sparse graph case using graph distance as similarity measure between nodes.

A clustering method that can find the clusters from large enough samples is called consistent. As stated in [15], there are quite few results in proving consistency of clustering methods. The consistency of classical k-means clustering was proven by Pollard [16].

von Luxburg et al. [15] proved consistency of spectral clustering in a special case. It was shown that under some mild conditions, the spectral clustering of sampled similarity matrix of data with two clusters will converge to the spectral clustering of the limit corresponding to the infinite data. In this respect the so-called normalized approach that uses normalized Laplacian (see Sect. 6.2) was found preferable. Spectral clustering can be seen as an alternative to regular decomposition, since it can reveal similar block structures in adjacency—or similarity matrices [17].

3 Regular Decomposition of Graphs and Matrices

Regular decomposition finds a structure that mimics the regular partitions promised by SRL. SRL says, roughly speaking, that the nodes of any large enough graph can be partitioned into a bounded number, k, of equally sized 'clusters,' and one small exceptional set, in such a way that links between most pairs of clusters look like those in a random bipartite graphs with independent links and with the link probability that equals to the link density.

SRL is significant for large and dense graphs. However, similar results hold also to many other cases and structures, see the references in [18].

In RD, we simply replace the random-like bipartite graphs of SRL by a set of truly random bipartite graphs and use it as a modeling space in the MDL theory. We disregard the requirement of having blocks of equal sizes and also model the internal links of regular clusters by a random graph. The next step is to find an optimal model that explains the graph in a most economic way using MDL [2].

In the case of a matrix with nonnegative entries, we replace random graph models with a kind of bipartite Poissonian block models: A matrix element $a_{i,j}$ between row i and column j is thought to correspond to a random multi-link between nodes i and j, with the number of links distributed as a Poisson random variable with mean $a_{i,j}$ (see [9]). The bipartition is formed by the sets of rows and columns, respectively. Then the resulting RD is very similar to the RD of binary graphs. This approach allows also the analysis of several interrelated data sets and corresponding graphs, for instance, using corresponding tensor models. In all cases, the RD algorithm is very compact, containing just few lines of code in high-level languages like Python, and uses standard matrix-algebraic operations.

In RD, the unlossy code that describes a simple graph $G = (V, E)$ with respect to a partition $\xi = \{A_1, \dots, A_k\}$ of V has a length at most (and, for large graphs, typically close to)

$$\begin{aligned}
L(G|\xi) &= L_1(G|\xi) + L_2(G|\xi) + L_3(G|\xi) \\
&\quad + L_4(G|\xi) + L_5(G|\xi), \\
L_1(G|\xi) &= \sum_{i=1}^{k} l^*(|A_i|), \\
L_2(G|\xi) &= \sum_{i=1}^{k} l^*\left(\binom{|A_i|}{2d(A_i)}\right) \\
&\quad + \sum_{i<j} l^*\left(|A_i||A_j|d(A_i, A_j)\right), \\
L_3(G|\xi) &= |V|H(\xi), \\
L_4(G|\xi) &= \sum_{i=1}^{k} \binom{|A_i|}{2} H(d(A_i)), \\
L_5(G|\xi) &= \sum_{i<j} |A_i||A_j|H(d(A_i, A_j)),
\end{aligned} \tag{1}$$

where $l^*(m) := \log(m) + \log\log(m) + \cdots$ is the coding length of an integer m and

$$H(p) = -p\log p - (1-p)\log(1-p)$$

is the entropy of the Bernoulli(p) distribution. A corresponding representation exists also for the coding length of a matrix, interpreted as an average of a Poissonian block model (see details in [2]).

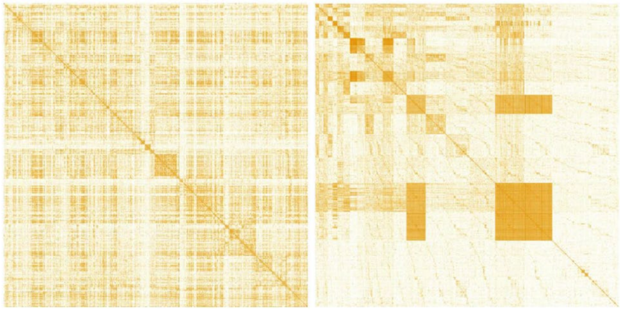

Fig. 1 Left: the similarity matrix between 4088 sentences from Charles Darwin's *Origin of Species*. Right: the same matrix with sentences reordered in 20 regular groups revealing a characteristic chessboard structure

Figure 1 shows a typical RD analysis result of a symmetric real matrix. The regular groups are obtained by permuting the rows (and accordingly the columns) of the left-hand side symmetric matrix and are presented on the right-hand side panel as homogeneous quadratic or rectangular blocks. For a non-symmetric rectangular matrix, the visual effect of RD is similar to chessboard.

3.1 Matrix Formulation of the RD Algorithm for Graphs

Definition 1 For a given graph G with n nodes, adjacency matrix A and a partition matrix $R \in \mathcal{R}_k$, we denote:

$$P_1(R) := R^T AR,$$

where $\cdot^T$ stands for matrix transpose. R is by definition a binary $n \times k$ matrix with rows that are indicators for regular group membership of the corresponding nodes. For instance, if the row i of R is (0, 1, 0, 0), this means that node i belongs to the regular group number 2.

The block sizes are the column sums of R and are denoted

$$n_\alpha := (R^T R)_{\alpha,\alpha}, \quad 1 \le \alpha \le k.$$

The number of links within each block and between block pairs (α, β) is uniquely defined by $P_1(R)$ and denoted as

$$e_{\alpha,\beta}(R) = \left(1 - \frac{1}{2}\delta_{\alpha,\beta}\right)(P_1(R))_{\alpha,\beta}.$$

Finally, we define the $k \times k$ link density matrix P:

$$\begin{aligned}
(P(R))_{\alpha,\alpha} &:= 1_{\{n_\alpha > 1\}} \frac{e_{\alpha,\beta}(R)}{\binom{n_\alpha}{2}}, \\
(P(R))_{\alpha,\beta} &:= \frac{e_{\alpha,\beta}(R)}{n_\alpha n_\beta}, \; \alpha \ne \beta, \\
&\quad \alpha, \beta \in \{1, 2, \dots, k\}
\end{aligned}$$

By (1), the length of the code l_k that uniquely describes the graph $G(A)$ corresponding to A, using the model (R, P), $R \in \mathcal{R}$, can be computed as follows:

Definition 2

$$\begin{aligned} l_k(G(A) \mid R) := & \sum_{1 \le i < j \le k} n_i n_j H((P(R))_{i,j}) \\ & + \sum_{1 \le i \le k} \binom{n_i}{2} H((P(R))_{i,i}) \\ & + l_k(R), \end{aligned}$$

where

$$l_k(R) = \sum_{1 \le i \le k} n_i H(n_i/n) + \sum_{1 \le i \le j \le k} l^*(e_{i,j}(R))$$

presents the length of a code describing the model (the partition and the link densities).

The two-part MDL program of finding the optimal model, denoted as R_{k^*}, can now be written as:

$$(k^*, R_{k^*}) := \arg \min_{1 \le k \le n} \min_{R \in \mathcal{R}_k} l_k(G(A) \mid R \in \mathcal{R}_k) \tag{2}$$

To solve this program approximately, we use the following greedy algorithm.

Algorithm 1 Greedy Two-part MDL

```
Input: $G = G(A)$ is a simple graph of size $n$.
Output: $(k^*, R_{k^*} \in \mathcal{R}_{k^*}), k^* \in \{1, 2, \dots, n\}$, such that the two-part code for G is approximately equal to the shortest possible for all possible block models with number of blocks in the range from 1 to n.
Start: $k = 1$, $l^* = \infty$, $R \in \mathcal{R}_1 = \{\mathbf{1}\}$, $k^* = 1$, where $\mathbf{1}$ is the $n \times 1$ matrix with all elements equal to 1.
1. Find
   $\hat{R}_k(G) := \arg \min_{R \in \mathcal{R}_k} l_k(G \mid R)$
   using subroutine ARGMAX k (Algorithm 2).
2. Compute $l_k(G) = \lceil l_k(G \mid \hat{R}(G)) \rceil + l_k(\hat{R}(G))$
3. If $l_k(G) < l^*$ then $l^* = l_k(G)$, $R_{k^*} = \hat{R}_k(G)$ , $k^* = k$
4. $k = k + 1$
5. If $k > n$, Print $(R_{k^*}, k^*)$ and STOP the program.
6. GoTo 1.
```

Definition 3 A mapping $\Phi : \mathcal{R}_k \to \mathcal{R}_k$ is defined as follows. First, define the following matrices element-wise:

$$(LogP(R))_{\alpha,\beta} := \log(P(R)_{\alpha,\beta})$$

$$(Log[1 - P(R)])_{\alpha,\beta} := \log[1 - P(R)_{\alpha,\beta}],$$

$$(1 - A)_{\alpha,\beta} := 1 - A_{\alpha,\beta},$$

$$L(R) := -AR(LogP(R))^T - (1 - A)RLog[1 - P(R)],$$

where we set $\log 0 := 0$ since it appears only in the combination $0 \log 0 = 0$ in matrix L, $\alpha, \beta \in \{1, 2, \dots, k\}$,

$$\beta(i, R) := \inf\{\beta : \beta = \arg \min_{1 \le \alpha \le k} (L(R))_{i,\alpha}\}, 1 \le i \le n,$$

and finally define the matrix function $\Phi(\cdot)$ element-wise on argument R:

$$\Phi(R)_{i,\alpha} = \delta_{\alpha,\beta(i,R)}, i \in \{1, 2, \dots, n\}.$$

The mapping $\Phi(R)$ defines a greedy optimization of partition R, where each node is replaced to a new block independently of all other placements, hence the term 'greedy algorithm.'

Algorithm 2 ARGMAX k

```
Algorithm for finding regular decomposition for fixed k.
Input: $A$: the adjacency matrix of a graph (an $n \times n$ symmetric binary matrix with zero trace); $N$: an integer (the number of iterations in the search of a global optimum); $k$: a positive integer $\le n$.
   Start: $m = 1$.
1. $i := 0$; generate a uniformly random element $R^i \in \mathcal{R}_k$.
2. If any of the column sums of $R^i$ is zero, GoTo 1, if not, then compute:
   $R^{i+1} := \Phi(R^i)$.
3. If $R^{i+1} \neq R^i$, set $i := i + 1$ and GoTo 2,
4. $R(m) := R^i$; $m = m + 1$;
   $l(m) := \sum_{i=1}^n \min_{1 \le \alpha \le k} (L(R(m)))_{i,\alpha}$.
5. If $m < N$, GoTo 1.
6. $M := \{m : l(m) \le l(i), i = 1, 2, \dots, N\}$; $m^* := \inf M$.
Output Solution: $R(m^*)$.
```

In **ARGMAX k** the outer loop 1–5 runs several (N) optimization rounds finding each time a local optimum, and finally, the best local optimum is the output of Algorithm 2. At each optimization round, the inner loop 2–3 improves the initial random partition until a fixed point is reached and no further improvements of the partition are possible. This fixed point is an indication that a local optimum is found.

For very large graphs, the program may not be solvable in the sense that it is not possible and reasonable to go through all possible values of $k \in \{1, 2, \dots, n\}$. One option is to limit the range of k. In case that no minimum is found, then use as an optimal choice the model found for the largest k within this range. Another option is to find the first minimum with smallest k and stop. When the graph is extremely large, it makes sense to use only a randomly sampled subgraph as an input—indeed, when $k^* << n$, a large-scale structure can be estimated from a sample as we shall show.

3.2 Regular Decomposition with Missing Link Information

Now assume that part of the link information is lost. We only consider the simplest case when the link information is lost uniformly randomly over all node pairs. We also exclude the possibility of a sparse representation of graphs, when only those node pairs that have links are listed, since this would lead to unsolvable ambiguity in our setting; indeed, if some node pair is not in the list, it could mean one of two cases: There is no link, or the link information is lost.

We formulate how the matrices used in the previous section are modified. The RD algorithms themselves are unaltered.

Denote by A the adjacency matrix with missing link values replaced by -1's. A is symmetric, and we also assume that we know its true dimension (the number of nodes). We put -1 on the diagonal of A, just for the convenience of further formulas. Define

$$D := \frac{A + |A|}{2},$$

where the absolute value of a matrix is defined element-wise. Thus, D has the same elements as A, except that the entries equal to -1 are replaced by 0's. Next define a 'mask matrix'

$$B := \frac{A - |A|}{2} + 1,$$

where $+1$ means element-wise addition of 1 to all elements of the matrix. That is, $b_{i,j} = 0$ if and only if $a_{i,j} = -1$, and $b_{i,j} = 1$ otherwise. In other words, B is the indicator of known link data in A. The matrix P_1 is now:

$$P_1(R) := R^T DR.$$

The number of observed links in a block indexed by (α, β) is

$$e_{\alpha,\beta} = \left(1 - \frac{1}{2}\delta_{\alpha,\beta}\right)(P_1(R))_{\alpha,\beta}.$$

We also need the following matrix, whose elements present the number of node pairs in block (α, β) that contain link data:

$$\begin{aligned}(N)_{\alpha,\beta} = n_{\alpha,\beta} &:= \left(1 - \frac{1}{2}\delta_{\alpha,\beta}\right) \textstyle\sum_{i,j} r_{i,\alpha} r_{j,\beta} b_{i,j} \\ &= \left(1 - \frac{1}{2}\delta_{\alpha,\beta}\right)(R^T BR)_{\alpha,\beta}.\end{aligned}$$

The P-matrix is defined as the link densities in the observed data:

$$(P(R))_{\alpha,\beta} := 1_{\{n_{\alpha,\beta}>0\}} \frac{e_{\alpha,\beta}}{n_{\alpha,\beta}}.$$

As stated above, we assume that the matrix elements of P are close to the corresponding actual link densities in the case that we would have complete link data. This also assumes that the graph and the blocks must have large size.

The cost function in Definition 2 remains unaltered. Now the P-matrix elements are only estimates based on observed data and not the exact ones. The block sizes are the true ones that are known since we assume that we know the actual number of nodes in the graph. The interpretation is that the contributions of missing data entries in a block are replaced by the average contribution found from the observed data of that block. It should be noted that similar scaling must be done in the RD algorithms that find the optimum of the cost function. For example, A should be replaced with D, and summing over elements of D should be rescaled so that the contribution is proportional to the true size of the corresponding block.

4 Sparse Sampling Scheme Formulation

Assume that a large graph has a 'regular structure' that minimizes the regular decomposition MDL program with k regular groups, $\xi := \{V_1, V_2, \dots, V_k\}$ having sizes $n_1, n_2, \dots, n_k$ and irreducible link density matrix with elements $0 < d_{i,j} < 1$. The number of nodes in G is denoted as N, and it is assumed that the relative sizes $r_i := n_i/N$ of the groups are such that $kr_i \geq c$, where c is a positive constant and not too small, say, $c = 1/10$. This condition means that none of the regular groups is very small. Such small groups would be 'artifacts' and are excluded by MDL criteria or by the practical RD algorithm that refuses forming too small groups. Here the aim of this condition is that when we make a uniformly random sample of n nodes, then all numbers $r_i n$, the expected number of sample nodes from each group, are big enough. On the other hand, this condition could probably be relaxed so that even small groups could be allowed. Their role in the classification problem we formulate is not decisive since the big groups are normally most decisive in this respect. However, we use this limitation in the sequel, for simplicity.

The optimal regular structure is, in particular, such that it is not possible to move even a single node from one group into another without some positive penalty in the following cost function (the log-likelihood part of it), the cost of placing a node v into group j

$$L_j(v) := \sum_{1 \leq i \leq k} (-e_i(v) \log d_{i,j} - (n_i - e_i(v)) \log(1 - d_{i,j})),$$

where $e_i(v)$ is the number of links between node v and nodes in group i. This means that the cost of changing a group membership is positive for all nodes:

$$v \in V_j \rightarrow L_s(v) - L_j(v) > 0, \quad \forall s \neq j,$$

where $V_j \in \xi$ is a regular group. The minimal penalty is denoted by δ:

$$\delta = \min_{v \in V_j, s \in V_s, s \neq j} (L_s(v) - L_j(v)) > 0.$$

Here we need a condition that δ is large. We require that the graph G and its optimal regular decomposition has the property:

$$\delta = cN, \quad c > 0.$$

This condition has similar motivation as the one for group sizes. We assume that N is very large; if δ is not comparable with N, then in a reasonable sample the corresponding difference becomes hopelessly small. We see in the sequel that the expected value of such a difference is large and it seems reasonable to place this condition. Also in the case of a SBM as the generator of G we would get such a condition on δ with exponential probability. We could relax this condition by assuming that it holds for most of the nodes v. For simplicity we use this condition as it is.

Consider making a uniformly random sample of n nodes of G, and retrieve the corresponding induced subgraph G_n. The claim is that if n is sufficiently large, then G_n has a regular structure that has almost the same densities and relative sizes of the groups and correct membership in regular groups. Further, if G_n is used as a classifier, the misclassification probability is small, provided the regular structure of G_n has large enough groups to allow good approximation of the parameters of the regular structure of G_n. Compare this claim with Lemma 2.3 from [18]:

Lemma 1 (Fox et al. [18]) *Let X and Y be vertex subsets of a graph G. Let $X' \subseteq X$ and $Y' \subseteq Y$ be picked uniformly at random with $|X'| = |Y'| = k$. Then*

$$\mathbf{P}(|d(X' - Y') - d(X, Y)| < \delta) \geq 1 - 2e^{-\delta^2 k/4}.$$

The proof is based on Azuma's equality for the edge exposure martingale. In our case we use a similar idea for the proofs.

5 Sampled Cost Function Estimation

First we show that in expectation the sampled cost function resolves well the correct optimum. Next, we study a harder problem of estimating the probability of a deviation of the randomly sampled cost function from its expected value. Finally, it is shown that with exponential probability, in n, the sampled cost function is able to resolve the correct block structure.

5.1 Expectation of the Sampled Cost Function

The cost function of a node $v \in V_i$ is just the length of the code (say, in number of bits) that describes the links of that node in G using the regular structure as the link probability distribution function. Select a node v uniformly at random. Then the cost function is a random variable. If $v \in V_i$, the cost function of placing $v \in V_\alpha$ has the value

$$L_{i,\alpha}(v) := \sum_{j \in \{1,2,\ldots,k\}} \big(-e_j(v) \log d_{j,\alpha} \\ -(n_j - e_j(v)) \log(1 - d_{j,\alpha})\big),$$

where $e_j(v), (j = 1, 2, \ldots, k)$ are random variables, the other parameters being constants. The expectation of $e_j(v)$, conditionally that $v \in V_i$, is

$$\mathbf{E}(e_j(v) | v \in V_i) = n_j d_{i,j},$$

which follows from the definition of $d_{i,j}$ as an arithmetic average of the fraction of node pairs that have links between regular groups i and j, or inside a regular group i if $i = j$. As a result:

$$\begin{aligned} C_{i,\alpha} &:= \mathbf{E}(L_{i,\alpha} | v \in V_i) \\ &= n_j \sum_{j \in \{1,2,\ldots,k\}} \big(-d_{i,j} \log d_{j,\alpha} \\ &\quad -(1 - d_{i,j}) \log(1 - d_{j,\alpha})\big). \end{aligned}$$

Proposition 3 *Requiring that all groups are large and that for all pairs $(i, \alpha), i \neq \alpha$, there is at least one index j with $d_{i,j} \neq d_{j,\alpha}$, we have:*

$$\min_{\{i,\alpha : \alpha \neq i, i, \alpha \in \{1,2,\ldots,k\}\}} (C_{i,\alpha} - C_{i,i}) = cN, \quad c > 0,$$

where c is a constant.

Proof It is easy to check that in the above formula for $C_{i,\alpha}$, all terms in the sum, $-d_{i,j} \log d_{j,\alpha} - (1 - d_{i,j}) \log(1 - d_{j,\alpha})$, are nonnegative and each of them has an absolute minimum on argument $d_{j,\alpha}$ when the corresponding densities are equal, $d_{i,j} = d_{j,\alpha}$. Therefore, in $C_{i,\alpha} - C_{i,i}$ the coefficients of n_j, $-d_{i,j} \log d_{j,\alpha} - (1 - d_{i,j}) \log(1 - d_{j,\alpha})-(-d_{i,j} \log d_{i,j} - (1 - d_{i,j}) \log(1 - d_{i,j}))$, are nonnegative for all j, and, according to the condition, at least one of them is strictly positive, because the corresponding densities are not equal. Since all n_j are proportional to N, the claim follows. □

In general setting, the difference $L_{i,\alpha}(v) - L_{i,i}(v)$ is not necessarily as large as the expected value for all nodes v. The only thing that can be guaranteed is that the value is positive (definition of an optimum). In the case of a true stochastic block model as a generator of the graph G, almost all differences $L_{i,\alpha}(v) - L_{i,i}(v)$ are of the order of the expectation (positive and linear in N). Sub-linear terms could be called 'outliers,' since there is not much difference in terms of cost, in whichever group we place the corresponding nodes.

We now simply postulate that there are no outliers corresponding to the optimal regular structure we are dealing with. This means that

$$\forall v \in V : L_{i,\alpha}(v) - L_{i,i}(v) \geq \delta = cN, c > 0.$$

Consider a uniformly random sample of n nodes from V, denoted as $\hat{V} \subset V$. Denote by $\hat{\xi}$ (we shall use systematically 'hat,' $\hat{}$, over a symbol of a variable to denote a corresponding sampled value) the partition of V_n that is a subpartition of ξ, meaning that two nodes are in the same part of $\hat{\xi}$ if and only if they are in the same part of ξ. Assume that all parts of $\hat{\xi}$ are 'large,' meaning that their relative sizes are close to those in ξ. The probability of this condition can be made close to one, taking n large enough. This probability will be estimated later on, see Proposition 9. Take a uniformly random node $v \in V$. Consider the following classifier based on $\hat{\xi}$:

$$\begin{aligned} & v \to V_{\alpha *}, \\ & \alpha^* = \arg \min_{\alpha \in \{1,2,\ldots,k\}} \hat{L}_{i,\alpha}(v), \\ \hat{L}_{i,\alpha}(v) := & \sum_{j \in \{1,2,\ldots,k\}} \big(-\hat{e}_j(v) \log d_{j,\alpha} \\ & -(\hat{n}_j - \hat{e}_j(v)) \log(1 - d_{j,\alpha}) \big), \end{aligned} \tag{3}$$

where $\hat{e}_j(v)$ is the number of links joining group j of $\hat{\xi}$ and node v. The sizes of regulars groups in $\hat{\xi}$ are denoted as $\hat{n}_i$, and the groups of $\hat{\xi}$ are denoted as $\hat{V}_i$ for every $i = 1, 2, \ldots, k$, using same indexing as in ξ. Note that here we use the link densities between the regular groups of the large graph. Later, we shall show that the link densities in $\hat{\xi}$ are close enough with a probability close to 1. This is another claim that needs to be justified (see Sect. 5.3).

Obviously, $\hat{e}_j$ is a random variable with a hypergeometric distribution. If we call 'success' a case when a sampled node from $\hat{V}_j$ has a link to node v, then there are $e_j(v)$ favorable outcomes out of n_i possible choices. According to a well-known result, conditionally on sample size, $\hat{n}_j$, the expectation is

$$\mathbf{E}(\hat{e}_j(v)|\hat{n}_j) = e_j(v)\frac{\hat{n}_j}{N}.$$

Since $\hat{n}_i, i = 1, 2, \ldots, k$, are distributed according to a multivariate hypergeometric distribution, we get easily

$$\mathbf{E}\hat{n}_j = r_j n,$$

and we have

$$\mathbf{E}\hat{e}_j(v) = \frac{e_j(v)}{n_j} r_j n = e_j(v)\frac{n}{N}.$$

We have shown the following:

Proposition 4

$$\mathbf{E}\hat{L}_{i,\alpha}(v) = \frac{n}{N} L_{i,\alpha}(v).$$

In other words, the expectation of a sampled cost function is just a rescaled version of the cost function for the whole graph. Using this result, we see:

$$\begin{aligned} & \forall v \in V_i,\ \alpha \neq i : \\ & \mathbf{E}(\hat{L}_{i,\alpha}(v) - \hat{L}_{i,i}(v)) = \frac{n}{N}(L_{i,\alpha}(v) - L_{i,i}(v)) \\ & \qquad \geq \frac{n}{N} cN = cn, \quad c > 0. \end{aligned}$$

5.2 Deviations of the Sampled Block Sizes from the Expected Values

Assume that all link densities fulfill $0 < d_{i,j} < 1$. Excluding cases with zero or one densities are not limitations, since in both cases the sampled structures are deterministic (for density 1, all sampled pairs have links, and in the other case, no pairs have links).

When $v \in V_i$, we can write

$$\hat{L}_{i,\alpha}(v) = \sum_{1 \leq j \leq k} (a_{j,\alpha}\hat{e}_j(v) + b_{j,\alpha}(\hat{n}_{j,\alpha} - \hat{e}_{j,\alpha})),$$

where

$$a_{j,\alpha} := -\log d_{j,\alpha}, \quad b_{j,\alpha} := -\log(1 - d_{j,\alpha})$$

are all positive constants. Denote a normalized version as

$$\hat{Y}_{i,\alpha}(v) := \frac{\hat{L}_{i,\alpha}(v)}{c_\alpha},$$

where

$$c_\alpha := \max_{1 \leq j \leq k} |a_{j,\alpha} - b_{j,\alpha}|.$$

This normalization guarantees that if one of the arguments of $\hat{L}_{i,\alpha}(v)$, $\hat{e}_j$, changes by an amount ϵ with $|\epsilon| \leq 1$, then the absolute value of the difference is upper-bounded by 1:

$$|\Delta \hat{Y}_{i,\alpha}| = \frac{|\epsilon(a_{j,\alpha} - b_{j,\alpha})|}{c_\alpha} \leq |\epsilon| \leq 1.$$

Define the following random process with discrete time $t \in \{0, 1, 2, \ldots, n\}$. Choose $\hat{n}_j$ nodes uniformly at random from V_j, without replacement. Denote the sampled nodes in order they were drawn: $(v_j(1), \dots, v_j(\hat{n}_j))) = \hat{V}_j$ and similarly for all $1 \le j \le k$. Then choose $v \in V_i$ uniformly at random from the remaining nodes. Correspondingly, define the random variables $x_j(1), \dots, x_j(\hat{n}_j)$, so that $x_j(i) = 1$ if $(v_j(i), v)$ is a link in G and otherwise $x_j(i) = 0$.

Obviously, we have the following conditional probabilities:

$$\mathbf{P}(x_j(t) = 1 | x_j(t-1), x_j(t-2), \dots, x_j(1)) \\ = \frac{e_j(v) - \sum_{1 \le s \le t-1} x_j(s)}{n_j - (t-1)}.$$

Define the following vertex exposure process, corresponding to $x_j(t)$:

$$E_j(0; v) = \mathbf{E}\hat{e}_j(v) = \hat{n}_j \frac{e_j(v)}{n_j},$$

$E_j(t; v)$ is the expectation of $\hat{e}_j$ after $x_j(1), \dots, x_j(t)$ are known, and the contribution of unexposed variables $x_j(t+1), \dots, x_j(\hat{n}_j)$ is replaced by the expected value. Using the formulas for hypergeometric expectations, we get the following:

$$E_j(t; v) = \sum_{1 \le s \le t} x_j(s) + \frac{(\hat{n}_j - t)(e_j(v) - \sum_{1 \le s \le t} x_j(s))}{n_j - t}.$$

Proposition 5 *$E_j(\cdot)$ is a martingale.*

Proof We need to show that

$$\mathbf{E}(E_j(t; v) | E_j(t-1; v), \dots E_j(0, v)) = E_j(t-1; v).$$

It is well known that such an exposure process is always a martingale. However, we check this condition as a 'sanity check' of our construction.

$$\begin{aligned} &\mathbf{E}(E_j(t; v) | E_j(t-1; v), \dots E_j(0; v)) \\ &\quad = \mathbf{E}(x_j(t) | x_j(t-1), \dots, x_j(1)) + \sum_{1 \le s \le t-1} x_j(s) \\ &\qquad + (\hat{n}_j - t)\{e_j(v) - \sum_{1 \le s \le t-1} x_j(s) \\ &\qquad - \mathbf{E}(x_j(t) | x_j(t-1), \dots, x_j(1)))\} / (n_j - t). \end{aligned}$$

Note that

$$\begin{aligned} &\mathbf{E}(x_j(t) | x_j(t-1), \dots, x_j(1)) \\ &\quad = \mathbf{P}(x_j(t) = 1 | x_j(t-1), x_j(t-2), \dots, x_j(1)) \\ &\quad = \frac{e_j(v) - \sum_{1 \le s \le t-1} x_j(s)}{n_j - (t-1)}. \end{aligned}$$

Denote $x = \hat{n}_j, s_m = \sum_{s=1}^m x_j(s), e_j(v) = e$ and $n_j = n$. In these terms, we can write

$$\begin{aligned} &E_j(t) = s_t + \frac{(x-t)(e-s_t)}{n-t}, \\ &\mathbf{E}(x_j(t) | x_j(t-1), \dots, x_j(1)) = \frac{e - s_{t-1}}{n-(t-1)}. \end{aligned}$$

Now,

$$\begin{aligned} &\mathbf{E}(E_j(t; v) | E_j(t-1; v), \dots E_j(0; v)) \\ &\quad = s_{t-1} + \frac{e - s_{t-1}}{n-(t-1)} + \frac{(x-t)\Big(e - s_{t-1} - \frac{e-s_{t-1}}{n-(t-1)})\Big)}{n-t} \\ &\quad = E_j(t-1; v) - \frac{(x-(t-1))(e-s_{t-1})}{n-(t-1)} + \frac{e-s_{t-1}}{n-(t-1)} \\ &\qquad + \frac{(x-t)\Big(e - s_{t-1} - \frac{e-s_{t-1}}{n-(t-1)})\Big)}{n-t} \\ &\quad = E_j(t-1; v) - \frac{(x-t))(e-s_{t-1})}{n-(t-1)} \\ &\qquad + \frac{(x-t)\Big(e - s_{t-1} - \frac{e-s_{t-1}}{n-(t-1)})\Big)}{n-t} \\ &\quad = E_j(t-1; v) + (x-t)(e-s_{t-1})\left(\frac{1}{n-t} - \frac{1}{n-(t-1)}\right) \\ &\qquad - \frac{(x-t)(e-s_{t-1})}{(n-t)(n-(t-1))} \\ &\quad = E_j(t-1; v) + \frac{(x-t)(e-s_{t-1})}{(n-t)(n-(t-1))} \\ &\qquad - \frac{(x-t)(e-s_{t-1})}{(n-t)(n-(t-1))} \\ &\quad = E_j(t-1; v). \end{aligned}$$

□

Let us construct the random process

$$X_{i,\alpha}(t; v), \quad v \in V,$$

where t is discrete time from 0 to n. First, define the following ramp functions for each $1 \le i \le k$:

$$\begin{aligned} \theta_i(t) :=\, &\theta\left(t - \sum_{j=1}^{i} \hat{n}_j\right)\left(\left(t - \sum_{j=1}^{i} \hat{n}_j\right)\right. \\ &\left.\times\, \theta\Bigg(\sum_{j=1}^{i+1} \hat{n}_j - t) + \hat{n}_i \theta\left(t - \sum_{j=1}^{i+1} \hat{n}_j\right)\right), \end{aligned}$$

where $\theta(\cdot)$ is the Heaviside step function, with agreement $\theta(0) = 0$, $\hat{n}_{k+1} = 0$ and $t \in \{0, 1, \dots, \sum_{j=1}^{k} \hat{n}_j\}$ (Fig. 2).

By definition,

$$X_{i,\alpha}(t; v) := \frac{1}{c_\alpha} \sum_{j=1}^{k} \big(-E_j(\theta_j(t); v) \log d_{j,\alpha} - (\hat{n}_j - E_j(\theta_j(t); v)) \log(1 - d_{j,\alpha})\big),$$

where we refer to the previous definitions of $E_j(\cdot)$, v and $\theta_j(\cdot)$ and $t = 0, 1, \dots, n$.

Proposition 6 *$X_{i,\alpha}(\cdot; v)$ is a martingale and*

$|X_{i,\alpha}(t; v) - X_{i,\alpha}(t-1; v)| \leq 1, \forall t : 1 \leq t \leq n.$

Proof At each discrete time, only one of the subprocesses $E_j(\cdot)$ is 'active.' By this, we mean that for any moment of time t only one subprocess $E_j(\cdot)$ is progressing, the others are either in the final state when all links to v are revealed, or are completely un-revealed, and the corresponding value equals the expectation. Therefore, Proposition 5 is sufficient to show the conditional expectation property. The normalization coefficient c_α guarantees the upper bound for the absolute values of step-wise changes of the process. □

As a result, we get the following:

Proposition 7 *For all $\alpha, i \in \{1, 2, \dots, k\}$:*

$$\mathbf{P}|(X_{i,\alpha}(n; v) - \mathbf{E}X_{i,\alpha}(n; v)| \leq t) \geq 1 - 2e^{-\frac{t^2}{2n}}.$$

Proof Use the standard Azuma's inequality (e.g., Theorem 2.25 in [19]). The inequality is true for all indices α and i since there is only one link exposure process, and for any of them the inequality is true with right-hand side that is independent of those indices. □

Next, define the following vector process. It corresponds to uniformly random sampling of n nodes from V. Its components correspond to the numbers of nodes in the classes of $\hat{\xi}$. It has $n + 1$ time steps and starts from all components equal to zero, and at $t = 1$, the value of ith component is $\mathbf{E}\hat{n}_i = nn_i/N$. After that the first node and its class in ξ, let it be i, are revealed, and the corresponding component is changed to the expected value of $\hat{n}_i$ conditionally to this information. And so on, until the process ends with all components having their final values, like $\hat{n}_i$. By normalizing this process with the coefficient $13\sqrt{k}$ such a process is denoted as $\mathbf{N}(t)$.

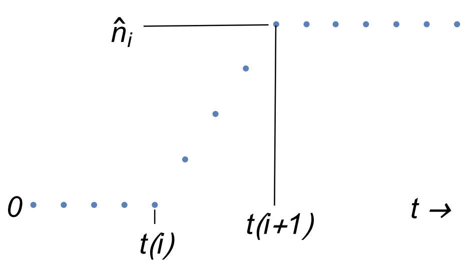

Fig. 2 Schematic graphics of $\theta_i(t)$; it ramps from zero to $\hat{n}_i$ within some time interval $[t(i), t(i+1)]$, with step size equal to 1, and outside the interval, remains constant. This interval corresponds to the one of span of process $E_i(\cdot)$

Proposition 8 *Assume that the graph size N is large enough and the sample size is small with $n \leq \sqrt{n_i}, \forall i$. Then $\mathbf{N}(\cdot)$ is a vector martingale with maximal increment of its Euclidean norm equal to 1.*

Proof The martingale property follows from the general character of the process as an exposure (see Proposition 5). It remains to check the step bound. Let $x_i(t)$ denote the increase in the number of nodes in class i at time step t. Then $x_i(t)$ equals 1 if the tth node happens to belong to class i and equals zero otherwise. Using the obvious multivariate hypergeometric distribution we see that the non-normalized ith component of the process at time t is

$$X_i(t) = \sum_{s=1}^{t} x_i(t) + \frac{(n-t)\big(n_i - \sum_{s=1}^{t}\big)}{N-t}.$$

Using the short-hand notations $x = n_i$, $s_t = \sum_{s=1}^{t} x_i(t)$, we write:

$$\begin{aligned}
&|X_i(t) - X_i(t-1)| \\
&\quad = \left| s_t - s_{t-1} + \frac{(n-t)(x-s_t)}{N-t} - \frac{(n-t+1)(x-s_{t-1})}{N-t+1} \right| \\
&\quad = \Bigg| s_t - s_{t-1} \\
&\qquad + \frac{(1+N-t)(s_{t-1} + (t-n)(s_t - s_{t-1}) - x))}{(N-t)(N-t+1)} \Bigg|.
\end{aligned}$$

We see that the dominating term (largest in absolute value) in the nominator is $-xN = -n_iN$, since we assume a large graph and much smaller sample size, and we can assume that the whole numerator is negative; thus, the upper bound of the absolute value of the numerator can be obtained by making the rest of terms as small as possible:

$$\begin{aligned}
&|(1+N-t)(s_{t-1} + (t-n)(s_t - s_{t-1}) - x))| \\
&\quad = |-xN \\
&\qquad + (1+N-t)\big(s_{t-1} + (t-n)(s_t - s_{t-1})\big) - x(1-t)| \\
&\quad \leq |-xN + (N+1)(-n) - x| \\
&\quad \leq xN + (N+1)n + x \leq 3xN,
\end{aligned}$$

where we used also $s_t - s_{t-1} \leq 1$, $0 \leq t \leq n$, and in the final upper bound we assume that n is much smaller than n_i or N. As a result we get:

$$|X_i(t) - X_i(t-1)| \le 1 + \frac{3n_iN}{(N-n)^2}.$$

Assuming that $n \le N/2$, we have

$$|X_i(t) - X_i(t-1)| \le 1 + \frac{12n_iN}{N^2} \le 13.$$

Finally, for the Euclidean norm we get:

$$||\mathbf{X}(t) - \mathbf{X}(t-1)||^2 = \sum_{i=1}^{k}(X_i(t) - X_i(t-1))^2 \le k13^2.$$

Thus, by defining $\mathbf{N}(t) = \frac{\mathbf{X}(t)}{13\sqrt{k}}$, we obtain

$$||\mathbf{N}(t) - \mathbf{N}(t-1)|| \le 1.$$

□

Using the Azuma inequality for a vector-valued martingale, given in [20], we get:

Proposition 9 *Under the same conditions as in Proposition 8, we have:*

$$\mathbf{P}\left(\max_{1\le i\le k} |\hat{n}_i - r_in| \le 13\sqrt{k}(n^{3/4}+1)\right) \ge 1 - 2e^{1-\sqrt{n}/2}.$$

Proof We just check the conditions of the theorem in [20] that are all fulfilled after centering the $\mathbf{N}(\cdot)$ to expected value at time $t = n$. The theorem then states:

$$\mathbf{P}(||\mathbf{N}(n) - \mathbf{E}\mathbf{N}(n)|| \ge a) < 2e^{1-(a-1)^2/2n}.$$

This is equivalent to

$$\mathbf{P}(||\mathbf{X}(n) - \mathbf{E}\mathbf{X}(n)|| \ge 13a\sqrt{k}) < 2e^{1-(a-1)^2/2n}.$$

The expectations of components are $\mathbf{E}X_i(n) = \mathbf{E}\hat{n}_i = r_in$. Since $||\mathbf{X}(n) - \mathbf{E}\mathbf{X}(n)|| \ge \max_{1\le i\le k} |\hat{n}_i - r_in|$, we get the claimed inequality by choosing $a = 1 + n^{3/4}$. □

5.3 Deviations of Sampled Link Densities from Their Expectations

Finally, we estimate the deviations of link densities between sampled groups, $\hat{d}_{i,j}$, from their expected values $d_{i,j}$, conditionally on sampled group sizes $\hat{n}_i$. We simply reuse Theorem 2.2. from [18].

Proposition 10 *Provided $r_in/2 \le \hat{n}_i \le 2r_in$, we have:*

$$\mathbf{P}\left\{\max_{1\le i\le j} |\hat{d}_{i,j} - d_{i,j}| \le \frac{1}{\sqrt[4]{n}}\right\} \ge 1 - k(k-1)\exp(-\sqrt{n}a),$$

where

$$a = \min_{i\ne j}\{(r_ir_j)^2/(32(r_i+r_j)^3), (r_i - 1/n)^2/r_i^3\}.$$

Proof Use Theorem 2.2 of [18] and reasoning therein for a given pair (i, j) and the conditions on sizes $r_in/2 \le \hat{n}_i \le 2r_in$, and finally the union bound to estimate that all densities are within the range. The coefficient a corresponds to the weakest exponent. □

5.4 Probability of Deviation of Sampled Cost Function from the Expectation

Using the results of previous sections we can formulate and prove the main result:

Theorem 1 *Provided $n \ge m$, $m = f^4$, then uniformly random sampling of n nodes and grouping of these nodes as in ξ has the property: If $v \in V$ is a random node from V_i, outside the sample, then for all $\alpha \ne i$:*

$$\mathbf{P}(\hat{L}_{i,\alpha}(v;\hat{D}) - \hat{L}_{i,i}(v;\hat{D}) > 0) \ge 1 - g\exp(-\phi\sqrt{n}),$$

where

$$\begin{aligned}
\hat{L}_{i,s}(v;\hat{D}) :=& \sum_{j\in\{1,2,\dots,k\}} (-\hat{e}_j(v)\log\hat{d}_{j,s} \\
&- (\hat{n}_j - \hat{e}_j(v))\log(1-\hat{d}_{j,s})),\\
\phi &= \min(a, 1/2),\\
g &= 2 + k(k+1),\\
a &= \min_{i\ne j}\left((r_ir_j)^2/(32(r_i+r_j)^3), (r_i - 1/n)^2/r_i^3\right),\\
d_1 &= \min(d_{i,j}),\\
d_2 &= 1 - \min(1 - d_{i,j})\\
d_m &= \begin{cases} d_1, & \text{if } \min(d_1, 1-d_2) = d_1,\\ d_2, & \text{if } \min(d_1, 1-d_2) = 1 - d_2,\end{cases}\\
B &= 14k^{3/2}\log\left(\frac{1}{d_1(1-d_2)}\right),\\
m_{1,2} &= \max\left(\frac{1}{d_1^2}, \frac{1}{(1-d_2)^2}\right)\\
f_1 &= \frac{2}{c}\left(B + |\log(1-d_m)/d_m)|\right.\\
&\left.+\frac{k}{d_1(1-d_2)} + 2km_{1,2}\right),\\
f_2 &= 1/\min(d_1/2, (1-d_2)/2),\\
f &= \max(f_1, f_2),
\end{aligned}$$

and where $\hat{d}_{i,j}$ are the (random) link densities between sampled group pairs $(\hat{V}_i, \hat{V}_j)$ or inside a single sampled group.

The meaning of Theorem 1 is that if we build a classifier based on observed random sample of n nodes and its regular structure, then the rest of nodes can be classified correctly with probability that is exponentially close to one.

Proof To avoid extensively lengthy writing we use a shorthand notation with probabilistic bounds. For instance, instead of

$$\mathbf{P}\left\{ \max_{1\le i\le j} |\hat{d}_{i,j} - d_{i,j}| \le \frac{1}{\sqrt[4]{n}} \right\} \ge 1 - k(k-1)\exp(-\sqrt{n}a),$$

we write $\hat{d}_{i,j} = d_{i,j} + \frac{c_{i,j}}{\sqrt[4]{n}}$, $|c_{i,j}| \le 1$. In the final step we return to the corresponding probabilistic bounds. First, we want to show that

$$\hat{L}_{i,s}(v;\hat{D}) = \hat{L}_{i,s}(v) + \frac{f_{i,s}k}{d_1(1-d_2)}n^{3/4} + 2\sqrt{n}kg_{i,s}m_{1,2},$$
$$|f_{i,s}| \le 1, \quad |g_{i,s}| \le 1,$$

provided that

$$\frac{1}{\sqrt[4]{n}} \le \min(d_1/2, (1-d_2)/2).$$

We use the Taylor expansion of logarithm function truncated at the linear term and Taylor's remainder theorem for estimating the error:

$$\begin{aligned}
&\hat{L}_{i,s}(v;\hat{D}) \\
&\quad := \sum_{j\in\{1,2,\ldots,k\}} \left(-\hat{e}_j(v)\log\hat{d}_{j,s} - (\hat{n}_j - \hat{e}_j(v))\log(1-\hat{d}_{j,s})\right) \\
&\quad = \hat{L}_{i,s} + \sum_{j\in\{1,2,\ldots,k\}} \left[\frac{-\hat{e}_j(v)}{d_{j,s}} + \frac{(\hat{n}_j - \hat{e}_j(v))}{1-d_{j,s}}\right] \frac{c_{i,s}}{\sqrt[4]{n}} + R_2,
\end{aligned}$$

where R_2 is the remainder. First, the bound of the absolute value of the second term is:

$$\begin{aligned}
&\left| \sum_{j\in\{1,2,\ldots,k\}} \left\{ \frac{-\hat{e}_j(v)}{d_{j,s}} + \frac{(\hat{n}_j - \hat{e}_j(v))}{1-d_{j,s}} \right\} \frac{c_{i,s}}{\sqrt[4]{n}} \right| \\
&\quad \le \sum_{j\in\{1,2,\ldots,k\}} \left(\frac{|\hat{n}_{j,s}d_{j,s} - \hat{e}_j(v)0|}{d_{j,s}(1-d_{j,s})} \right) \frac{|c_{i,s}|}{\sqrt[4]{n}} \\
&\quad \le \sum_{j\in\{1,2,\ldots,k\}} \left(\frac{n}{d_{j,s}(1-d_{j,s})} \right) \frac{1}{\sqrt[4]{n}} \\
&\quad \le \frac{k}{d_1(1-d_2)} n^{3/4},
\end{aligned}$$

where we used the facts that $\hat{n}_{j,s} \le n$ and $\hat{e}_j(v) \le n$. As a result, we get the second term in the Taylor expansion of $\hat{L}_{i,s}(v;\hat{D})$. Next, we estimate the absolute value of R_2. Obviously, R_2 is a sum of remainders corresponding to each term of the sum of $\hat{L}_{i,s}(v;\hat{D})$. For each term we use the proposition according to which $f(a+\delta) = f(a) + f'(a)\delta + r_2(x)$, where $|r_2(x)| \le M\delta^2/2$, where $M = \sup_{x\in(a-|\delta|,a+|\delta|)} |f''(x)|$. In our case we have terms like

$$\begin{aligned}
&\frac{1}{2} \sup_{d_{j,s}-1/\sqrt[4]{n}\le x\le d_{j,s}+1/\sqrt[4]{n}} \left| \frac{\hat{e}_j(v)}{x^2} + \frac{\hat{n}_j - \hat{e}_j}{(1-x)^2} \right| \frac{|c_{j,s}|^2}{\sqrt{n}} \\
&\quad \le \frac{1}{2} \left\{ \frac{n}{(d_{j,s} - 1/\sqrt[4]{n})^2} + \frac{n}{(1-d_{j,s}-1/\sqrt[4]{n})^2} \right\} \frac{|c_{j,s}|^2}{\sqrt{n}} \\
&\quad \le \frac{1}{2} \left\{ \frac{1}{(d_1 - 1/\sqrt[4]{n})^2} + \frac{1}{(1-d_2-1/\sqrt[4]{n})^2} \right\} \sqrt{n} \\
&\quad \le 2\sqrt{n} \max\left(\frac{1}{d_1^2}, \frac{1}{(1-d_2)^2} \right),
\end{aligned}$$

where we used the condition $\frac{1}{\sqrt[4]{n}} \le \min(d_1/2, (1-d_2)/2)$. Using this uniform bound for all terms, we obtain the third term in the expansion.

Next, we need to analyze the first term $\hat{L}_{i,s}(v)$, the cost function with the right densities, using Propositions 5 and 7. From Proposition 5 it follows that

$$\hat{L}_{i,s}(v) = \mathbf{E}\hat{L}_{i,s}(v) + c_s z_{i,s} n^{3/4}, \quad |z_{i,s}| \le 1,$$

where the expectation, conditioned on the sizes of sampled groups $\hat{n}_j$, is

$$\begin{aligned}
&\mathbf{E}\hat{L}_{i,s}(v) \\
&\quad = \sum_{j=1}^{k} \left[-\frac{e_j(v)\hat{n}_j}{n_j} \log d_{j,s} - \left[\hat{n}_j - \frac{e_j(v)\hat{n}_j}{n_j} \right] \log(1-d_{j,s}) \right].
\end{aligned}$$

Using Proposition 7, we get

$$\hat{n}_j = r_j n + 14\sqrt{k} s_j n^{3/4}, \quad |s_j| \le 1.$$

Substituting this, and noting the fact $n_j = r_j N$, we have

$$\begin{aligned}
&\mathbf{E}\hat{L}_{i,s}(v)\\
&\quad= \sum_{j=1}^{k} \left\{ -\frac{e_j(v) r_j n}{r_j N} \log d_{j,s} \right.\\
&\quad\left. - \left(r_j n - \frac{e_j(v) r_j n}{r_j N} \right) \log(1 - d_{j,s}) \right\}\\
&\quad+ 14\sqrt{k} n^{3/4} \sum_{j=1}^{k} s_j \left\{ -\frac{e_j(v)}{n_j} \log d_{j,s} \right.\\
&\quad\left. - \left(1 - \frac{e_j(v)}{n_j} \right) \log(1 - d_{j,s}) \right\}.
\end{aligned}$$

The first term is

$$\begin{aligned}
&\frac{n}{N} \sum_{j=1}^{k} \left(-e_j(v) \log d_{j,s} - (n_j - e_j(v)) \log(1 - d_{j,s}) \right)\\
&\quad= \frac{n}{N} L_{i,s}(v),
\end{aligned}$$

and the second term has an absolute bound:

$$\begin{aligned}
&14\sqrt{k} n^{3/4} \left| \sum_{j=1}^{k} s_j \left\{ -\frac{e_j(v)}{n_j} \log d_{j,s} \right. \right.\\
&\quad \left. \left. - \left(1 - \frac{e_j(v)}{n_j} \right) \log(1 - d_{j,s}) \right\} \right|\\
&\qquad \le 14\sqrt{k} n^{3/4} \sum_{j=1}^{k} (-\log d_{j,s} - \log(1 - d_{j,s}))\\
&\qquad \le 14\sqrt{k} k n^{3/4} \log \left(\frac{1}{d_1(1 - d_2)} \right),
\end{aligned}$$

where we used the fact that $0 \le e_j(v)/n_j \le 1$. As a result we have:

$$\mathbf{E}\hat{L}_{i,s}(v) = \frac{n}{N} L_{i,s}(v) + B b_{i,s} n^{3/4},$$

with $|b| \le 1$ and B as in the claim of the theorem. Combining these estimates, we have

$$\hat{L}_{i,s}(v) = \frac{n}{N} L_{i,s}(v) + (B b_{i,s} + c_s z_{i,s}) n^{3/4}.$$

Gathering all pieces, we get finally:

$$\hat{L}_{i,s}(v; \hat{D}) = \frac{n}{N} L_{i,s}(v) + u_{i,s} n^{3/4}$$

with

$$u_{i,s} = B b_{i,s} + c_s z_{i,s} + \frac{f_{i,s} k}{d_1(1 - d_2)} + \frac{2k g_{i,s} m_{1,2}}{\sqrt[4]{n}}.$$

Thus,

$$\begin{aligned}
&\hat{L}_{i,s}(v; \hat{D}) - \hat{L}_{i,i}(v; \hat{D})\\
&\quad= \frac{n}{N} \left(L_{i,s}(v) - L_{i,i}(v) \right) + (u_{i,s} - u_{i,i}) n^{3/4},
\end{aligned}$$

since by condition

$$\frac{n}{N} \left(L_{i,s}(v) - L_{i,i}(v) \right) \ge cn,$$

a sufficient condition for $\hat{L}_{i,s}(v; \hat{D}) - \hat{L}_{i,i}(v; \hat{D}) > 0$ is that $cn > |u_{i,s} - u_{i,i}| n^{3/4}$. Since $|u_{i,s} - u_{i,i}|/c$ is bounded by a constant f_1, this condition is achievable with sufficiently large n. It can be seen that we can choose f_1 as in the claim of the theorem. We had another condition $\frac{1}{\sqrt[4]{n}} \le \min(d_1/2, (1 - d_2)/2)$, or $\sqrt[4]{n} \ge 1/\min(d_1/2, (1 - d_2)/2) := f_2$.

As a result, if $f = \max(f_1, f_2)$, then $n \ge f^4$ implies $\hat{L}_{i,s}(v; \hat{D}) - \hat{L}_{i,i}(v; \hat{D}) > 0$, with a probability that we estimate now. We use a simple union bound. Let $A_1, A_2, \ldots$ denote some events that correspond to the violation of one of the conditions in our proof. Then the event that at least one of them happens is given by the union of these events. The probability of this event always fulfills $\mathbf{P}(\cup_i A_i) \le \sum_i \mathbf{P}(A_i)$. Thus, the probability that none of the conditions is violated is not less than $1 - \sum_i \mathbf{P}(A_i)$. All $\mathbf{P}(A_i)$ are exponentially small in n, and so is their sum, with an exponent that has the smallest positive constant. The result of this count is given in the formulation of the theorem. □

6 Simulations for Large-Graph Analysis Based on Sampling

6.1 Regular Decomposition

We illustrate the suggested sampling scheme and its application to analyze very large graphs. The results are illustrated in Figs. 3, 4, 5, 6, 7 and 8.

Assume that a large graph is generated from a SBM. Denote the blocks by

$V_1, V_2, \dots, V_k$
and the link probability matrix by P, that is, a $k \times k$ symmetric matrix with entries in (0, 1). We also assume that no two rows are identical, to avoid a redundant structure. Then we generate links between blocks and within blocks using P as independent link probabilities. The size of the graph G is assumed large and denoted by N.

Now take a uniform random sample of size n of the nodes. Denote the graph induced by sampling as G_n. The relative sizes of blocks are denoted as $r_i := \frac{|V_i|}{N}$. The probability of sampling a node from block i is r_i. We want to test whether the block structure of G can be reconstructed from a small sample, the size of which does not depend on N and with time complexity $\Theta(N)$. We believe that the answer is positive.

For simplicity, assume that we know the block structure of G_n by having run an ideally working RD algorithm. This is not too far from reality, although some of the nodes may in reality be misclassified. However, there is not a big difference, if most of the nodes are correctly classified and n is not very small. The real graphs, however, are not generated from any SBMs, and this is a more serious drawback. In the latter case we would like to show that it is not necessary to run RD on G; rather, it is sufficient to run RD on G_n and just classify the rest of the nodes based on that structure.

In G_n, we denote the blocks as $\hat{V}_1, \hat{V}_2, \dots, \hat{V}_k$, and the empirical link densities between these blocks as $\hat{d}_{i,j}$. The relative sizes of the blocks as well as the link densities deviate in G_n from their counterparts in G. Now assume that we take a node $i \in V_\beta$ outside G_n together with its links to G_n. What is the probability that the RD classifier places this node into $\hat{V}_\beta$?

Denote by $e_\alpha(v)$ the number of links from v to $\hat{V}_\alpha$, $n_\alpha := |\hat{V}_\alpha|$. The RD classifier based on G_n is the following program:

$$C_\alpha(v|\hat{\xi}, \hat{d}, k) := \sum_{j=1}^{k} \left[-e_j(v) \log \hat{d}_{j,\alpha} - (n_j - e_j(v)) \log(1 - \hat{d}_{j,\alpha})\right],$$
$$\alpha^* = \arg\min_\alpha (C_\alpha(v|\hat{\xi}, \hat{d}, k)),$$

where $\hat{\xi} = \left\{\hat{V}_1, \hat{V}_2, \dots, \hat{V}_k\right\}$ and $\hat{d}$ stands for the $k \times k$ matrix with matrix elements $\hat{d}_{i,j}$. To compute the optimal block for node i we need to compute and check a list of k numbers as well as compute the densities $e_\alpha(i)$. The computation time is upper-bounded by $\Theta(kn)$. For a size n that is independent on N, the computation takes only a constant time. By this, we mean that it is enough to have some constant n regardless of how large the graph is. Such an n depends on the relative sizes of blocks and on k. This is because it is obviously necessary to have enough samples from each block to be able to obtain good estimates of link densities, and if some blocks are very small, more samples are needed.

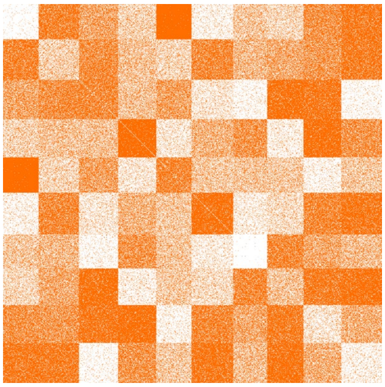

Fig. 3 Adjacency matrix of a random graph with $k = 10$, and equal size blocks and 1000 nodes

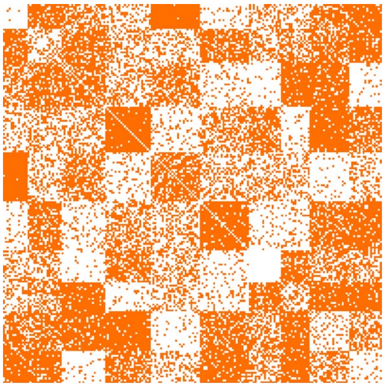

Fig. 4 Adjacency matrix of a random graph with $k = 10$, and equal size blocks, generated from the same model as in previous picture but with only 200 nodes

We made experiments with $k = 10$ and with equal relative sizes of blocks. The P-matrix elements were i.i.d. Uniform(0, 1) r.v.'s. In this case, already $n > 200$ is such that no classification error was found in a long series of experiments with repetitions of random samples and classification instances. Of course, errors are always possible, but they become insignificant already when each block has about 20 members. A similar situation is met with non-equal block sizes, but then the small blocks dictate the required n. When the smallest block has more than a couple of dozen nodes, the errors become unnoticeable in experiments.

We conjecture that if such a block structure or close to it exists for a large graph, it can be found out using only a very small subgraph and a small number of links per node in order to place every node into right blocks with only very few errors. This would give a coarse view on the large graph

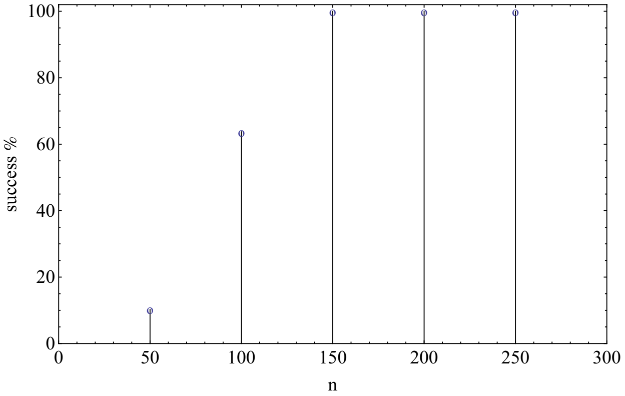

Fig. 5 Classification success percentile as a function of n using several repetitions of experiments and 1000 classification instances for each. Already a sample with 200 nodes generates a model with almost a perfect classifier

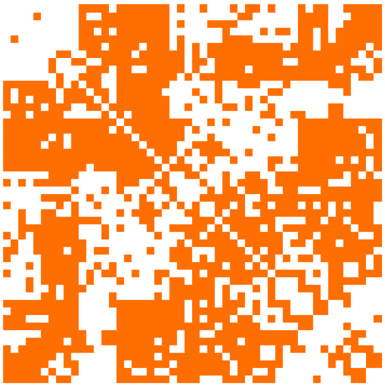

Fig. 6 A sample graph with 50 nodes that is insufficient to create a successful classifier—the result is similar to completely random classification

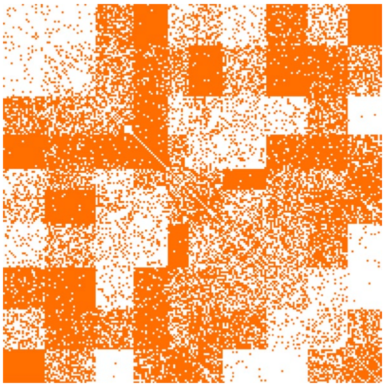

Fig. 7 200-node sample, from the same model as above, that generates almost a perfect classifier—no errors detected in experiments

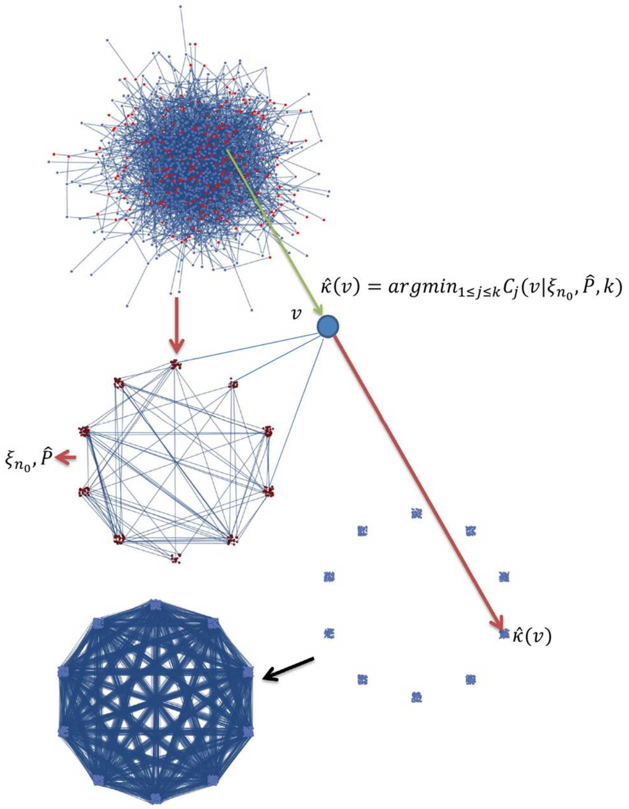
$\hat{\kappa}(v) = argmin_{1\le j\le k} C_j(v|\xi_{n_0}, \hat{P}, k)$
v
$\xi_{n_0}, \hat{P}$
$\hat{\kappa}(v)$

Fig. 8 A scheme of RD for a huge graph shown at the top; in reality, we assume a much larger graph than what is shown in the picture. First, a moderate size sample of n_0 nodes and induced subgraph is formed. RD is applied to this small graph; groups $\hat{\xi}_{n_0}$ and matrix $\hat{P}$ are found, shown as the graph in the middle. Then, sequentially, any node v from the big graph can be classified to a regular group using just counts of links of this chosen node to regular groups of the sample graph. This classification requires a constant number of computations, upper-bounded by $k^2 n_0$, with elementary functions. As a result, nodes of the whole graph can be classified in just linear time with respect to number of nodes. After the classification is done (shown in the ring shape structure at the right side) the RD of the large graph is done simply by retrieving link information. The result is shown in the lower graph

useful for getting an overall idea of link distribution without actually retrieving any link information besides labeling of its nodes. For instance, if we have a large node set, then the obtained labeling of nodes and the revealed block densities would allow computation of, say, the number of triangles or other small subgraphs, or the link density inside a set. It would be possible to use this scheme in the case of a dynamical graph, to label new nodes and monitor the evolution of its large-scale structure.

We can also adapt our RD method to the case of multilayer networks and tensors, etc. A similar sampling scheme would also be desirable and probably doable. On the other hand, large sparse networks need principally new solutions and development of a kind of sparse version of RD. These are major directions of our future work.

6.2 Spectral Clustering and Comparison with RD

For comparison, we repeated the experiments of the previous section using a spectral clustering method (see, e.g., [21]), to find a block structure in the sample. For more about the rigorous theory of spectral clustering and its variants, see [17].

The purpose is to compare the performance of spectral clustering with RD. The quality of spectral block structure was then evaluated similarly as in previous RD case, by counting the success rate of classification of additional 1000 nodes of the graph, and using the same classifier as in previous section. We also checked the accuracy of spectral clustering versus RD.

Spectral clustering is a highly popular way of obtaining community and other stochastic block model structures in data. It combines spectral analysis of most significant eigenvectors associated with the graph with k-means clustering. For definiteness, we describe the steps of this method as follows. Let $W_{i,j}$ be the adjacency matrix of a simple graph with n nodes. Define a diagonal matrix D with elements

$$D_{i,i} = \sum_{1 \le j \le n} W_{i,j}$$

and the normalized adjacency matrix W_D as

$$W_D = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}.$$

W_D is related to the normalized graph Laplacian as $L = I - W_D$. (Note that in [21] W_D is denoted as L.) Denote k eigenvectors of W_D corresponding to k largest by absolute value eigenvalues as $X_1, \dots X_k$. Form a matrix X, where the columns are $X_1, \dots X_k$. As a result, X is an $n \times k$ matrix. Each row of X is treated as vector in k-dimensional Euclidean space. Next, run the k-means algorithm on these n vectors and cluster them into k clusters. The clustering of nodes is taken to be identical to the k-means clustering of rows of X.

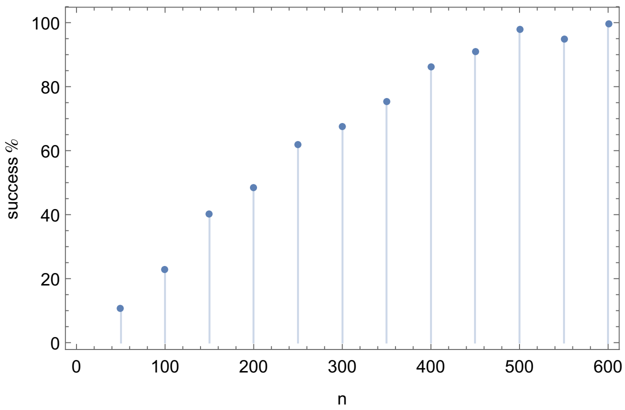

Fig. 9 Spectral method for finding block structure. Classification success percentile as a function of n using several repetitions (100) of experiments and 1000 classification instances for each

In our case, it turned out that if we take $k = 10$ in the matrix X and $k = 10$ in k-means algorithm, the result was very poor and the blocks were not found. Based on numerical experiments we choose $k = 5$ for X and $k = 10$ in k-means, in order to have best results for finding the 10×10 block structure of W. In this way the method was able to find the blocks with reasonable accuracy.

The spectral method performed somewhat worse than RD, as can be seen in Fig. 9. It needed a sample size of about 1000 nodes to have a low average error rate, instead of 200 that was enough for RD. Moreover, even for $n = 1000$, some random cases resulted in very poor performance, corresponding to a completely unsuccessful classifier. When n is around thousand and making 100 repetitions, we observed that almost all cases were error-free, but in a couple of cases the success rate was only around 10%, meaning a completely random classification. The blocks in our experiments did not have fixed sizes, only their expected sizes were fixed and equal. In a series of experiments, the link probability matrix was kept constant. Maybe these random variations of block sizes can explain such an instability. If n is not large enough, these variations can be very big. One cluster can become very small just by fluctuations caused by sampling. Such small clusters can be seen as outliers, since the nodes in such groups are very few and different from all others. It is well known that spectral clustering suffers from such outliers that should be removed from data before clustering. RD does not suffer from this feature, and we did not observe sensitivity of clustering to random fluctuations of the cluster sizes.

In the theory of spectral clustering there are usually balancing conditions on the way how the sizes of clusters grow as $n \to \infty$ [17]. For instance, the block sizes $n_1, \dots, n_k$ are such that $\exists c : \quad 0 < c \le 1/k$ s.t. $\sum_i n_i \to \infty$ and $n_i/n \ge c$ as $n \to \infty$. Such conditions are needed to prove consistency properties of spectral clustering methods [15, 17].

Results are demonstrated in Figs. 10, 11 and 12. Here we pick up one particular sample where spectral clustering produced a poor model with high rate of errors in classification. It was found that the block structure was only partially found using this method. Applying RD to the same sample produced a perfectly correct clustering. We also checked all cases where the spectral clustering produced a poor classification model. We found that in these cases the smallest and largest blocks of the sampled graph deviated a lot from their expectation, while in the case that the model was successful the sizes were systematically closer to the expectation, see Fig. 12.

We conclude that RD is more preferable and stable for finding a block structure of a sample graph. RD is capable of finding blocks in the case of poor balancing of block sizes. This can happen when sampling a large graph even though the blocks are balanced yet, say, the biggest block is twice

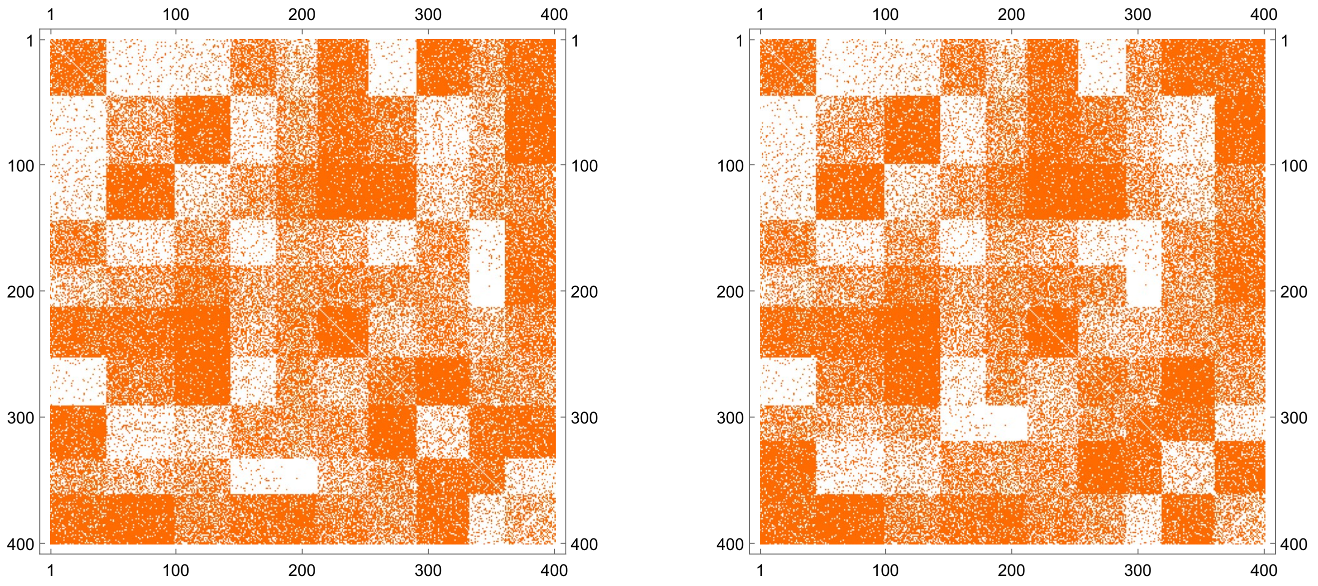

Fig. 10 Left: the true block structure. Right: the block structure found with spectral clustering. There are some large differences to be seen. In this case classifier based on the structure at right completely fails. In comparision, the RD method finds the correct structure without a single error

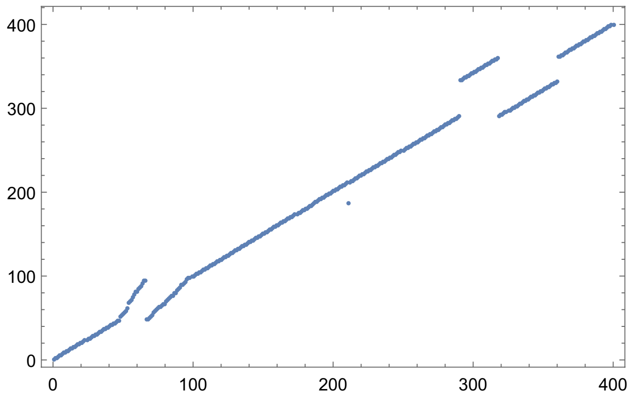

Fig. 11 Errors of spectral clustering in case when the corresponding classifier has high error rate, for a fixed sample graph with $n = 400$. The points correspond to nodes. Nodes exactly at diagonal are correctly clustered, and those off the diagonal are in wrong clusters. In case of applying RD to the same graph the nodes are 100% correctly clustered

the smallest. In the sample this can result in a fatal error for spectral clustering, when one sampled block, by chance, has only few representatives. RD seems to be more robust against such small samples of blocks and thus requires smaller sample sizes than spectral clustering.

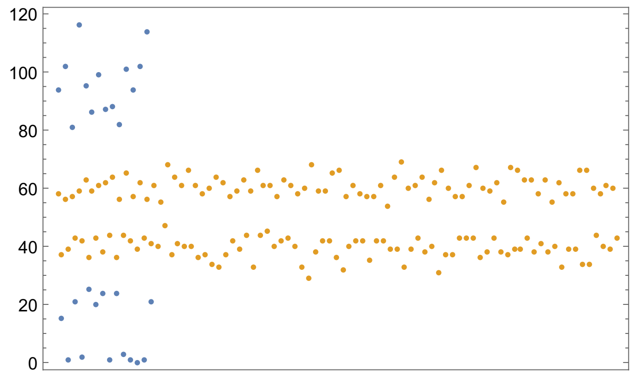

Fig. 12 Minimal and maximal block sizes in various experiments. These variations are result of choosing cluster assignment of each node uniformly at random and independently of all other nodes. Each horizontally aligned point pair corresponds to values in a single experiment, where the horizontal coordinate is the value itself. Blue points are from instances that resulted in high, more than 30 %, error rate in subsequent classification of nodes. The orange points correspond to cases when this error rate was less than 1%. Average block size is 50, following from parameters $n = 500$ and $k = 10$. This plot suggests high correlation between error rate and large deviations in block sizes. In case of blue points, the values are roughly twice larger or smaller than the average and in case of orange points the difference is much smaller

7 Conclusions and Future Work

Our future work will be dedicated to the case of sparse graphs, which is the most important in Big Data. One idea we are currently studying is to use the matrix of hop-count distances between nodes instead of a sparse adjacency matrix [10].

For large dense graphs, testable graph parameters are nonparametric statistics that can be consistently estimated by appropriate sampling, introduced by László Lovász and coauthors, see also [17]. We plan to show that the MDL statistics and the multiway discrepancies of [6] are testable, possibly with appropriate normalizations. If so, they can be concluded from a smaller part of the graph that has advantages with regard to the computational complexity of our algorithms.

Acknowledgements This work was supported by the Academy of Finland Project 294763 Stomograph and by the ECSEL MegaMaRT2 Project. The research of Marianna Bolla was supported by the BME-Artificial Intelligence FIKP grant of EMMI (BME FIKP-MI/SC).

References

1. Reittu H, Norros I, Bazsó F (2017) Regular decomposition of large graphs and other structures: scalability and robustness towards missing data, In: Al Hasan M, Madduri K, Ahmed N (eds) Proceedings of fourth international workshop on high performance big graph data management, analysis, and mining (BigGraphs 2017), Boston, USA
2. Reittu H, Bazsó F, Norros I (2017) Regular decomposition: an information and graph theoretic approach to stochastic block models. arXiv:1704.07114 [cs.IT]
3. Decelle A, Krzakala F, Moore C, Zdeborová L (2011) Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. Phys Rev E 84:066106
4. Abbe E (2017) Community detection and stochastic block models: recent developments. arXiv:1703.10146v1 [math.PR]
5. Szemerédi E (1976) Regular partitions of graphs. Problemés Combinatories et Téorie des Graphes, number 260 in Colloques Internationaux C.N.R.S., Orsay, pp 399–401
6. Bolla M (2016) Relating multiway discrepancy and singular values of nonnegative rectangular matrices. Discrete Appl Math 203:26–34
7. Nepusz T, Négyessy L, Tusnády G, Bazsó F (2008) Reconstructing cortical networks: case of directed graphs with high level of reciprocity. In: Bollobás B, Miklós D (eds) Handbook of large-scale random networks. Bolyai society of mathematical sciences, vol 18. Springer, Berlin, pp 325–368
8. Pehkonen V, Reittu H (2011) Szemerédi-type clustering of peer-to-peer streaming system. In: Proceedings of Cnet 2011, San Francisco, USA
9. Reittu H, Bazsó F, Weiss R (2014) Regular decomposition of multivariate time series and other matrices. In: Fränti P, Brown P, Loog M, Escolano F, Pelillo M (eds) Proceedings of S+SSPR 2014, vol 8621 in LNCS. Springer, pp 424–433
10. Reittu H, Leskelä L, Räty T, Fiorucci M (2018) Analysis of large sparse graphs using regular decomposition of graph distance matrices. In: Proceedings of IEEE BigData 2018, Seattle USA., pp 3783–3791, Workshop: advances in high dimensional (AdHD) big data, Ed. Sotiris Tasoulis
11. Grünwald PD (2007) The minimum description length principle. MIT Press, Cambridge
12. Peixoto TP (2013) Parsimonious module inference in large networks. Phys Rev Lett 110:148701
13. Caron F, Fox B (2017) Sparse graphs using exchangeable random measures. J R Stat Soc B 79(Part 5):1295–1366
14. Borgs C, Chayes J, Lee CE, Shah D (2017) Iterative collaborative filtering for sparse matrix estimation. arXiv:1712.00710
15. von Luxburg V, Belkin M, Bousquet O (2008) Consistency of spectral clustering. Ann Stat 38(2):555–586
16. Pollard D (1981) Strong consistency of k-means clustering. Ann Stat 9:35–40
17. Bolla M (2013) Spectral clustering and biclustering: learning large graphs and contingency tables. Wiley, Hoboken
18. Fox J, Lovász LM, Zhao Y (2017) On regularity lemmas and their algorithmic applications. arXiv:1604.00733v3 [math.CO]
19. Janson S, Łuczak T, Ruciński A (2000) Random graphs. Wiley, Hoboken
20. Hayes TP (2005) A large-deviation inequality for vector-valued martingales. http://www.cs.unm.edu/hayes/papers/VectorAzuma/VectorAzuma20030207.pdf
21. Rohe K, Chatterjee S, Yu B (2011) Spectral clustering and high-dimensional stochastic blockmodel. Ann Stat 39(4):1878–1915