Community Recovery in the Degree-Heterogeneous Stochastic Block Model

Vincent Cohen-Addad, Frederik Mallmann-Trenn, David Saulpic

Proceedings of Machine Learning Research vol 178:1–32, 2022 35th Annual Conference on Learning Theory

Community Recovery in the Degree-Heterogeneous Stochastic Block Model

Vincent Cohen-Addad VCOHENAD@GMAIL.COM
Google Research

Frederik Mallmann-Trenn FREDERIK.MALLMANN-TRENN@KCL.AC.UK
King's College London, UK

David Saulpic DAVID.SAULPIC@LIP6.FR
LIP6, Sorbonne Université, Paris France

Editors: Po-Ling Loh and Maxim Raginsky

Abstract

We consider the problem of recovering communities in a random directed graph with planted communities. To model real-world directed graphs such as the Twitter or Instagram graphs that exhibit very heterogeneous degree sequences, we introduce the *Degree-Heterogeneous Stochastic Block Model* (`DHSBM`), a generalization of the classic Stochastic Block Model (`SBM`), where the vertex set is partitioned into communities and each vertex u has two (unknown) associated probabilities, p_u and q_u, $p_u > q_u$. An arc from u to v is generated with probability p_u if u and v are in the same community and with probability q_u otherwise. Given a graph generated from this model, the goal is to retrieve the communities.

The `DHSBM` allows to generate graphs with planted communities while allowing heterogeneous degree distributions, a quite important feature of real-world networks.

In the case where there are two communities, we present an iterative greedy linear-time algorithm that recovers them whenever $\min_u \frac{p_u - q_u}{\sqrt{p_u}} \geq C\sqrt{\log(n)/n)}$, for some absolute constant C. We also show that, up to a constant, this condition is necessary. Our results also extend to the standard (undirected) `SBM`, where $p_u = p$ and $q_u = q$ for all nodes u. Our algorithm presents the first linear-time algorithm that recovers exactly the communities at the asymptotic information-theoretic threshold, improving over previous near-linear time spectral approaches.

Keywords: Community Detection; Stochastic Block Model; Degree-Heterogeneous Stochastic Block Model.

1. Introduction

Graph clustering is a central tool for understanding complex networks and extracting useful information from them. As such, graph clustering is used in a wide range of applications including: recommendation systems Gandomi and Haider (2015), link prediction Liben-Nowell and Kleinberg (2007), biological networks Girvan and Newman (2002) (e.g., protein-protein interaction networks), natural language processing Manning et al. (1999), or social networks Mishra et al. (2007). One of the most common test-beds for designing graph clustering algorithms is the *Stochastic Block Model* (`SBM`).

The `SBM` allows the sampling of random graphs having an intrinsic cluster structure with high densities of edges within clusters and low densities across clusters. In its most basic setting, a graph generated from the `SBM` consists of two hidden ground-truth clusters V_1 and V_2 each of size n; then for each pair of nodes, an edge is added to the graph with probability p if both nodes are from

the same cluster and with probability q otherwise. The SBM is used to analyze graph clustering algorithms in a *beyond-worst-case* scenario: a good clustering algorithm should be able to recover the ground-truth partition (V_1, V_2). However, a limitation of the model is that a graph sampled from the SBM is very likely to be almost-regular, i.e., all nodes have a degree that is concentrated around $np + nq$. Although unrealistic, that property is crucial to the proof of recovery of most known algorithms for graph clustering in SBM, which may fail when the degree distribution is not tightly concentrated (see the discussion below).

Of course, in the real-world, most interesting (directed) graphs such as Twitter, the Instagram graphs of followers or the Facebook friendship (undirected) graph are irregular and influential people are much more connected than the average user. For example, in the Twitter graph, influential people have a larger "incoming" degree: a larger number of people connected to them (following them). Thus, the degrees in real-word graphs are often highly heterogeneous. We therefore put forth a new more realistic model, which we call the *Degree-Heterogeneous Stochastic Block Model* (DHSBM) that would enable cluster structure and heterogeneous degrees. Here, for each node u there are two parameters p_u and q_u, and we think of p_u as the probability that vertex v in the same community as u "follows" u (or is connected towards u), while q_u is the probability of vertex v in a different community to follow u (or to be connected towards u). We can thus generate a directed graph according to the probability distributions defined by the p_us and q_us. We believe that this graph is a great model of directed network with cluster structure such as the Twitter or Instagram graphs where we expect influential people to have a large number of followers but not necessarily to follow much more people than the average. This can be easily reflected in our model by setting a high value for p_u for the most influential persons. Since p_u, q_u can take arbitrary values, this allows for a highly heterogeneous network representing the various levels of popularity of the nodes.

Then, the question is whether one can design efficient algorithms for identifying the ground-truth cluster structure in this model (i.e.: recovering the underlying communities). In fact, a more basic natural question is what is the information-theoretic threshold for exact recovery in this graph model? Is it the same as the SBM, harder, easier or are they incomparable?

1.1. Our Results

We answer the above questions as follows. Our positive results (Theorem 1 and Theorem 2), show that there exists a linear-time algorithm able to recover the ground-truth partition of the DHSBM w.p. $2/3$, assuming sufficient separation between p_u and q_u. Note here that there are two sources of randomness: the randomness coming from the model itself, which we call the graph randomness, and the random bits used by the algorithm.

Theorem 1 *Consider the* DHSBM *with two communities and N nodes, with probability vectors $\mathbf{p} = \{p_u \mid u \in V\}$ and $\mathbf{p} = \{q_u \mid u \in V\}$, and minimum community size N/f with $f = O(\log \log N)$. Let $\gamma = \min_u \frac{p_u - q_u}{\sqrt{p_u}}$. Then, there exists a constant C such that if $\gamma \geq C f^{5/2}\sqrt{\frac{\log N}{N}}$, then there exists an algorithm that recovers the communities, w.p. $1 - o(1)$ on the graph's randomness, and at least $2/3$ on the algorithm's randomness. Moreover, the algorithm runs in linear time.*

Theorem 2 *Consider the* SBM *with two communities and N nodes, with probabilities p and q, and a minimum community size N/f with $f = O(\log \log N)$. Let $\gamma = \frac{p-q}{\sqrt{p}}$. Then, there exists a constant C such that if $\gamma \geq C f^{5/2}\sqrt{\frac{\log N}{N}}$, then there exists an algorithm that recovers the communities, w.p.*

$1 - o(1)$ *on the graph's randomness, and at least* $2/3$ *on the algorithm's randomness. Moreover, the algorithm runs in linear time.*

In the *SBM*, it is known that $\gamma > \sqrt{2}\sqrt{\frac{\log n}{n}}$ is necessary for recovery (see Abbe et al. (2015) and Mossel et al. (2015)). Our algorithm has almost the same threshold – up to a constant. We complement our upper bounds by showing that in the *DHSBM*, there exists some setting of p_u, q_u where our bound is indeed tight up to a constant factor, and so the information-theoretic threshold for exact recovery matches the *SBM*'s one up to constant factors.

Theorem 3 *Fix any* $c_p < 1/80$. *Consider the* DHSBM *in which for all nodes* u, $p_u = c_p \log n/n$. *Assume that* q_u *is such that* $(p_u - q_u)/\sqrt{p_u} \leq \frac{c_p}{20}\sqrt{\log n/n}$. *Then, no algorithm recovers the community with a success probability of more than* $1/2$ *on the graph randomness.*

Note that the lower bounds in the *SBM* do not translate to lower bounds in the *DHSBM* due to the directedness of the edges. The direction of the edges provides more information: just because a node has more edges from the "wrong" cluster than from its own does not mean it is impossible to recover its cluster: the structure of the outgoing edges may contain enough information as the following example illustrates. Consider a graph where $n-1$ nodes violate the γ threshold by having $p_u, q_u = 1/2$ and one single node satisfies the threshold γ with $p_1 = 1, q_1 = 0$. In this contrived example, exact recovery is trivial.

1.2. Technical Contributions

Challenges The state-of-the-art algorithms for community detection, both in terms of running time and recovery threshold, are spectral algorithms. These algorithms crucially rely on the following property of the standard *SBM*: the expected adjacency matrix consists of only 2 different columns (and rows). In the *DHSBM*, however, it is not clear how spectral methods could be helpful. At first glance, this might be surprising since the expected adjacency matrix is of rank 2. However, what makes the recovery challenging is that nodes of the same community can have vastly different p_u. As a consequence, the homogeneity of the graph breaks, and approaches such as McSherry (2001); Chin et al. (2015) that rely on bounding the Frobenius norm do not work. Furthermore, algorithms that rely on gaps between the eigenvalues (e.g., Wang et al. (2020); Abbe (2018); Abbe and Sandon (2015)) cannot be used as the top-k eigenvalues can be modified almost arbitrarily by tuning the p_u. In contrast, SDP based algorithms are robust to some adversarial perturbation of the SBM (Moitra et al. (2016)), but are desperately slow with a large polynomial running time. Therefore, we need to move away from those spectral and SDP algorithms in order to design efficient algorithms for *DHSBM*.

Contribution Our algorithm relies on the following principle: given a partition, moving a vertex from one part to the part where it has most neighbors should somewhat *improve* the quality of the partition. Based on this idea, we design an algorithm that allows to formalize this notion of improvement.

For simplicity, suppose first that the current partition splits the vertices into two parts of equal size. Our main technical contribution is a precise understanding of the probability that a given vertex has more edges toward one side of the partition than the other, given how the two communities are split by the partition. We characterize this probability optimally, up to constant factors in the second-order term.

That probability depends on a key quantity, dubbed the *discrepancy*: in the case where the two communities C_1 and C_2 have the *same size*, the discrepancy of the partition S_1, S_2 is $\Delta = |C_1 \cap S_1| - |C_2 \cap S_1|$. This naturally impacts the probability of having more edges toward one side than the other, as that probability depends on the repartition of the communities. More precisely, we show that for a vertex u, the probability that it has more edges towards S_1 than S_2 is $1/2 + \Delta(p_u - q_u)/\sqrt{p_u}$, where Δ is the current discrepancy.

To prove that probability bound, we analyze a natural coupling between binomial variables that count the number of edges towards each community. The standard way to achieve was to go through Gaussians, but it falls short of achieving optimal bound – missing in particular the $1/\sqrt{p_u}$ factor, crucial to work in sparse graphs. Leaving Gaussians behind, we are able to bound the binomials directly. To do so, we characterize the binomial distributions $\textsc{Bin}(n, p)$ around $np \pm \sqrt{np}$, and show that, to our purpose, $\textsc{Bin}(n, p)$ is approximately uniform on that interval. Perhaps surprisingly, this approximation yields much stronger bounds than the Gaussian approach, in particular when np is small. We believe our analysis sheds a new light on the behavior of the family of algorithms that greedily improve a partition, based on that idea.

This allows us to show a lower bound on the probability to have more edges to one part of the partition than to the other, based on the discrepancy of the partition. Informally, the bound suggests that the higher the discrepancy is, the more likely a vertex will be moved to the *right* side of the partition, namely the side that has most vertices from the same community. In other words, the higher the discrepancy is, the faster it increases. Our algorithm is designed to exploit this fact, and works in rounds, each round designed to increase exponentially the discrepancy.

Working in rounds allows us allows us to crucially bypass dependency issues: if we only were to update a partition of the whole vertex set vertex by vertex, each step would be very dependent from previous ones (and the revealed randomness of the edges), and the previous probability statement would break. Instead, our algorithm works as follows: it breaks the vertices into several groups, and for each part it finds a partition using a partition of the previous group, by placing each vertex in the part towards which it has most edges. This ensures that decisions taking in a round are based only on edges from one group to the previous one: those decisions are therefore all independent, since each edge will be considered for at most one decision.

In the ideas presented above, we have swept one challenge under the rug: at some point of the algorithm, it may be that the two sides of the partition do not have the same exact size – for instance, when the two communities are not perfectly balanced. Say the partition S_1, S_2 of S is such that $|S_1| > |S_2|$. In that case, the previous argument needs to be changed: now it is more likely that any vertex has more edges towards S_1, regardless of the distribution of nodes. To cope with that issue, we introduce a subsampling procedure in our algorithm. Instead of comparing the edges towards the two sides, we sample randomly $|S_2|$ vertices from S_1, and compare the number of edges toward this sample.

While quite natural, this idea introduces a new layer of technicality: the discrepancy as defined previously becomes a random variable depending on the sampling's randomness. We manage to relate its expected value to the *proportion* of each community in part S_1, and show it is tightly enough concentrated around that expectation. Therefore, instead of tracking directly the discrepancy, our proofs tracks the proportion of vertices of each community.

We complement our algorithm by providing a lower bound for the case where we have $\frac{p_u - q_u}{\sqrt{p_u}} \leq c\sqrt{\log n/n}$, for some constant c, by showing that is not possible to recover the two communities.

The general idea is that, with high probability, two nodes u, v have exactly the same degree from and toward each community. Hence, the graph has the same chances of being drawn from a $\texttt{DHSBM}$ where $u \in C_1, v \in C_2$ or from a $\texttt{DHSBM}$ where $u \in C_2, v \in C_1$. Therefore, an algorithm that only observes the graph must fail to recover u and v's community on at least half of the graphs. More precisely, the graph generated has one chance out of two of fooling the algorithm. This idea is formalized in section C.

1.3. Related Work

The related work on the Stochastic Block Model is too vast to be covered entirely in this paper and we refer the interested reader to the survey of Abbe (2018). The precise understanding of what can be recovered as a function of p and q in the Stochastic Block Model is due to Abbe et al. (2015) and Mossel et al. (2015). They prove that, when p and q are in $\Theta(\log n/n)$, exact recovery is possible if and only if $^{p-q}/_{\sqrt{p}} > \sqrt{^{2\log n}/_{n}}$. Classic results include the fundamental result of McSherry (2001), the augmentation algorithm of Condon and Karp (2001). Iterative methods Gao et al. (2017); Zhang and Zhou (2020), Semi-Definite Programming (SDP) Hajek et al. (2016); Abbe et al. (2016); Fei and Chen (2020) and spectral algorithms Wang et al. (2020); Abbe (2018) have also been proven to be successful to recover the communities of the $\texttt{SBM}$ in various settings.

However, besides SDPs, all those works crucially rely on the degree homogeneneity of the $\texttt{SBM}$, and on the fact that all nodes are structurally similar (e.g.: of same degree). SDP algorithms are robust to some forms of variations, but they have prohibitive time complexity. The fastest algorithm recovering communities up to the optimal ratio $(p-q)/\sqrt{p}$ in the $\texttt{SBM}$ is nearly-linear, from Wang et al. (2020).

Several combinatorial algorithms run in linear time, as Cohen-Addad et al. (2020); Condon and Karp (2001); Carson and Impagliazzo (2001), but they require $(p-q)/\sqrt{p}$ to be polynomially larger than the optimal threshold or require knowledge of p, q.

Hence, for the $\texttt{SBM}$ we obtain the first linear-time algorithm that works up to the asymptotic optimal ratio $(p-q)/\sqrt{p}$.

Degree Heterogeneous Models There are a few extension of the $\texttt{SBM}$ that allow for some variety in the degrees. Most notably, the Degree Corrected Block Model ($\texttt{DCBM}$), the Inhomogeneous Model ($\texttt{IM}$) and the Heterogeneous $\texttt{SBM}$ ($\texttt{HeSBM}$). On the algorithmic side, it is worth noting that many of the spectral algorithms that work well in the $\texttt{SBM}$ fail in such models due to the massive changes in the eigenvalues induced by the heterogeneous degrees, as shown in Chung et al. (2003); Mihail and Papadimitriou (2002); Gulikers et al. (2017b) We first mention the work on the Degree Corrected Stochastic Block Model, as defined in Karrer and Newman (2011). In this model, every node u is assigned to a community and has a weight θ_u. There is an edge between two vertices u, v with probability $\theta_u\theta_v \cdot p$ if they are in the same community, with probability $\theta_u\theta_v \cdot q$ otherwise. θ_u controls therefore the degree of node u, and allows heterogeneity in the degree distribution. However, all the algorithms for this model that we are aware of have strong restrictions on the values of θ: the average degree may be assumed polynomial in n as in Chaudhuri et al. (2012), or the θ_u be within a constant factor, as in Qin and Rohe (2013); Gulikers et al. (2017a); θ_u are sometimes i.i.d distributed from a distribution that has constant variance, see Dall'Amico et al. (2019). A comprehensive study of the limitations of standard spectral algorithms for that model can be found in Gulikers et al. (2017b). Their paper provides an algorithm working on more general degree distribution, but still require the lowest and highest degree to be somewhat close to the

average.[1] Hence, in the DCBM, even though the vertices do not have exactly the same expected degree, they are required to be concentrated and cannot be arbitrary as in our results.

In the Inhomogeneous Model Bollobás et al. (2007), the probability for each edge $p_{i,j}$ of being present can be freely chosen. In some sense, DHSBM is a specialization of the Inhomogeneous Model to enforce ground-truth communities. Without this restriction, there is no community to recover, hence the problems considered in this model are very different from community detection.

The Heterogeneous SBM (HeSBM) Jalali et al. (2016) is the closest to our DHSBM. The HeSBM allows a different p_u for each community. However, there are three major differences. First, in the HeSBM the interconnectivity density p_u is the same for all nodes of the same community. Second, the interconnectivity density is the same for all communities q (and so all nodes). In comparison, the DHSBM allows different p_u, q_u for each node. On other hand, the HeSBM has no restriction on the sizes of communities. The authors of Jalali et al. (2016) give a semidefinite program (SDP) in which regime they can recover the communities. Their requirements are too involved to state here, but for the equi-sized two community setting with all p_u being the same, they show that their algorithm works for a ratio $(p-q)/\sqrt{p}$ similar to ours. Note that the SDP has a polynomial runtime, whereas our algorithm only requires linear time.

1.4. Model and Notation

We now define the *Degree-Heterogeneous Stochastic Block Model*.

Definition 4 (DHSBM) *Given integer N, a real $f \geq 2$ and probability vectors $\mathbf{p} = \{p_u \mid u \in [N]\}$ and $\mathbf{p} = \{q_u \mid u \in [N]\}$ a random graph G on N nodes is generated from the* DHSBM *as follows.*

1. *The nodes are partitioned into 2 communities C_1, C_2, with $|C_i| > N/f$.*
2. *For every pair of nodes $u, v \in V$ such that either both u and v are in C_i for some i, an arc (u, v) is created w.p. p_u.*
3. *For every pair of nodes $u, v \in V$ such that $u \in C_i$ and $v \notin C_i$, an arc (u, v) is created w.p. q_u.*[2]

We refer to the above model as $\mathtt{DHSBM}(N, f, \mathbf{p}, \mathbf{q})$.

We refer to C_i as the *communities* or ground-truth partition. Let $\mathrm{B}(p)$ denote the Bernoulli distribution with parameter p. We use $\mathrm{BIN}(n, p)$ to denote the binomial distribution with parameters n and p. For a random variable X and a probability distribution $\mathcal{D}$, $X \sim \mathcal{D}$ means that $\mathcal{D}$ is the probability distribution of X. We use the symbol $=_d$ to say that two random variables have the same distribution, e.g., for $X \sim \mathrm{B}(p)$ and $Y \sim \mathrm{B}(p)$, we have $X =_d Y$.

1. More precisely, their constraint (2.2) enforces high-degree vertices to be close to average, while (2.8) restricts the lowest degree.
2. The meaning of p_u and q_u is switched compared to the introduction, where p_u was the probability v is connected to u. This was easier for explaining the Twitter example, while this definition makes the notations easier. The two are equivalent up to reorienting each edge.

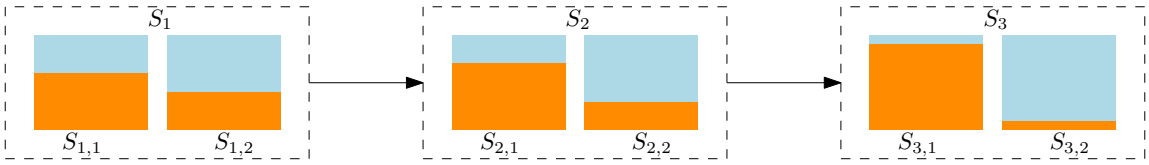

Figure 1: Illustration of algorithm Update

2. Description of the Algorithm

We present in this section the algorithm recovering the communities. To unify the directed and undirected model, all (undirected) edges are simply replaced by two arcs going in opposite directions.

The crux of algorithm ExactRecovery is the procedure Update, that takes as input a partitioned subset of vertices, and uses it to partition better another subset of vertices. We henceforth start by describing Update.

Algorithm Update takes as an input two subsets of vertices S^i, S^j, with S^i partitioned into $\left(S_1^i, S_2^i\right)$. Its goal is to produce a partition $\left(S_1^j, S_2^j\right)$ of S^j, that is closer to the ground-truth than the partition of S^i. For that, the general idea is to classify nodes of S^j according to the number of neighbors they have in S_1^i and S_2^i, as described in Section 1.2. See Figure 1 for an illustration where $i = 1$ and $j = 2$.

For that comparison to be meaningful, it is necessary to proceed first to a normalization. This is the first step of Algorithm Update: find subsets of S_1^i and S_2^i that have equal size. For $\ell = 1, 2$, let $\tilde{S}_\ell^i$ be a random subset of S_ℓ^i such that $\left|\tilde{S}_1^i\right| = \left|\tilde{S}_2^i\right| = \min_\ell \left|S_\ell^i\right|$. Then, each node $u \in S^j$ is then assigned to S_1^j or S_2^j according to the number of edges it has to $\tilde{S}_1^i, \tilde{S}_2^i$: if u has more edges towards $\tilde{S}_\ell^i$, it is assigned to S_ℓ^j.

We show in Lemma 5 that this procedure ensures that many more nodes of S^j are correctly classified in the partition (S_1^j, S_2^j) than in the partition of S^i: each call to update thus improves substantially the partition, see Figure 1.

Let us now describe our main algorithm, ExactRecovery. It has two phases: one that finds an almost correct partition, and another one that turns an almost correct partition into a correct one. More precisely, the algorithm splits the nodes randomly into two equal sized sets S and S'. The first phase computes an almost correct partition of S and S' looking only at edges within each sets, while the second uses the almost correct partition of S' (resp. S) to correctly classify S (resp. S').

The first phase works as described previously: the algorithm partitions vertices into parts $S_1, ..., S_b$ of equal size $n = N/b$, and at each time t finds a set $S_{i_{t+1}}$ such that $S_{i_{t+1}}$ has not been compared yet to S_{i_t}. Using the procedure Update, the algorithm partitions $S_{i_{t+1}}$ using the partition of S_{i_t}.

At the end of the first phase, the algorithm has a partition of each S and S', found by merging the partitions of all S_i. Note that the edges between S and S' have not been considered by the algorithm, and thus those partitions are independent: this is the reason why we partition into two halves, allowing us to save some independence and boost the success probabilities.

In phase 2, the algorithm simply classifies all nodes of S according to their edges to S', and vice-versa. This cleans up the partition found by the first phase, and ensures all nodes are correctly classified. The output is the union of those two partitions.

Algorithm 1 Update(S_i, S_j)
—**Input:** Two sets S_i, S_j, where S_i is partitioned into two subsets $S_{i,1}$ and $S_{i,2}$
—**Output:** A partition of S_j into two subsets of $S_{j,1}, S_{j,2}$, where $S_{j,1}$ is associated with $S_{i,1}$ and $S_{j,2}$ associated with $S_{i,2}$.

Initialize $S_{j,1} = S_{j,2} = \emptyset$.
If $|S_{i,1}| > |S_{i,2}|$: Let $\tilde{S}_{i,1}$ be a subset of $S_{i,1}$ of size $|S_{i,2}|$ chosen uniformly at random, and $\tilde{S}_{i,2} = S_{i,2}$
Else: Let $\tilde{S}_{i,1} = S_{i,1}$, and $\tilde{S}_{i,2}$ be a subset of $S_{i,2}$ of size $|S_{i,1}|$ chosen uniformly at random.
For every $v \in S_j$ do the following:
    If v has strictly more arcs to $\tilde{S}_{i,1}$ than $\tilde{S}_{i,2}$, assign v to $S_{j,1}$
    If v has strictly more arcs to $\tilde{S}_{i,2}$ than $\tilde{S}_{i,1}$, assign v to $S_{j,2}$
    Else, assign v randomly
Output $(S_{j,1}, S_{j,2})$

Algorithm 2 Phase1 (G)
— **Input:** Graph G
—**Output:** Partition S^*

Set $b = \begin{cases} \lceil \sqrt{\log N} \rceil & \text{if } \lceil \sqrt{\log N} \rceil \text{ is odd} \\ \lceil \sqrt{\log N} \rceil + 1 & \text{otherwise} \end{cases}$
Divide the nodes of S u.a.r. into subsets $S_1, S_2, \ldots, S_b$ of equal size
For all $i \leq b$: Split S_i into two halves $S_{i,1}$ and $S_{i,2}$
Let $h = 1$
while there exist i such that neither Update(S_h, S_i) nor Update(S_i, S_h) has been called **do**
        Update(S_h, S_i)
        $h \leftarrow i$
end while
Return partition (S_1^*, S_2^*) with $S_1^* = \cup_{i \leq b} S_{i,1}$ and $S_2^* = \cup_{i \leq b} S_{i,2}$

Algorithm 3 Phase2 ($G, U, (S_1, S_2)$)
—**Input:** Graph G, set U and a partition (S_1, S_2)
—**Output:** Partition P of U using (S_1, S_2)

Initialise $P_1, P_2 = \emptyset$
If $|S_1| > |S_2|$: Let $\tilde{S}_1$ be a subset of S_1 of size $|S_2|$, and $\tilde{S}_2 = S_2$.
Else: Let $\tilde{S}_2$ be a subset of S_2 of size $|S_1|$, and $\tilde{S}_1 = S_1$.
In parallel, for every node $u \in U$
    If u has strictly more arcs to $\tilde{S}_2$ than $\tilde{S}_1$, assign u to P_2
    Else, if u has strictly more arcs to $\tilde{S}_1$ than $\tilde{S}_2$, assign u to P_1
    Else, assign u randomly
Return partition $P' = (P_1, P_2)$

Algorithm 4 ExactRecovery(G)
—**Input:** Graph G on N nodes
—**Output:** Partition P

1: Partition the nodes randomly into two sets S and S' of equal size.
2: Let G_S be the subgraph induced by S and $G_{S'}$ be the subgraph induced by S'
3: $(S_1, S_2) \leftarrow$ Phase1 (G_S)
4: $(S'_1, S'_2) \leftarrow$ Phase1 $(G_{S'})$
5: $(S^*_1, S^*_2) \leftarrow$ Phase2 $(G, S, (S'_1, S'_2))$
6: $(S'^*_1, S'^*_2) \leftarrow$ Phase2 $(G, S', (S_1, S_2))$
7: Return merged partition $(S^*_1 \cup S'^*_1, S^*_2 \cup S'^*_2)$ of S'^* and S'^*

3. Analysis of ExactRecovery

We start by giving an overview followed by the analysis of the algorithm Update. Then, in Section 3.2 we analyze algorithm Phase1, showing how the calls to Update increase progressively the quality of the partition. Then, in Section 3.3, we analyze algorithm Phase2, showing that the partition obtained in the first phase is precise enough to allow for an exact recovery.

3.1. Notations and Overview of the Proof

We start by introducing some notation. We let $S^{(t)}, S^{(t+1)}$ be the input of the t-th call to Update, with $(S_1^{(t)}, S_2^{(t)})$ being the partition of $S^{(t)}$. We denote $C_i^{(t)} = S_i^{(t)} \cap S^{(t)}$ the part of community i that is in the set $S^{(t)}$. For $i, j \in \{1, 2\}$, denote $S_{i,j}^{(t)} = S_i^{(t)} \cap C_j$ and let $s_{i,j}^{(t)} := |S_{i,j}^{(t)}|$. We also define $\alpha^{(t)}$ and $\beta^{(t)}$ such that $|C_1 \cap S_1^{(t)}| = |C_1^{(t)}| \left(1/2 + \alpha^{(t)}\right)$ and that $|C_2 \cap S_2^{(t)}| = |C_2^{(t)}| \left(1/2 + \beta^{(t)}\right)$.

The algorithm subsamples the largest part. Recall the notations from the algorithm: in the case where $|S_1^{(t)}| > |S_2^{(t)}|$, let $\widetilde{S}_1^{(t)} \subseteq S_1^{(t)}$ be a set of size $|S_2^{(t)}|$ drawn uniformly at random from $S_1^{(t)}$, and $\widetilde{S}_2^{(t)} = S_2^{(t)}$. In the case where $|S_1^{(t)}| > |S_2^{(t)}|$, $\widetilde{S}_1^{(t)}$ and $\widetilde{S}_2^{(t)}$ are defined symmetrically. We also extend the notations to $\widetilde{S}_{i,j}^{(t)}$ and $\tilde{s}_{i,j}^{(t)}$. We also write $\mathbb{P}_{\mathcal{A}}[\cdot]$ to emphasize that the probability is over the random choices made by the algorithm. Similarly, define $\mathbb{P}_{\mathcal{G}}[\cdot]$ for randomness that stems from the graph. Lastly, the graph is drawn either from $\mathtt{DHSBM}(N, f, \mathbf{p}, \mathbf{q})$ or $\mathtt{SBM}(N, C_1, C_2, p, q)$, with $\min_u \frac{p_u - q_u}{\sqrt{p_u}} \geq C f^{5/2} \sqrt{\frac{\log N}{N}}$, and n is defined to be N/b, where b is defined in algorithm 2.

The general idea behind our analysis is to track the values of $\alpha^{(t)}$ and $\beta^{(t)}$ throughout different stages, to show that they increase drastically.

In that goal, a key quantity is the *discrepancy* of the partition of $S^{(t)}$ – namely, the number of correctly classified vertices minus the number of misclassified ones, in the subsampled partition: $\Delta^{(t)} = \tilde{s}_{1,1}^{(t)} - \tilde{s}_{2,1}^{(t)}$. We can relate the probability that a node $u \in C_1^{(t+1)}$ is placed into $S_1^{(t+1)}$ –i.e., the probability of making a good choice – to the discrepancy of the partition of $S^{(t)}$:

Lemma 5 *There exists an absolute constant c such that the following holds. Consider a graph drawn either from $G \sim \mathtt{DHSBM}(N, f, \mathbf{p}, \mathbf{q})$ or $G \sim \mathtt{SBM}(N, C_1, C_2, p, q)$. Let $\Delta^{(t)} = \tilde{s}_{1,1}^{(t)} - \tilde{s}_{2,1}^{(t)}$. Then, for any vertex of $u \in C_1^{(t+1)}$, the probability on the graph randomness that u is assigned to $S_1^{(t+1)}$ by the algorithm is at least $1/2 + c \min\left(1, \frac{\Delta^{(t)}(p_u - q_u)}{\sqrt{np_u}}\right)$.*

The idea of the proof is to establish a coupling between the random variables X and X' that govern the number of arcs a node of $C_1^{(t+1)}$ has to $S_1^{(t)}$ and to $S_2^{(t)}$, respectively. We have $X \sim \text{BIN}\left(\tilde{s}_{1,1}^{(t)}, p\right) + \text{BIN}\left(\tilde{s}_{1,2}^{(t)}, q\right)$ and $X' \sim \text{BIN}\left(\tilde{s}_{2,1}^{(t)}, p\right) + \text{BIN}\left(\tilde{s}_{2,2}^{(t)}, q\right)$, and the goal is to show $\mathbb{P}_{\mathcal{G}}[X > X'] \geq 1/2 + \delta$.

The following lemma is key to that. It allows us to decompose X into three binomials Y_1, Y_2 and Y_3, such that $X = Y_1 + Y_2 + Y_3$ and X' follows the same law as $Y_1 + Y_2$.

Lemma 6 *Fix a step t. Consider the three independent random variables X, Y_1, Y_2, with distributions*

$$X \sim \text{BIN}\left(\tilde{s}_{1,1}^{(t)}, p\right) + \text{BIN}\left(\tilde{s}_{1,2}^{(t)}, q\right), Y_1 \sim \text{BIN}\left(\tilde{s}_{2,1}^{(t)}, p\right) + \text{BIN}\left(\tilde{s}_{1,2}^{(t)}, q\right), Y_2 \sim \text{BIN}\left(\tilde{s}_{2,2}^{(t)} - \tilde{s}_{1,2}^{(t)}, q\right)$$

Consider also the variable Y_3, independent from X and Y_1, with distribution conditioned on Y_2 $Y_3 \sim \text{BIN}\left(\tilde{s}_{1,1}^{(t)} - \tilde{s}_{2,1}^{(t)} - Y_2, \frac{p-q}{1-q}\right)$.

Those variables are well-defined, and X follows the same law as $Y_1 + Y_2 + Y_3$. In symbols,

$$X =_d Y_1 + Y_2 + Y_3 \tag{1}$$

Since $X' =_d Y_1 + Y_2$, we get $\mathbb{P}[Y_1 + Y_2 \geq X'] = 1/2$. The additional δ is then obtained by bounding $\mathbb{P}[Y_3 > X' - X | X' > Y_1 + Y_2]$. Note that Y_1, Y_2 and Y_3 are in contrast to X simple binomials and therefore obtaining bounds for them is much easier than obtaining bounds for the sum of binomials X. Our proof uses that each binomial involved can be approximated by a variable with a simple probability distribution: when σ is the standard deviation of the binomial, we approximate the distribution with one equal to $1/\sigma$ on an interval of length $\Omega(\sigma)$. This essentially allows us to "decouple" the variable Y_3 from Y_1 and Y_2, and express $Y_3 > X' - X$ knowing $X' > Y_1 + Y_2$ into simpler events with probability easier to compute.

Lemma 5 relates the probability of improving the partition to the discrepancy: it is therefore necessary to control that quantity precisely. Note that it is computed *after* the subsampling in order to compensate for potential size imbalance between the parts of a group: otherwise, its value would be too impacted by the size of each part. $\Delta^{(t)}$ is therefore a random variable. We can show that the expectation of $\Delta^{(t)}$ is crucially related to $\alpha^{(t)}$ and $\beta^{(t)}$: this will be helpful, as the growth of α and β ensured by Lemma 5 will enforce the growth of Δ as well. Furthermore, $\Delta^{(t)}$ is tightly concentrated around that expectation.

Lemma 7 *Let $\Delta^{(t)} := \tilde{s}_{2,2}^{(t)} - \tilde{s}_{1,2}^{(t)}$. Then, $\Delta^{(t)} = \tilde{s}_{1,1}^{(t)} - \tilde{s}_{2,1}^{(t)}$, and, for any $x > 0$,*

$$\mathbb{P}_{\mathcal{A}}\left[\Delta^{(t)} \geq \frac{|C_1^{(t)}| \cdot |C_2^{(t)}|}{\max(|S_1^{(t)}|, |S_2^{(t)}|)}(\alpha^{(t)} + \beta^{(t)}) - x\right] \geq 1 - \exp\left(-\frac{2x^2}{\min\left(|S_1^{(t)}|, |S_2^{(t)}|\right)}\right).$$

3.2. Analysis of Algorithm Phase1

Phase1 repeatedly calls Update to improve the partition. Using Lemma 5, we show by induction that Algorithm Phase1 produces a partition that is mostly correct. All those lemmas apply to a graph drawn from $\texttt{DHSBM}(N, f, \mathbf{p}, \mathbf{q})$. For that, we first show the effect of our initializations steps. The following lemma shows that each community is fairly well represented in each of the subsets $S_1, ..., S_b$, and provides a lower bound on the initial discrepancy:

Lemma 8 *With high probability (w.r.t. to the randomness of the algorithm), the communities are almost evenly distributed in the subsets: for $i \in \{1,2\}$, it holds that*

$$\mathbb{P}_{\mathcal{A}}\left[\forall \ell \in [1, \sqrt{\log N}],\ |C_i \cap S_\ell| \geq \frac{|C_i|}{2\sqrt{\log N}}\right] \geq 1 - 1/N.$$

Furthermore, there exist a constant κ such that

$$\mathbb{P}_{\mathcal{A}}\left[|\alpha^{(0)}| \geq \frac{1}{8f^{3/2}\sqrt{2n}}\right] \geq 1 - \frac{1}{8f} - \frac{8\kappa\sqrt{f}}{\sqrt{n}}.$$

Without loss of generality, we will assume $\alpha^{(0)} \geq 0$ in the following. This implies $\beta^{(0)} > 0$, as by design the sets $S_1^{(0)}$ and $S_2^{(0)}$ have equal size (see Fact 3 in Appendix). Assuming we have a good initialization (Lemma 8), we can use our bounds one the probability of assigning nodes correctly (Lemma 5) to argue that $\alpha^{(t)}, \beta^{(t)}$ follows geometric series.

Lemma 9 *Assume that $\gamma \geq Cf^{5/2}\sqrt{\frac{\log N}{N}}$. Let c be the constant from Lemma 5.*

It holds that, after t iterations of the algorithm, $\alpha^{(t)}, \beta^{(t)} > \min(c/2, \log(N)^{t/4}/\sqrt{8n})$ with probability $1 - \frac{1}{8f} - \sum_{i=1}^{t} \exp\left(O\left(-\frac{\log(N)^{t/2}}{f^3}\right)\right)$ on the algorithm's randomness, and probability $1 - \sum_{i=0}^{t} \max\left(\exp\left(-\frac{\log(N)^{(i+1)/2}}{4f}\right), \exp\left(-\frac{c^2 N}{16\sqrt{\log N}}\right)\right)$ on the graph.

The proof of Lemma 9 relies on Lemma 5, Chernoff bound and the relationship between the discrepancy Δ and the value of α, β. One key part here is the independence between calls of Update, ensured by the design of our algorithm: no edge will be looked at twice, and therefore the probabilities from Lemma 5 are all independent.

To conclude the analysis of Phase1, it only remains to argue that there are enough calls. Concretely, we show that the algorithm calls Update $\binom{b}{2}$ times by relating the calls of Update to Euler paths.

Lemma 10 *The algorithm* ExactRecovery *performs $\binom{b}{2}$ updates.*

The following corollary summarizes the outcome of phase one.

Corollary 11 *Let $c > 0$ be the constant from Lemma 5. At the end of the first phase, we have that obtained partition $P^* = (S_1^*, S_2^*)$ satisfies either $|C_1 \cap S_1^*| \geq |C_1|(1/2 + c/2)$ or $|C_1 \cap S_2^*| \geq |C_1|(1/2 + c/2)$ w.p. at least $1 - 9/(80f)$ over the algorithm's randomness and w.p. $1 - o(1)$ over the randomness of the graph.*

Proof Lemma 9 shows that the algorithm quickly increases the value of α, β. Lemma 10 shows that $t_{max} \geq \binom{\sqrt{\log N}}{2}$ comparison will be done, resulting in values of α, β at least $c/2$ for the final sets, where c is the constant from Lemma 5.

More precisely, $\alpha^{(t)}$ is at least $c/2$ when $t \geq 4\log(n/c)/\log\log N$: hence, the set considered in at least half of the time steps will have $\alpha > c/2$. Those time steps must involve at least $\sqrt{\log N}/2$ many sets, as an Eulerian walk on fewer sets visits at most $\log N/4$ of them. Hence, the merged partition ensures that $P^* = (S_1^*, S_2^*)$ satisfies either $|C_1 \cap S_1^*| \geq |C_1|(1/2 + c/2)$ or $|C_1 \cap S_2^*| \geq |C_1|(1/2 + c/2)$. ■

3.3. Analysis of Algorithm Phase2

Using results from the previous section, we now analyze the algorithm Phase2, and show it leads to an exact recovery in *DHSBM*. Our reduction of *SBM* to *DHSBM* allows to extend directly the result to the *SBM*.

Lemma 12 *Let S, S' be two equal-size subsets of the vertices of G. Fix a partition $P_{S'} = (P_1, P_2)$ of S', with $|C_i \cap P_i| \geq |C_i \cap S'|(1/2 + c/2)$ for the constant c given in Lemma 5. The following holds with probability at least $1 - 2\exp(-100 \log N)$ on the graph and $1 - \exp\left(-\frac{N\kappa^2}{2f}\right)$ on the algorithm. For any vertex $v \in S \cap C_i$, v has more edges to $\tilde{S}_i$ than to the other part ($\tilde{S}_1$ or $\tilde{S}_2$), where $\tilde{S}_i$ are subsamples of S_i as done in algorithm Phase2.*

The proof of that lemma relies on concentration bounds on the number of edges a vertex has to each side of the cut. A direct corollary of that lemma is that Phase2 of ExactRecovery identifies correctly the two communities:

Corollary 13 *Under the assumptions of Lemma 12, Phase2 leads to an exact recovery with probability $1 - \exp(-Nc^2/f^3)$ on the algorithm and $1 - 1/N^3$ on the graph.*

Combining Corollary 11 and Corollary 13 concludes the proof Theorem 1 and Theorem 2.

Acknowledgments

F. Mallmann-Trenn was in part supported by the EPSRC grant EP/W005573/1. This work was [partially] funded by the grant ANR-19-CE48-0016 from the French National Research Agency (ANR).

References

E. Abbe, A. S. Bandeira, and G. Hall. Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62(1):471–487, 2016. doi: 10.1109/TIT.2015.2490670.

Emmanuel Abbe. Community detection and stochastic block models: Recent developments. *Journal of Machine Learning Research*, 18(177):1–86, 2018. URL http://jmlr.org/papers/v18/16-480.html.

Emmanuel Abbe and Colin Sandon. Recovering communities in the general stochastic block model without knowing the parameters. In *NIPS*, 2015.

Emmanuel Abbe, Afonso S Bandeira, and Georgina Hall. Exact recovery in the stochastic block model. *IEEE Transactions on Information Theory*, 62(1):471–487, 2015.

Andrew C Berry. The accuracy of the gaussian approximation to the sum of independent variates. *Transactions of the american mathematical society*, 49(1):122–136, 1941.

Béla Bollobás, Svante Janson, and Oliver Riordan. The phase transition in inhomogeneous random graphs. *Random Structures & Algorithms*, 31(1):3–122, 2007. doi: https://doi.org/10.1002/rsa.20168. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/rsa.20168.

Ted Carson and Russell Impagliazzo. Hill-climbing finds random planted bisections. In *Proc. 12th Symposium on Discrete Algorithms (SODA 01), ACM press, 2001*, pages 903–909, 2001.

Kamalika Chaudhuri, Fan Chung, and Alexander Tsiatas. Spectral clustering of graphs with general degrees in the extended planted partition model. In *Conference on Learning Theory*, pages 35–1. JMLR Workshop and Conference Proceedings, 2012.

Peter Chin, Anup Rao, and Van Vu. Stochastic block model and community detection in sparse graphs: A spectral algorithm with optimal rate of recovery. In *Conference on Learning Theory*, pages 391–423. PMLR, 2015.

Fan Chung, Linyuan Lu, and Van Vu. Spectra of random graphs with given expected degrees. *Proceedings of the National Academy of Sciences*, 100(11):6313–6318, 2003. ISSN 0027-8424. doi: 10.1073/pnas.0937490100. URL https://www.pnas.org/content/100/11/6313.

Vincent Cohen-Addad, Adrian Kosowski, Frederik Mallmann-Trenn, and David Saulpic. On the power of louvain in the stochastic block model. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/29a6aa8af3c942a277478a90aa4cae21-Abstract.html.

Anne Condon and Richard M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Struct. Algorithms*, 18(2):116–140, March 2001. ISSN 1042-9832. doi: 10.1002/1098-2418(200103)18:2⟨116::AID-RSA1001⟩3.0.CO;2-2. URL http://dx.doi.org/10.1002/1098-2418(200103)18:2<116::AID-RSA1001>3.0.CO;2-2.

Lorenzo Dall'Amico, Romain Couillet, and Nicolas Tremblay. Revisiting the bethe-hessian: Improved community detection in sparse heterogeneous graphs. In *Annual Conference on Neural Information Processing Systems 2019*, pages 4039–4049, 2019.

Y. Fei and Y. Chen. Achieving the bayes error rate in synchronization and block models by sdp, robustly. *IEEE Transactions on Information Theory*, 66(6):3929–3953, 2020. doi: 10.1109/TIT.2020.2966438.

Amir Gandomi and Murtaza Haider. Beyond the hype: Big data concepts, methods, and analytics. *IJIM*, 35(2):137–144, 2015.

Chao Gao, Zongming Ma, Anderson Y. Zhang, and Harrison H. Zhou. Achieving optimal misclassification proportion in stochastic block models. *Journal of Machine Learning Research*, 18(60):1–45, 2017. URL http://jmlr.org/papers/v18/16-245.html.

Dmitry Gavinsky, Shachar Lovett, Michael Saks, and Srikanth Srinivasan. A tail bound for read-k families of functions. *Random Struct. Algorithms*, 47(1):99–108, August 2015. ISSN 1042-9832. doi: 10.1002/rsa.20532. URL http://dx.doi.org/10.1002/rsa.20532.

Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.

Lennart Gulikers, Marc Lelarge, and Laurent Massoulié. Non-Backtracking Spectrum of Degree-Corrected Stochastic Block Models. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference (ITCS 2017)*, volume 67 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 44:1–44:27, Dagstuhl, Germany, 2017a. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. ISBN 978-3-95977-029-3. doi: 10.4230/LIPIcs.ITCS.2017.44. URL http://drops.dagstuhl.de/opus/volltexte/2017/8179.

Lennart Gulikers, Marc Lelarge, and Laurent Massoulié. A spectral method for community detection in moderately sparse degree-corrected stochastic block models. *Advances in Applied Probability*, pages 686–721, 2017b.

B. Hajek, Y. Wu, and J. Xu. Achieving exact cluster recovery threshold via semidefinite programming: Extensions. *IEEE Transactions on Information Theory*, 62(10):5918–5937, 2016. doi: 10.1109/TIT.2016.2594812.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. In *The collected works of Wassily Hoeffding*, pages 409–426. Springer, 1994.

Amin Jalali, Qiyang Han, Ioana Dumitriu, and Maryam Fazel. Exploiting tradeoffs for exact recovery in heterogeneous stochastic block models. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29, pages 4871–4879. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper/2016/file/57bafb2c2dfeefba931bb03a835b1fa9-Paper.pdf.

Brian Karrer and Mark EJ Newman. Stochastic blockmodels and community structure in networks. *Physical review E*, 83(1):016107, 2011.

S Lahiri and A Chatterjee. A berry-esseen theorem for hypergeometric probabilities under minimal conditions. *Proceedings of the American Mathematical Society*, 135(5):1535–1545, 2007.

David Liben-Nowell and Jon Kleinberg. The link-prediction problem for social networks. *Journal of the American society for information science and technology*, 58(7):1019–1031, 2007.

Christopher D Manning, Christopher D Manning, and Hinrich Schütze. *Foundations of statistical natural language processing*. MIT press, 1999.

Frank McSherry. Spectral partitioning of random graphs. In *Proceedings 2001 IEEE International Conference on Cluster Computing*, pages 529–537. IEEE, 2001.

Milena Mihail and Christos Papadimitriou. On the eigenvalue power law. In José D. P. Rolim and Salil Vadhan, editors, *Randomization and Approximation Techniques in Computer Science*, pages 254–262, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg. ISBN 978-3-540-45726-8.

Nina Mishra, Robert Schreiber, Isabelle Stanton, and Robert E Tarjan. Clustering social networks. In *International Workshop on Algorithms and Models for the Web-Graph*, pages 56–67. Springer, 2007.

Ankur Moitra, William Perry, and Alexander S. Wein. How robust are reconstruction thresholds for community detection? In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the*

48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016, pages 828–841. ACM, 2016. doi: 10.1145/2897518.2897573. URL https://doi.org/10.1145/2897518.2897573.

Elchanan Mossel, Joe Neeman, and Allan Sly. Consistency thresholds for the planted bisection model. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 69–75. ACM, 2015. doi: 10.1145/2746539.2746603. URL https://doi.org/10.1145/2746539.2746603.

Tai Qin and Karl Rohe. Regularized spectral clustering under the degree-corrected stochastic blockmodel. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, pages 3120–3128. Curran Associates, Inc., 2013. URL https://proceedings.neurips.cc/paper/2013/file/0ed9422357395a0d4879191c66f4faa2-Paper.pdf.

Peng Wang, Zirui Zhou, and Anthony Man-Cho So. A nearly-linear time algorithm for exact community recovery in stochastic block model. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10126–10135. PMLR, 13–18 Jul 2020. URL http://proceedings.mlr.press/v119/wang20ac.html.

Anderson Y. Zhang and Harrison H. Zhou. Theoretical and computational guarantees of mean field variational inference for community detection. *Ann. Statist.*, 48(5):2575–2598, 10 2020. doi: 10.1214/19-AOS1898. URL https://doi.org/10.1214/19-AOS1898.

Contents

Appendix A. Auxiliary Claims

Theorem 14 (Gavinsky et al. (2015)) *We say a family $Y_1, ..., Y_r$ of indicator variables is* read-k *if there there exists a sequence $X_1, ..., X_m$ of independent variables and a sequence $S_1, ..., S_r$ of subsets of $\{1, ..., m\}$ such that*

- *Y_i is a function of $\{X_j, j \in S_i\}$*
- *no element of $\{1, ..., m\}$ appears in more than k of the S_i's*

For any sequence of read-k variables we have that $\mathbb{P}[Y_1 + ... + Y_r > \mathbb{E}[Y_1 + ... + Y_r] + \gamma r]$ and $\mathbb{P}[Y_1 + ... + Y_r < \mathbb{E}[Y_1 + ... + Y_r] - \gamma r]$ are bounded by $\exp(-2\gamma^2 r/k)$.

Lemma 15 *Let $\tau = 1 + 1/\sqrt{2\pi}$. There exists a constant ℓ such that, for any binomially distributed random variable X with variance σ^2, expectation μ, and for all $i \in [\mu - \frac{\tau}{8}\sigma, \mu + \frac{\tau}{8}\sigma]$,*

$$\mathbb{P}[X = i] \geq \frac{1}{8\sigma}.$$

Furthermore, for all integers i, $\mathbb{P}[X = i] \leq \frac{\tau}{\sigma}$.

Proof We start by showing the upper bound: that is,

$$\forall i, \mathbb{P}[X = i] \leq \frac{\tau}{\sigma}$$

We use for this Esseen's inequality (Theorem 16). The binomial distribution is the sum of n i.i.d Bernoulli random variables $\sum_i X_i$ such that $\mu_3(X_1)/\mu_2(X_1) = \frac{p(1-p)(1-2p+2p^2)}{p(1-p)} \leq 1$. So, using Theorem 16, for any i:

$$\begin{aligned}
\mathbb{P}[X=i] &= \mathbb{P}[X \leq i] - \mathbb{P}[X \leq i-1] \\
&\leq \mathbb{P}[N(\mu,\sigma) \in [-\infty, i]] - \mathbb{P}[N(\mu,\sigma) \in [-\infty, i-1]] + 2 \cdot \frac{0.4748}{\sigma} \\
&\leq \mathbb{P}[N(\mu,\sigma) \in [i-1, i]] + \frac{1}{\sigma} \\
&\leq \frac{1}{\sigma\sqrt{2\pi}} \int_{i-1}^{i} \mathbb{P}[N(\mu,\sigma) = t]\, dt + \frac{1}{\sigma} \\
&\leq \frac{1}{\sigma\sqrt{2\pi}} + \frac{1}{\sigma} = \frac{\tau}{\sigma},
\end{aligned}$$

since $\tau = \frac{1}{\sqrt{2\pi}} + 1$. Define $\ell = \tau/8$. Let m be the mode of X, i.e., $\operatorname{argmax}\mathbb{P}[X=i]$. The probability density function of $\mathcal{D}$ is monotonically increasing for $k < m$ and monotonically decreasing for $k > m$. Consider the interval $\mathcal{I} = [\mu - 2\sigma, \mu + 2\sigma]$. By the Chebyshev's inequality, the interval $\mathcal{I}$ has a probability mass of at least $\frac{3}{4}$.

Now, if there are more than $2\ell\sigma$ integers i such that $\mathbb{P}[X=i] \geq \frac{1}{8\sigma}$, then the lemma follows. Assume therefore it is not the case.

Since the maximal probability is τ/σ, for all $x \in [\mu - \ell\sigma, \mu + \ell\sigma]$ then $\mathbb{P}[X=i] \leq \tau/\sigma$. For all $x \in \mathcal{I} \setminus [\mu - \ell\sigma, \mu + \ell\sigma]$, our assumptions implies that $\mathbb{P}[X=i] \leq \frac{1}{8\sigma}$.

Hence, the probability mass in $\mathcal{I}$ is at most

$$\frac{c}{\sigma} 2\ell\sigma + \frac{1}{8\sigma}(4 - 2\ell)\sigma < \frac{1}{4} + \frac{1}{8}4 = \frac{3}{4},$$

a contradiction. ■

The following is a slightly weaker version of Theorem 1 in Berry (1941).

Theorem 16 (Esseen inequality Berry (1941)) *Let $\mu_k(X)$ denote the kth absolute central moment $\mu_k = \int |x - \mathbb{E}[X]|^k \mathbb{P}[X=x]\, dx$. Let $X_1, \ldots, X_n$ be a collection of n random variables, with $\mu_2(X_i) > 0$ for all i. Let $X = \sum_i X_i$. Let $\mu = \mathbb{E}[X]$ and $\sigma^2 = \sum_i \operatorname{Var}[X_i]$. Let $F(\cdot)$ be the cumulative density function of X and let $G(\cdot)$ be the cdf of $\mathcal{N}(\mu, \sigma^2)$. Then,*

$$\sup_{-\infty < x < \infty} |F(x) - G(x))| \leq \frac{0.4748}{\sigma} \max_i \frac{\mu_3(X_i)}{\mu_2(X_i)}.$$

Theorem 17 (Concentration of hypergeometric distribution, see Hoeffding (1994)) *Let X be a random variable such that $\mathbb{P}[X=k]$ is the probability of drawing k red balls in n draws, without replacement, from a set of N balls that contains exactly K red ones. Then, for any $x \in (1, nK/N]$ it holds that*

$$\begin{aligned}
\mathbb{P}[X \leq nK/N - nx] &\leq \exp(-2x^2 n) \\
\mathbb{P}[X \geq nK/N + nx] &\leq \exp(-2x^2 n).
\end{aligned}$$

Appendix B. Proofs of the Upper Boud

B.1. Proof of Section 3.1

Lemma 7 *Let* $\Delta^{(t)} := \tilde{s}^{(t)}_{2,2} - \tilde{s}^{(t)}_{1,2}$. *Then,* $\Delta^{(t)} = \tilde{s}^{(t)}_{1,1} - \tilde{s}^{(t)}_{2,1}$, *and, for any* $x > 0$,

$$\mathbb{P}_{\mathcal{A}}\left[\Delta^{(t)} \geq \frac{|C_1^{(t)}| \cdot |C_2^{(t)}|}{\max(|S_1^{(t)}|, |S_2^{(t)}|)}(\alpha^{(t)} + \beta^{(t)}) - x\right] \geq 1 - \exp\left(-\frac{2x^2}{\min\left(|S_1^{(t)}|, |S_2^{(t)}|\right)}\right).$$

Proof Since there is no ambiguity, we drop the superscript (t) for simplicity. We also let $c_i = |C_i^{(t)}|$ First note that $\tilde{s}_{2,2} - \tilde{s}_{1,2} = \tilde{s}_{1,1} - \tilde{s}_{2,1}$ since the subsampling ensures that $\tilde{s}_{1,1} + \tilde{s}_{1,2} = \tilde{s}_{2,1} + \tilde{s}_{2,2}$. For the other part, we fist assume that $|S_1| > |S_2|$. Hence, $\tilde{s}_{2,2} = s_{2,2} = c_2(1/2 + \beta)$ and in expectation $\tilde{s}_{1,2} = \frac{|S_2|}{|S_1|} \cdot c_2(1/2 - \beta)$. The random variable $\tilde{s}_{1,2}$ follows a hypergeometric law, therefore by Theorem 17 $\mathbb{P}_{\mathcal{A}}[\tilde{s}_{1,2} \leq \frac{|S_2|}{|S_1|} \cdot c_2(1/2 - \beta) + x] \geq 1 - \exp(-2x^2/|S_2|)$

Hence, the value of Δ is with that probability:

$$\begin{aligned}
\Delta &\geq c_2(1/2 + \beta) - \frac{|S_2|}{|S_1|} \cdot c_2(1/2 - \beta) - x \\
&\geq \frac{c_2}{|S_1|}\left(|S_1|(1/2 + \beta) - |S_2|(1/2 - \beta)\right) - x \\
&\geq \frac{c_2}{|S_1|}\left((c_1(1/2 + \alpha) + c_2(1/2 - \beta)) \cdot (1/2 + \beta) - (c_1(1/2 - \alpha) + c_2(1/2 + \beta)) \cdot (1/2 - \beta)\right) - x \\
&\geq \frac{c_2}{|S_1|}(c_1(1/4 + \alpha/2 + \beta/2 + \alpha\beta - 1/4 - \beta/2 + \alpha/2 - \alpha\beta) \\
&\qquad + c_2(1/4 + \beta/2 - \beta/2 - \beta^2 - 1/4 - \beta/2 + \beta/2 + \beta^2) - x \\
&\geq \frac{c_1 c_2}{|S_1|}(\alpha + \beta) - x.
\end{aligned}$$

When $|S_1| \leq |S_2|$, we use instead $\mathbb{E}_{\mathcal{A}}[\tilde{s}_{2,2}] = \frac{|S_1|}{|S_2|} s_{2,2}$ and $\tilde{s}_{1,2} = s_{1,2}$, so that the second line is $\frac{c_2}{|S_2|}\left(|S_1|(1/2 + \beta) - |S_2|(1/2 - \beta)\right)$. What follows is exactly the same. ■

Lemma 6 *Fix a step* t. *Consider the three independent random variables* X, Y_1, Y_2, *with distributions*

$$X \sim \textsc{Bin}\left(\tilde{s}^{(t)}_{1,1}, p\right) + \textsc{Bin}\left(\tilde{s}^{(t)}_{1,2}, q\right), Y_1 \sim \textsc{Bin}\left(\tilde{s}^{(t)}_{2,1}, p\right) + \textsc{Bin}\left(\tilde{s}^{(t)}_{1,2}, q\right), Y_2 \sim \textsc{Bin}\left(\tilde{s}^{(t)}_{2,2} - \tilde{s}^{(t)}_{1,2}, q\right)$$

Consider also the variable Y_3, *independent from* X *and* Y_1, *with distribution conditioned on* Y_2 $Y_3 \sim \textsc{Bin}\left(\tilde{s}^{(t)}_{1,1} - \tilde{s}^{(t)}_{2,1} - Y_2, \frac{p-q}{1-q}\right)$.

Those variables are well-defined, and X *follows the same law as* $Y_1 + Y_2 + Y_3$. *In symbols,*

$$X =_d Y_1 + Y_2 + Y_3 \tag{1}$$

Proof Since t is fixed, we write $\tilde{s}_{i,j} = \tilde{s}^{(t)}_{i,j}$ for all $i, j \in \{1, 2\}$ and we write $\Delta = \Delta^{(t)}$.

We first show that Y_2 is well-defined, i.e., that $\tilde{s}_{2,2} - \tilde{s}_{1,2} > 0$: in the case where $\tilde{s}_{2,2} = s_{2,2}$, this is because $\tilde{s}_{1,2} \leq s_{1,2} \leq s_{2,2}$. For the other case, we use $\tilde{s}_{2,2} - \tilde{s}_{1,2} = \Delta = \tilde{s}_{1,1} - \tilde{s}_{2,1}$ and $\tilde{s}_{1,1} = s_{1,1}$.

We now show Equation 1. Let $B(x)$ denote the Bernoulli distribution with parameter x. To prove Equation 1, we consider the following process, that allows decomposing $B_1 \sim \mathrm{B}(p)$ into $B_2 + B_3$ where $B_2 \sim \mathrm{B}(q)$ and $B_3 = B_4 - B_2$ with $B_4 \sim \mathrm{B}((p-q)/(1-q))$. We claim that $B_1 =_d B_2 + B_3$. To see this, consider the following generating process which yields a coupling.

- Step 1: Draw a random variable R u.a.r. from $[0, 1]$.
- Step 2: If $R \leq q$ set $B_2 = 1$. Set $B_3 = 0$.
- Step 3: If $R \geq q$, set $B_2 = 0$ and if $R \leq p$ set $B_3 = 1$. Otherwise set $B_3 = 0$.

Note that B_3 depends on B_2 and $B_2 + B_3 \in \{0, 1\}$. First note that $\mathbb{P}[\, R \leq p \mid R > q \,] = (p - q)/(1 - q)$. Thus,

$$\begin{aligned}\mathbb{P}[\, B_2 + B_3 = 1 \,] &= \mathbb{P}[\, R \leq q \,] + \mathbb{P}[\, R \leq p \mid R > q \,]\, \mathbb{P}[\, R > q \,] \\ &= \mathbb{P}[\, R \leq q \,] + \mathbb{P}[\, R \in (p, q] \,] = p = \mathbb{P}[\, B_1 = 1 \,].\end{aligned}$$

Using this composition we can construct a coupling that ensures Equation 1: Both X and X' have as a component $\textsc{Bin}\left(\frac{s}{2} - \Delta, p\right) + \textsc{Bin}\left(\frac{s}{2} - \Delta, q\right)$. Clearly, we can can couple these Bernoulli random variables. It remains that X has 2Δ Bernoulli random variables with parameter p and X' has instead 2Δ Bernoulli random variables with parameter $(p-q)/(1-q)$. For those random variables we can use the above process to couple them. This corresponds to the term Y_3. It follows that that the coupling ensures $\textsc{Bin}(2\Delta, p) = Y_2 + Y_3$. This completes the proof. ■

Lemma 5 *There exists an absolute constant c such that the following holds. Consider a graph drawn either from $G \sim \mathtt{DHSBM}(N, f, \mathbf{p}, \mathbf{q})$ or $G \sim \mathtt{SBM}(N, C_1, C_2, p, q)$. Let $\Delta^{(t)} = \tilde{s}^{(t)}_{1,1} - \tilde{s}^{(t)}_{2,1}$. Then, for any vertex of $u \in C_1^{(t+1)}$, the probability on the graph randomness that u is assigned to $S_1^{(t+1)}$ by the algorithm is at least $1/2 + c \min\left(1, \frac{\Delta^{(t)}(p_u - q_u)}{\sqrt{np_u}}\right)$.*

Proof We use the variables described in Lemma 6. We also write $\Delta = \Delta^{(t)}$, $p = p_u$ and $q = q_u$. First, note that X' and $Y_1 + Y_2$ follow the same law

$$X' =_d Y_1 + Y_2 \tag{2}$$

Due to symmetry, $\mathbb{P}_{\mathcal{G}}[Y_1 + Y_2 = \max\{Y_1 + Y_2, X'\}] = 1/2$. Hence, since $X =_d Y_1 + Y_2 + Y_3$, we get

$$\mathbb{P}_{\mathcal{G}}\left[X \geq X' \wedge Y_1 + Y_2 = \max\{Y_1 + Y_2, X'\}\right] \geq \mathbb{P}_{\mathcal{G}}\left[Y_1 + Y_2 = \max\{Y_1 + Y_2, X'\}\right] \geq 1/2$$

Let $\sigma = \sqrt{\mathrm{Var}[\, Y_{1,2} \,]} = \sqrt{\mathrm{Var}[\, X' \,]}$, and $\mu = \mathbb{E}_{\mathcal{G}}[X'] = \mathbb{E}_{\mathcal{G}}[Y_1 + Y_2]$.

We want to find a lower bound of

$$\mathbb{P}_{\mathcal{G}}\left[Y_1 + Y_2 + Y_3 > X' \mid Y_1 + Y_2 < X'\right] + \frac{\mathbb{P}_{\mathcal{G}}\left[Y_1 + Y_2 + Y_3 = X' \mid Y_1 + Y_2 < X'\right]}{2},$$

which is the probability of moving the vertex to the right cluster. The first idea is the following: with constant probability, X' will be in the interval $\mathbb{E}_{\mathcal{G}}[X'] \pm \ell\sigma(X')$ given by Theorem 15. Moreover, the probability that X' takes any value in that interval is at least $\frac{1}{8\sigma(X')}$. Formally:

$$\begin{aligned}
&\mathbb{P}_{\mathcal{G}}\left[Y_1+Y_2+Y_3 > X' \mid Y_1+Y_2 < X'\right] + \frac{\mathbb{P}_{\mathcal{G}}\left[Y_1+Y_2+Y_3 = X' \mid Y_1+Y_2 < X'\right]}{2} \\
&\geq \frac{1}{2}\cdot \mathbb{P}_{\mathcal{G}}\left[Y_1+Y_2+Y_3 \geq X' \mid Y_1+Y_2 < X'\right] \\
&\geq \frac{1}{2}\sum_{r=\lceil -\ell\sigma(X')\rceil}^{\lfloor \ell\sigma(X')\rfloor} \mathbb{P}_{\mathcal{G}}\left[Y_1+Y_2+Y_3 \geq X' \mid Y_1+Y_2 < X' \wedge X' = \mu + r\right]\cdot \mathbb{P}_{\mathcal{G}}\left[X' = \mu + r\right] \\
&\geq \frac{1}{16\sigma(X')}\sum_{r=\lceil -\ell\sigma(X')\rceil}^{\lfloor \ell\sigma(X')\rfloor} \mathbb{P}_{\mathcal{G}}\left[Y_1+Y_2+Y_3 \geq \mu + r \mid Y_1+Y_2 < \mu + r\right].
\end{aligned}$$

We let $\mathcal{I}_r = [\mu+r-\Delta(p-q), \mu+r)$ when $\Delta(p-q) \geq 1$, and $\mathcal{I}_r = [\mu+r-1, \mu+r)$ otherwise. We also restrict the probability to the event that $Y_1+Y_2 \in \mathcal{I}_r$. In that case, $Y_1+Y_2+Y_3 \geq \mu+r$ is equivalent to $Y_3 \geq \max\left(1, \lfloor \Delta(p-q)\rfloor\right)$, since the variables are integers. Informally speaking, this is helpful since $\mathbb{E}_{\mathcal{G}}[Y_3] = 2\Delta(p-q)$, so $\mathbb{P}_{\mathcal{G}}\left[Y_3 \geq \Delta(p-q)\right]$ will holds with constant probability . We continue the above inequalities:

$$\begin{aligned}
&\geq \frac{1}{16\sigma(X')}\sum_{r=\lceil -\ell\sigma(X')\rceil}^{\lfloor \ell\sigma(X')\rfloor} \mathbb{P}_{\mathcal{G}}\left[Y_3 \geq \max\left(1, \lfloor \Delta(p-q)\rfloor\right) \mid Y_1+Y_2 \in \mathcal{I}_r\right]\cdot \mathbb{P}_{\mathcal{G}}\left[Y_1+Y_2 \in \mathcal{I}_r\right] \\
&\geq \frac{1}{16\sigma(X')}\sum_{r=\lceil -\ell\sigma(X')\rceil}^{\lfloor \ell\sigma(X')\rfloor} \mathbb{P}_{\mathcal{G}}\left[Y_3 \geq \max\left(1, \lfloor \Delta(p-q)\rfloor\right), Y_1+Y_2 \in \mathcal{I}_r\right]
\end{aligned}$$

We now decouple Y_1 and Y_2, and use Theorem 15 on Y_2, as we did for Y_1. We write $\mathcal{I}_r - i = \{x - i, \forall i \in \mathcal{I}_r\}$.

$$\begin{aligned}
&\geq \frac{1}{16\sigma(X')}\sum_{r=\lceil -\ell\sigma(X')\rceil}^{\lfloor \ell\sigma(X')\rfloor} \sum_{i\in 2\Delta q \pm \ell\sigma(Y_2)} \mathbb{P}_{\mathcal{G}}\left[Y_3 \geq \max\left(1, \lfloor \Delta(p-q)\rfloor\right) \mid Y_2 = i\right]\mathbb{P}_{\mathcal{G}}\left[Y_1 \in \mathcal{I}_r - i\right]\mathbb{P}_{\mathcal{G}}\left[Y_2 = i\right] \\
&\geq \frac{1}{16\sigma(X')}\sum_{r=\lceil -\ell\sigma(X')\rceil}^{\lfloor \ell\sigma(X')\rfloor} \frac{1}{8\sigma(Y_2)}\sum_{i\in 2\Delta q \pm \ell\sigma(Y_2)} \mathbb{P}_{\mathcal{G}}\left[Y_3 \geq \max\left(1, \lfloor \Delta(p-q)\rfloor\right) \mid Y_2 = i\right]\mathbb{P}_{\mathcal{G}}\left[Y_1 \in \mathcal{I}_r - i\right] \\
&\geq \frac{1}{16\sigma(X')}\sum_{r=\lceil -\ell\sigma(X')\rceil}^{\lfloor \ell\sigma(X')\rfloor} \frac{1}{8\sigma(Y_2)}\sum_{i\in 2\Delta q \pm \ell\sigma(Y_2)} \mathbb{P}_{\mathcal{G}}\left[Y_3 \geq \max\left(1, \lfloor \Delta(p-q)\rfloor\right) \mid Y_2 = i\right]\mathbb{P}_{\mathcal{G}}\left[Y_1 \in \mathcal{I}_r - i\right].
\end{aligned}$$

Our goal is now to show that, for all $i \in 2\Delta q \pm \ell\sigma(Y_2)$,

$$\mathbb{P}_{\mathcal{G}}\left[Y_3 \geq \max\left(1, \lfloor \Delta(p-q)\rfloor\right) \mid Y_2 = i\right]\mathbb{P}_{\mathcal{G}}\left[Y_1 \in \mathcal{I}_r - i\right] = \min\left(\Omega(1), \frac{\Delta(p-q)}{\sigma}\right).$$

Fact 1 *When $\Delta(p-q) \leq 1$, then*

$$\mathbb{P}_{\mathcal{G}}\left[Y_3 \geq 1 \mid Y_2 = i\right]\mathbb{P}_{\mathcal{G}}\left[Y_1 \in \mathcal{I}_r - i\right] \geq \frac{\Delta(p-q)}{24\sigma.}$$

Proof For such $i \in 2\Delta q \pm \ell\sigma(Y_2)$, we start by showing that we have:

$$\mathbb{P}_{\mathcal{G}}\left[Y_1 \in \mathcal{I}_r - i\right] \geq \frac{1}{16\sigma}. \tag{3}$$

To see this, write $i = 2\Delta q + r'$, and $\mathcal{I}_r - i = \{\mathbb{E}_{\mathcal{G}}[Y_1] + r - r' - 1\}$. Since $\Delta \leq \frac{s}{3}$, it holds that $\sigma(X') + \sigma(Y_2) \leq 2\sigma(Y_1)$. Thus, for all $r \in \pm\frac{\ell\sigma(X')}{2}$ and $r' \in \pm\frac{\ell\sigma(Y_2)}{2}$, it holds that $|r - r'| \leq \ell\sigma(Y_1)$. Theorem 15 concludes.

Now, we turn to the other term, namely $\mathbb{P}_{\mathcal{G}}\left[Y_3 \geq \Delta(p-q) \mid Y_2 = i\right]$. We first observe that

$$\mathbb{P}_{\mathcal{G}}\left[Y_3 \geq \Delta(p-q) \mid Y_2 = i\right] \geq \mathbb{P}\left[\textsc{Bin}\left(2\Delta - 2\Delta q - \sigma(Y_2), \frac{p-q}{1-q}\right) > 0\right],$$

and

$$\mathbb{E}\left[\textsc{Bin}\left(2\Delta - 2\Delta q - \sigma(Y_2), \frac{p-q}{1-q}\right)\right] = 2\Delta(p-q) - \sqrt{2\Delta q(1-q)}\frac{p-q}{1-q} \in [\Delta(p-q), 2\Delta(p-q)]. \tag{4}$$

We now use the following generic fact about binomials: let $m \in \mathbb{N}$ and $x \in \mathbb{R}_+$ such that $mx \leq 2$. Then:

$$\begin{aligned}\mathbb{P}\left[\textsc{Bin}(m,x) > 0\right] &= 1 - (1-x)^m \geq 1 - \exp(-mx) \\ &\geq 1 - \frac{1}{1+mx} = \frac{mx}{1+mx} \geq \frac{mx}{3}.\end{aligned}$$

Hence, when $\Delta(p-q) \leq 1$, $\mathbb{P}\left[\textsc{Bin}\left(2\Delta - 2\Delta q - \sigma(Y_2), \frac{p-q}{1-q}\right) \geq \Delta(p-q)\right] \geq \frac{\Delta(p-q)}{3}$, since the expectation of that binomial is at most 2 (Equation 4).

Using Equation 3, we thus get

$$\mathbb{P}_{\mathcal{G}}\left[Y_3 \geq \Delta(p-q) \mid Y_2 = i\right]\mathbb{P}_{\mathcal{G}}\left[Y_1 \in \mathcal{I}_r - i\right] \geq \frac{\Delta(p-q)}{24\sigma}.$$

■

Fact 2 *When $\Delta(p-q) > 1$, then*

$$\mathbb{P}_{\mathcal{G}}\left[Y_3 > \Delta(p-q) \mid Y_2 = i\right]\mathbb{P}_{\mathcal{G}}\left[Y_1 \in \mathcal{I}_r - i\right] \geq \frac{1}{32}\min\left(1, \frac{\Delta(p-q)}{\sigma}\right).$$

Proof For such $i \in 2\Delta q \pm \ell\sigma(Y_2)$, we start by showing that we have:

$$\mathbb{P}_{\mathcal{G}}\left[Y_1 \in \mathcal{I}_r - i\right] \geq \frac{1}{16}\min\left(1, \frac{\Delta(p-q)}{\sigma}\right). \tag{5}$$

To see this, write $i = 2\Delta q + r'$,

$$\mathcal{I}_r - i = [\mathbb{E}_{\mathcal{G}}[Y_1] + r - r' - \Delta(p-q), \mathbb{E}_{\mathcal{G}}[Y_1] + r - r'),$$

and as before $|r - r'| \leq \ell\sigma(Y_1)$. Either $\Delta(p-q) \leq \ell\sigma(Y_1)$ and we apply Theorem 15, or $\Delta(p-q) > \ell\sigma(Y_1)$ and $\mathbb{P}_{\mathcal{G}}[Y_1 \in \mathcal{I}_r - i]$ is at least the weight of the interval $\mathbb{E}_{\mathcal{G}}[Y_1] \pm \ell\sigma(Y_1)$, which from Theorem 15 is $1/8$.

Now, we turn to the other term, namely $\mathbb{P}_{\mathcal{G}}[Y_3 \geq \lfloor \Delta(p-q) \rfloor \mid Y_2 = i]$. As in the case $\Delta(p-q) \leq 1$, we observe that

$$\mathbb{P}_{\mathcal{G}}[Y_3 \geq \lfloor \Delta(p-q) \rfloor \mid Y_2 = i] \geq \mathbb{P}\left[\textsc{Bin}\left(2\Delta - 2\Delta q - \sigma(Y_2), \frac{p-q}{1-q}\right) \geq \lfloor \Delta(p-q) \rfloor \right],$$

and

$$\mathbb{E}\left[\textsc{Bin}\left(2\Delta - 2\Delta q - \sigma(Y_2), \frac{p-q}{1-q}\right) \right] \geq \Delta(p-q). \tag{6}$$

Since the median of a binomial X is at least $\lfloor \mathbb{E}[X] \rfloor$, Equation 6 ensures that the median of $\textsc{Bin}\left(2\Delta - 2\Delta q - \sigma(Y_2), \frac{p-q}{1-q}\right)$ is at least $\lfloor \Delta(p-q) \rfloor$, and so $\mathbb{P}_{\mathcal{G}}[Y_3 \geq \lfloor \Delta(p-q) \rfloor \mid Y_2 = i] \geq 1/2$. Combining with Equation 3, we get

$$\mathbb{P}_{\mathcal{G}}[Y_3 > \Delta(p-q) \mid Y_2 = i]\, \mathbb{P}_{\mathcal{G}}[Y_1 \in \mathcal{I}_r - i] \geq \frac{1}{32} \min\left(1, \frac{\Delta(p-q)}{\sigma}\right).$$

■

Therefore, those two facts give that:

$$\begin{aligned}
&\mathbb{P}_{\mathcal{G}}\left[Y_1 + Y_2 + Y_3 > X' \mid Y_1 + Y_2 < X'\right] \\
&\geq \frac{1}{32\sigma(X')} \sum_{r=\lceil -\ell\sigma(X') \rceil}^{\lfloor \ell\sigma(X') \rfloor} \frac{1}{\sigma(Y_2)} \sum_{i \in 2\Delta q \pm \ell\sigma(Y_2)} \frac{1}{32} \min\left(1, \frac{\Delta(p-q)}{\sigma}\right) \\
&\geq c \min\left(1, \frac{\Delta(p-q)}{\sigma}\right),
\end{aligned}$$

with $c = \ell/256$. Putting everything together,

$$\begin{aligned}
&\mathbb{P}_{\mathcal{G}}\left[X \geq X'\right] \geq \\
&\quad \geq \mathbb{P}_{\mathcal{G}}\left[X \geq X' \wedge Y_1 + Y_2 = \max\{Y_1 + Y_2, X'\}\right] + \mathbb{P}_{\mathcal{G}}\left[X \geq X' \wedge Y_1 + Y_2 \neq \max\{Y_1 + Y_2, X'\}\right] \\
&\quad = 1/2 + \mathbb{P}_{\mathcal{G}}\left[Y_1 + Y_2 + Y_3 > X' \mid Y_1 + Y_2 < X'\right] \\
&\quad \geq 1/2 + c \min\left(1, \frac{\Delta(p-q)}{\sigma}\right).
\end{aligned}$$

■

B.2. Proof of Section 3.2

Fact 3 *If $\alpha^{(0)} \geq 0$, then $\beta^{(0)} \geq 0$.*

Proof By design of our algorithm, $|S_1^{(0)}| = |S_2^{(0)}|$. The definitions of $\alpha^{(0)}$ and $\beta^{(0)}$ ensure therefore that:

$$|S_1^{(0)}| = \frac{|C_1^{(0)}| + |C_2^{(0)}|}{2} + \alpha^{(0)}|C_1^{(0)}| - \beta^{(0)}|C_2^{(0)}|$$
$$|S_2^{(0)}| = \frac{|C_1^{(0)}| + |C_2^{(0)}|}{2} - \alpha^{(0)}|C_1^{(0)}| + \beta^{(0)}|C_2^{(0)}|$$

which implies $\beta^{(0)}|C_2^{(0)}| = \alpha^{(0)}|C_1^{(0)}|$, and in particular $\alpha^{(0)}$ and $\beta^{(0)}$ have same sign. ■

Lemma 8 *With high probability (w.r.t. to the randomness of the algorithm), the communities are almost evenly distributed in the subsets: for $i \in \{1,2\}$, it holds that*

$$\mathbb{P}_{\mathcal{A}}\left[\forall \ell \in [1, \sqrt{\log N}],\ |C_i \cap S_\ell| \geq \frac{|C_i|}{2\sqrt{\log N}}\right] \geq 1 - 1/N.$$

Furthermore, there exist a constant κ such that

$$\mathbb{P}_{\mathcal{A}}\left[|\alpha^{(0)}| \geq \frac{1}{8f^{3/2}\sqrt{2n}}\right] \geq 1 - \frac{1}{8f} - \frac{8\kappa\sqrt{f}}{\sqrt{n}}.$$

Proof We start by showing the first statement of the lemma. For that, we actually show by induction the following stronger property: for any ℓ, it holds with probability $1 - \ell/N^2$ that, for $i \in \{1,2\}$,

$$|C_i \setminus \cup_{\ell' < \ell} S_{\ell'}| \in |C_i|(1 - n(\ell-1)/N) \pm x_\ell, \text{ with } x_\ell = \sqrt{N} \cdot (6\log N)^\ell. \tag{7}$$

This would show the first part of the lemma, by noting that $x_\ell \leq \frac{|C_i|}{2\sqrt{\log N}}$.

The base case ($\ell = 1$) trivially holds. Consider $\ell \in [1, \sqrt{\log N} - 2]$, and $i \in \{1,2\}$, and assume that the statement holds for up to ℓ. Let $C_i' = C_i \setminus_{\ell' \leq \ell} C_i \cap S_{\ell'}$. S_ℓ is computed by drawing n points from $C_1' \cup C_2'$. Hence, $|C_i \cap S_{\ell+1}|$ follows an hypergeometric distribution: we draw without replacement $n = N/\sqrt{\log N}$ balls from a set of $|C_1' \cup C_2'|$ balls, and $|C_i'|$ is the number of successful draws. We can therefore use concentration of hypergeometric distribution (Theorem 17):

$$\forall x < \frac{|C_i'| \cdot n}{|C_1' \cup C_2'|},\ \mathbb{P}_{\mathcal{A}}\left[|C_i \cap S_{\ell+1}| < \frac{|C_i'| \cdot n}{|C_1' \cup C_2'|} - x\right] \leq \exp\left(-\frac{x^2}{N}\right).$$

We want to apply this concentration inequality with $x = \frac{|C_i'| \cdot n}{|C_1' \cup C_2'|} - \frac{|C_i|}{\sqrt{\log N}} + x_{\ell+1}$. For that, we first note that applying this inequality is possible, since $x_{\ell+1} \leq \sqrt{N} \cdot (6\log N)^{\sqrt{\log N}} < \frac{|C_i|}{\sqrt{\log N}}$ and so $x \leq \frac{|C_i'| \cdot n}{|C_1' \cup C_2'|}$.

To show the induction, we start by bounding $\frac{|C_i'| \cdot n}{|C_1' \cup C_2'|}$. By induction hypothesis, we know that $|C_i'| \in |C_i|(1 - n(\ell-1)/N) \pm x_\ell$ holds with probability $1 - \ell/N^2$. Hence,

$$\begin{aligned}
\frac{|C_i'| \cdot n}{|C_1' \cup C_2'|} &\geq n \cdot \frac{|C_i|(1 - n(\ell-1)/N) - x_\ell}{N(1 - n(\ell-1)/N) + 2x_\ell} \\
&\geq \frac{|C_i|}{\sqrt{\log N} + \frac{2x_\ell}{n(1-n(\ell-1)/N)}} - \frac{x_\ell}{\sqrt{\log N} - \ell + 1 + \frac{2x_t\sqrt{\log N}}{N}}, \text{ using } n = N/\sqrt{\log N} \\
&\geq \frac{|C_i|}{\sqrt{\log N}} - \frac{2|C_i|x_\ell}{n(1 - n(\ell-1)/N)} - \frac{x_t}{\sqrt{\log N} - \ell + 1 + \frac{2x_\ell\sqrt{\log N}}{N}},
\end{aligned}$$

where the last inequality uses $\frac{1}{1+\theta} \geq 1-\theta$. Now, we use that $\ell \leq \sqrt{\log N} - 2$ and the fact that $\theta \to -\frac{1}{1-\theta}$ is decreasing to get:

$$\begin{aligned}\frac{|C_i'| \cdot n}{|C_1' \cup C_2'|} &\geq \frac{|C_i|}{\sqrt{\log N}} - \frac{2|C_i|x_\ell}{N/\sqrt{\log N}(1-\sqrt{\log N}/\sqrt{\log N})} - \frac{x_\ell}{1+\frac{2x_\ell\sqrt{\log N}}{N}} \\ &\geq \frac{|C_i|}{\sqrt{\log N}} - \frac{4|C_i|x_\ell \log N}{N} - 2x_\ell \geq \frac{|C_i|}{\sqrt{\log N}} - 3x_\ell \log N.\end{aligned}$$

In that case, we have that

$$x = \frac{|C_i'| \cdot n}{|C_1' \cup C_2'|} - \frac{|C_i|}{\sqrt{\log N}} + x_{\ell+1} \geq \frac{|C_i|}{\sqrt{\log N}} - 3x_\ell \log N - \frac{|C_i|}{\sqrt{\log N}} + x_{\ell+1} \geq x_\ell \log N.$$

Since $\frac{|C_i'|\cdot n}{|C_1' \cup C_2'|} - x = \frac{|C_i|}{\sqrt{\log N}} - x_{\ell+1}$, and $x^2 \geq 2N \log N$, we get

$$\mathbb{P}_{\mathcal{A}}\left[|C_i \cap S_{\ell+1}| < \frac{|C_i|}{\sqrt{\log N}} - x_{\ell+1}\right] \leq \exp\left(-2\log N\right).$$

Similarly, we can show that

$$\mathbb{P}_{\mathcal{A}}\left[|C_i \cap S_{\ell+1}| > \frac{|C_i|}{\sqrt{\log N}} + x_{\ell+1}\right] \leq \exp\left(-2\log N\right).$$

Hence given that $|C_i'| \in |C_i|(1-n(\ell-1)/N) \pm x_t$, we have with probability $1-2/N^2$ that $|C_i \cap S_{\ell+1}| \in \frac{|C_i|}{\sqrt{\log N}} \pm x_{\ell+1}$. This shows the statement's lemma for $\ell+1$. We can also conclude the induction principle:

$$\begin{aligned}|C_i \setminus \cup_{\ell'<\ell+1} S_{\ell'}| &= |C_i \setminus \cup_{\ell'<\ell} S_{\ell'}| - |C_i' \cap S_{\ell+1}| \\ &\in |C_i|(1-n\ell/N) \pm x_\ell - \frac{|C_i|}{\sqrt{\log N}} \pm x_{\ell+1} \\ &\in |C_i|(1-n(\ell+1)/N) \pm x_{\ell+1},\end{aligned}$$

hence, the proposition is inductive and holds for any $\ell \leq \sqrt{\log N} - 2$.

Doing a union-bound on all steps $\ell = 1, ..., \sqrt{\log N} - 1$, we get:

$$\mathbb{P}_{\mathcal{A}}\left[\forall \ell,\ |C_i \cap S_\ell| = \frac{|C_i|}{\sqrt{\log N}} \pm x_\ell\right] \geq 1 - 1/N.$$

The statement for all $\ell \leq \sqrt{\log N} - 1$ implies it for $\ell = \sqrt{\log N}$, which concludes the first statement of the lemma.

We now prove the second part. Consider a random variable X with hypergeometric law given by the parameters N_X, K_X, n_X, corresponding to the number of red balls drawn during n_X draws with replacement out of an urn with N_X balls, K_X of them being red. Let $p_X = K_X/N_X$ be the probability of drawing a red ball, and $f_X = n_X/N_X$ be the proportion of balls drawn. The Berry-Esseen theorem for hypegeometric laws (theorem 2.2 in Lahiri and Chatterjee (2007)) says that when $\sigma_X^2 := N_X p_X(1-p_X) f_X(1-f_X) \to \infty$, then

$$\sup_{x \in \mathbb{R}} \left| \mathbb{P}\left[\frac{X - n_X p_X}{\sigma_X} \leq x\right] - G(x)\right| \leq \frac{\kappa}{\sigma_X}, \tag{8}$$

where $G(x)$ is the cumulative density function of $\mathcal{N}(0,1)$.

We use X to denote $|C_1^{(0)} \cap S_{1,1}^{(0)}|$. This variable follows indeed a hypergeometric law, with parameters $n_X = n/2$, $N_X = n$, and $K_X = |C_1^{(0)}| \geq \frac{n}{2f}$. Hence, $p_X \geq \frac{1}{2f}$ and $f_X = 1/2$, so it indeed holds that $\sigma_X \to_{N\to\infty} \infty$. Note furthermore that $n_X p_X = |C_1^{(0)}|/2$, and that $\sigma_X^2 \geq \frac{|C_1^{(0)}|}{4} \cdot (1 - \frac{1}{f}) \geq \frac{|C_1^{(0)}|}{8}$. A simple Chernoff concentration bound gives furthermore that $|C_1^{(0)}| \geq \frac{n}{2f}$: hence, $\sigma_X^2 \geq \frac{n}{16f}$. Similarly, $\sigma_X^2 \leq \frac{|C_1^{(0)}|}{4f}$.

Our goal is to bound the probability that $|\alpha^{(0)}| \geq \frac{1}{8f^{3/2}\sqrt{2n}}$. First, note that $\alpha^{(0)} = \frac{X}{|C_1^{(0)}|} - 1/2$, and so $\frac{X - n_X p_X}{8\sigma_X^2} \leq \alpha^{(0)} \leq \frac{X - n_X p_X}{4f\sigma_X^2}$. We first lower bound $\alpha^{(0)}$: using Equation 8, we get that

$$\begin{aligned}
\mathbb{P}_{\mathcal{A}}\left[\alpha^{(0)} \geq \frac{1}{8f^{3/2}\sqrt{2n}}\right] &\geq \mathbb{P}_{\mathcal{A}}\left[\frac{X - n_X p_X}{\sigma_X} \geq \frac{1}{8f^{3/2}\sqrt{2n}} 8\sigma\right] \\
&\geq \mathbb{P}\left[\mathcal{N}(0,1) \geq \frac{1}{8f^{3/2}\sqrt{2n}}\sqrt{\frac{2n}{f}}\right] - \frac{\kappa}{\sigma} \\
&= \mathbb{P}\left[\mathcal{N}(0,1) \geq \frac{1}{8f^2}\right] - \frac{\kappa}{\sigma}
\end{aligned}$$

We can similarly upper bound $\alpha^{(0)}$:

$$\begin{aligned}
\mathbb{P}_{\mathcal{A}}\left[\alpha^{(0)} \leq -\frac{1}{8f^{3/2}\sqrt{2n}}\right] &\geq \mathbb{P}_{\mathcal{A}}\left[\frac{X - n_X p_X}{\sigma_X} \leq -\frac{1}{8f^{3/2}\sqrt{2n}} 4f\sigma\right] \\
&\geq \mathbb{P}\left[\mathcal{N}(0,1) \leq -\frac{1}{8f^{3/2}\sqrt{2n}}\sqrt{\frac{fn}{2}}\right] - \frac{\kappa}{\sigma} \\
&= \mathbb{P}\left[\mathcal{N}(0,1) \leq -\frac{1}{16f}\right] - \frac{\kappa}{\sigma}
\end{aligned}$$

We now conclude, using that $\mathbb{P}\left[\mathcal{N}(0,1) \in \left[-\frac{1}{16f}, \frac{1}{8f^2}\right]\right] \leq \frac{1}{8f}$:

$$\mathbb{P}_{\mathcal{A}}\left[|\alpha^{(0)}| \geq \frac{1}{16\sqrt{2fn}}\right] \geq 1 - \frac{1}{8f} - 16\frac{\kappa\sqrt{f}}{\sqrt{n}}.$$

■

Lemma 9 *Assume that $\gamma \geq C f^{5/2}\sqrt{\frac{\log N}{N}}$. Let c be the constant from Lemma 5.*

It holds that, after t iterations of the algorithm, $\alpha^{(t)}, \beta^{(t)} > \min(c/2, \log(N)^{t/4}/\sqrt{8n})$ with probability $1 - \frac{1}{8f} - \sum_{i=1}^{t} \exp\left(O\left(-\frac{\log(N)^{t/2}}{f^3}\right)\right)$ on the algorithm's randomness, and probability $1 - \sum_{i=0}^{t} \max\left(\exp\left(-\frac{\log(N)^{(i+1)/2}}{4f}\right), \exp\left(-\frac{c^2 N}{16\sqrt{\log N}}\right)\right)$ on the graph.

Proof First, we assume that $\forall t \in [1, \sqrt{\log N}]$, $|C_i^{(t)}| \geq \frac{|C_i|}{2\sqrt{\log N}}$. This happens with probability $1 - 1/N$ on the algorithm's randomness, due to Lemma 8. We now can show the claim by induction.

The second statement of Lemma 8 ensures that $\alpha^{(0)}, \beta^{(0)} > \frac{1}{8f^{3/2}\sqrt{2n}}$ with probability $1 - \frac{1}{8f} - O(\sqrt{f/n})$ on the algorithm. This initializes our induction.

Suppose now that after step t, $\alpha^{(t)}, \beta^{(t)} > \frac{\log(N)^{t/4}}{8f^{3/2}\sqrt{2n}}$. In order to apply Lemma 5, we need to lower bound $\Delta = \tilde{s}^{(t)}_{2,2} - \tilde{s}^{(t)}_{1,2}$: for that, we deal differently with the case $t = 0$ and $t > 0$.

At the very first step, there is no need to subsample, and so it holds that $\Delta = 2|C_1^{(0)}| \cdot \alpha^{(0)} = 2|C_2^{(0)}| \cdot \beta^{(0)}$. Using Lemma 8, we therefore have $\Delta \geq \frac{n\alpha^{(0)}}{f}$ and $\Delta \geq \frac{n\beta^{(0)}}{f}$. By average (and taking some slack for later), $\Delta \geq \frac{n(\alpha^{(0)}+\beta^{(0)})}{8f}$.

At the other steps, one needs to deal with the effect of subsampling: Lemma 8 gives that $|C_i^{(t)}| > \frac{|C_i|}{2\sqrt{\log N}} \geq \frac{n}{2f}$ and $|S_1^{(t)}| + |S_2^{(t)}| = n$, so

$$\frac{|C_1^{(t)}| \cdot |C_2^{(t)}|}{\max(|S_1^{(t)}|, |S_2^{(t)}|)} \geq \frac{n^2(1 - \frac{1}{2f})}{2f} \cdot \frac{1}{n} = \frac{n}{4f}.$$

Hence, $\frac{|C_1^{(t)}| \cdot |C_2^{(t)}|}{\max(|S_1^{(t)}|, |S_2^{(t)}|)}(\alpha^{(t)} + \beta^{(t)}) \geq \frac{\log(N)^{t/4}\sqrt{n}}{32\sqrt{2}f^{5/2}}$. Therefore, Lemma 7 with $x = \frac{n}{8f}(\alpha^{(t)} + \beta^{(t)}) \geq \frac{\log(N)^{t/4}\sqrt{n}}{64\sqrt{2}f^{5/2}}$ ensures that

$$\begin{aligned}\mathbb{P}_{\mathcal{A}}\left[\Delta \geq \frac{n}{8f}(\alpha^{(t)} + \beta^{(t)})\right] &\geq 1 - \exp\left(-\frac{2\log(N)^{t/2} \cdot n}{64^2 f^5 \cdot \min\left(|S_1^{(t)}|, |S_2^{(t)}|\right)}\right) \\ &\geq 1 - \exp\left(O\left(-\frac{\log(N)^{t/2}}{f^3}\right)\right).\end{aligned}$$

We now condition on the event $\Delta \geq \frac{n}{8f}(\alpha^{(t)} + \beta^{(t)})$, which depends only on the algorithm's randomness. Under that condition, Lemma 5 gives that each vertex of C_1 (resp. C_2) is assigned by the algorithm to the $S_1^{(t+1)}$ (resp. $S_2^{(t+1)}$) with probability $1/2 + c\max\left(1, \frac{\sqrt{n}}{8f} \cdot \frac{(p-q)}{\sqrt{p}}(\alpha^{(t)} + \beta^{(t)})\right)$ on the graph. Using the assumption $\frac{p-q}{\sqrt{p}} \geq Cf^{5/2}\sqrt{\frac{\log N}{N}} = \frac{Cf^{5/2}\log^{1/4} N}{\sqrt{n}}$ this is at least

$$1/2 + \max\left(c, Cf^{3/2}\log(N)^{1/4}(\alpha^{(t)} + \beta^{(t)})\right) \geq 1/2 + \max\left(c, \frac{\log(N)^{(t+1)/4}}{\sqrt{n}}\right).$$

We focus now on vertices of community 1, and bound $\alpha^{(t+1)}$. The proof for $\beta^{(t+1)}$ is exactly alike. When $3\sqrt{f}\log(N)^{1/4}(\alpha^{(t)} + \beta^{(t)}) \leq c$, the previous equation gives that in expectation over the graph randomness, $\mu := \mathbb{E}_{\mathcal{G}}[|C_1 \cap S_1^{(t+1)}|] \geq |C_1^{(t+1)}|(1/2 + \frac{\log(N)^{(t+1)/4}}{\sqrt{n}})$. The variable $|C_1 \cap S_1^{(t+1)}|$ is a sum of $|C_1^{(t+1)}| \geq \frac{n}{2f}$ independent $0/1$ variable, on which we can apply Hoeffding's inequality to bound the deviation to the expectation:

$$\mathbb{P}_{\mathcal{G}}\left[|C_1 \cap S_1^{(t+1)}| < \mu - \frac{|C_1^{(t+1)}| \cdot \log(N)^{(t+1)/4}}{2\sqrt{n}}\right] \leq \exp\left(-\frac{|C_1^{(t+1)}| \log(N)^{(t+1)/2}}{2n}\right)$$
$$\leq \exp\left(-\frac{\log(N)^{(t+1)/2}}{4f}\right).$$

Hence, we conclude: when $t > 0$, $\alpha^{(t)}, \beta^{(t)} > \log(N)^{t/4}/\sqrt{n}$ and $Cf^{3/2}\log(N)^{1/4}(\alpha^{(t)} + \beta^{(t)}) \leq c$, then with probability $1 - \exp\left(O\left(-\frac{\log(N)^{t/2}}{f^3}\right)\right)$ on the algorithm and probability $1 - \exp\left(-\frac{\log(N)^{(t+1)/2}}{4f}\right)$ on the graph, $\alpha^{(t+1)} \geq \frac{\log(N)^{(t+1)/4}}{2\sqrt{n}}$. The same conclusion hold for $t = 0$, but with probability 1 on the algorithm.

The case where $3\sqrt{f}\log(N)^{1/4}(\alpha^{(t)}+\beta^{(t)}) > c$ follows similarly: in that case $\mu = \left|C_1^{(t+1)}\right|(1/2+c)$, and so $\alpha^{(t+1)} > c/2$ with probability

$$1 - \exp\left(-\frac{c^2\left|C_1^{(t+1)}\right|}{8}\right) \geq 1 - \exp\left(-\frac{c^2 N}{16\sqrt{\log N}}\right)$$

on the graph. The induction principle concludes the lemma. ■

Lemma 10 *The algorithm* ExactRecovery *performs* $\binom{b}{2}$ *updates.*

Proof To see this, we view the algorithm as a walk on an undirected complete graph with nodes $\{1, 2, \ldots, b\}$. Whenever Update(S_j, S_i) is executed, then the walk traverses the edges $\{j, i\}$. The algorithm terminates when it is on node h and for all i Update(S_h, S_i) and Update(S_i, S_h) has been called. This occurs when all edges of the current node have been traversed. Therefore, it suffices to argue that any valid path that does not traverse any edge twice, results in an Euler cycle of the graph. To see this, consider the path $v_1, v_2, \ldots, v_\ell$. Where v_1 is the node 1 and v_ℓ is the current node. Suppose ever time an edge is traversed, the edge gets removed. Therefore we have for all $i \in [2, \ell - 1]$ the degree of the nodes is even (using that b is odd). Moreover, if $\ell \neq 1$, then both nodes v_1 and v_ℓ have an odd degree and there exists an Euler path between ℓ and 1. Otherwise, if $\ell = 1$, then node 1 is of even degree and there exists an Euler cycle. Thus in all case the exists an Euler cycle. After continuing the path from p_ℓ to any neighbor $p_{\ell+1}$ the argument can be repeated. ■

B.3. Proof of Section 3.3

Lemma 12 *Let S, S' be two equal-size subsets of the vertices of G. Fix a partition $P_{S'} = (P_1, P_2)$ of S', with $|C_i \cap P_i| \geq |C_i \cap S'|(1/2 + c/2)$ for the constant c given in Lemma 5. The following holds with probability at least $1 - 2\exp(-100\log N)$ on the graph and $1 - \exp\left(-\frac{N\kappa^2}{2f}\right)$ on the algorithm. For any vertex $v \in S \cap C_i$, v has more edges to $\tilde{S}_i$ than to the other part ($\tilde{S}_1$ or $\tilde{S}_2$), where $\tilde{S}_i$ are subsamples of S_i as done in algorithm Phase2.*

Proof Let u be an arbitrary vertex of S' of community C_1 (a symmetric argument applies to vertices of C_2). We aim at bounding the number of edges from u to $\tilde{S}_1$ minus the number of edges from u to $\tilde{S}_2$, where $\tilde{S}_1, \tilde{S}_2$ are obtained by subsampling the partition (S_1, S_2) of S (as defined in algorithm Phase2).

We define two types of edges for the edges outgoing from u. The *type-1* edges $e_1(u)$ are the edges from u to a vertex of $\tilde{S}_1$. The *type-2* $e_2(u)$ edges are the edges from u to a vertex of $\tilde{S}_2$.

Thus, the number of edges u has to $\tilde{S}_1$ minus the number of edges to $\tilde{S}_2$ is given by $e_1(u) - e_2(u)$. Let Σ_u denote this quantity, namely $\Sigma_u = e_1(u) - e_2(u)$. Our goal is to show that Σ_u is positive w.h.p. ensuring that u is assigned correctly.

Using $\tilde{s}_{i,j}$ to denote $|\tilde{S}_{i,j}|$, we have $\mathbb{E}_{\mathcal{G}}[e_1(u)] = \tilde{s}_{1,1}p_u + \tilde{s}_{1,2}q_u$ and $\mathbb{E}_{\mathcal{G}}[e_2(u)] = \tilde{s}_{2,1}p_u + \tilde{s}_{2,2}q_u$, and so:

$$\mathbb{E}_{\mathcal{G}}[\Sigma_u] = \mathbb{E}_{\mathcal{G}}[e_1 - e_2] = (\tilde{s}_{1,1} - \tilde{s}_{2,1})p_u + (\tilde{s}_{1,2} - \tilde{s}_{2,2})q_u = (\tilde{s}_{1,1} - \tilde{s}_{2,1})p_u - (\tilde{s}_{2,2} - \tilde{s}_{1,2})q_u = \Delta(p_u - q_u),$$

with $\Delta = \tilde{s}_{1,1} - \tilde{s}_{2,1} = \tilde{s}_{2,2} - \tilde{s}_{1,2}$ by Lemma 7. We now show that both $e_1(u)$ and $e_2(u)$ are concentrated by applying a standard multiplicative Chernoff bound. Let $\mu = \mathbb{E}_{\mathcal{G}}[e_2(u)]$ and

$$\delta = \frac{\Delta(p_u - q_u)}{4\mu}.$$

Thus, when $\delta < 1$, we have

$$\begin{aligned}
\mathbb{P}_{\mathcal{G}}\left[e_2(u) > (1+\delta)\mathbb{E}\left[e_2(u)\right]\right] &\leq \exp\left(-\frac{1}{2}\delta^2 \mathbb{E}\left[e_2\right]\right) \leq \exp\left(-\frac{\Delta^2 (p_u - q_u)^2}{32\mu}\right) \\
&\leq \exp\left(-\frac{\Delta^2 (p_u - q_u)^2}{32(\tilde{s}_{2,1}p + \tilde{s}_{2,2}q)}\right) \\
&\leq \exp\left(-\frac{\Delta^2 (p_u - q_u)^2}{32(s_{2,1}p + s_{2,2}p)}\right) \leq \exp\left(-\frac{\Delta^2 \gamma^2}{32|S_2|}\right)
\end{aligned}$$

Lemma 7 can be used to bound Δ:

$$\mathbb{P}_{\mathcal{A}}\left[\Delta \geq \frac{|C_1 \cap S| \cdot |C_2 \cap S|}{\max(|P_1|, |P_2|)}c - x\right] \geq 1 - \exp\left(-\frac{2x^2}{\min\left(|P_1|, |P_2|\right)}\right).$$

Since $|C_i \cap S| \geq \frac{N}{2f}$, $\frac{N}{2} \geq |P_i| \geq \frac{N}{4f}$, it holds that $\Delta \geq \frac{Nc\left(1-\frac{1}{2f}\right)}{4f} \geq \frac{N}{8f}$ with probability $1 - \exp\left(-\frac{N\kappa^2}{2f}\right)$ on the random choices of the algorithm. In that case, we conclude using $|S_2| \leq N$ and $\gamma = \min_u \frac{p_u - q_u}{\sqrt{p_u}} \geq C f^{5/2}\sqrt{\frac{\log N}{N}}$:

$$\begin{aligned}
\mathbb{P}_{\mathcal{G}}[e_2 > (1+\delta)\mathbb{E}\left[e_2\right]] &\leq \exp\left(-\frac{\left(\frac{Nc}{8f}\right)^2 \cdot C^2 f^5 \frac{\log N}{N}}{32N}\right) \\
&\leq \exp\left(-\frac{c^2 C^2}{2048} \cdot f^3 \log N\right) \\
&\leq \exp(-100 \log N),
\end{aligned}$$

where the last inequality follows from $C > \frac{500}{c}$ and $f > 1$.

When $\delta \geq 1$, we have that

$$\begin{aligned}\mathbb{P}_{\mathcal{G}}[e_2(u) > (1+\delta)\mathbb{E}\left[\, e_2(u)\,\right]] &\leq \exp\left(-\frac{1}{32}\Delta(p_u - q_u)\right)\\ &\leq \exp\left(-\frac{1}{32}\frac{N\kappa}{4f}p_u\right)\\ &\leq \exp\left(-100\log N\right),\end{aligned}$$

where we have used $p_u \geq \frac{p_u - q_u}{p_u} \geq C^2 f^5 \frac{\log N}{N}$. The concentration bound for $e_1(u)$ is identical.

We thus have with probability probability at least $1 - 2\exp(-100\log N)$ on the graph that $e_2 < \mathbb{E}_{\mathcal{G}}[e_2(u)] + \Delta(p_u - q_u)/2$ and $e_1 > \mathbb{E}_{\mathcal{G}}[e_1(u)] - \Delta(p_u - q_u)/2$, and so by taking a union bound we have that with probability at least $1 - 2\exp(-100\log N)$.

$$\Sigma_u > \Delta(p - q) \geq 1. \tag{9}$$

Therefore, for any arbitrary vertex $u \in S'$ of community C_i, we have that its number of edges to $\tilde{S}_i$ is bigger than to the other part with probability $1 - \exp\left(-\frac{N\kappa^2}{2f}\right)$ on the algorithm and $1 - 2\exp(-100\log N)$ on the graph. Applying a union bound over all the vertices $u \in S'$ shows that the probability that all nodes are correctly assigned is at least $1 - 2\exp(-100\log N)$ on the graph. The proof for vertices of S is exactly alike. ■

Appendix C. Proof of Lower Bound

Lemma 18 *Let $c \in (0, 1/10)$, $p = c\log n/n$, $X \sim \textsc{Bin}(n, p)$ For all $i \in [np - \frac{c^{3/2}\log n}{10}, np + \frac{c^{3/2}\log n}{10}]$ we have*

$$\mathbb{P}\left[\, X = i\,\right] \geq n^{-8c}$$

Proof Our goal is to bound the following expression for all $\varepsilon \in [-\frac{c^{3/2}\log n}{10n}, \frac{c^{3/2}\log n}{10n}]$

$$\binom{n}{np + \varepsilon n} p^{np+\varepsilon n}(1-p)^{n-np-\varepsilon n}. \tag{10}$$

To bound the asymptotic of Equation 10, we use Stirling's approximation:

$$\begin{aligned}&\binom{n}{np + \varepsilon n} p^{np+\varepsilon n}(1-p)^{n-np-\varepsilon n} =\\ &= (1 - o(1))\frac{(n/e)^n\sqrt{2\pi n}\cdot p^{np+\varepsilon n}(1-p)^{n-np-\varepsilon n}}{(n(1-p-\varepsilon)/e)^{n-np-\varepsilon n}\sqrt{2\pi n(1-p-\varepsilon)}\cdot(n(p+\varepsilon)/e)^{np+\varepsilon n}\sqrt{2\pi n(p+\varepsilon)}}\\ &= (1 - o(1))\frac{1}{\sqrt{2\pi n\cdot(1-p-\varepsilon)(p+\varepsilon)}}\cdot\frac{(1-p)^{n-np-\varepsilon n}p^{np+\varepsilon n}}{(1-p-\varepsilon)^{n-np-\varepsilon n}(p+\varepsilon)^{np+\varepsilon n}}\\ &\geq (1 - o(1))\frac{1}{2\sqrt{\pi pn}}\cdot\frac{(1-p)^{n-np-\varepsilon n}p^{np+\varepsilon n}}{(1-p-\varepsilon)^{n-np-\varepsilon n}(p+\varepsilon)^{np+\varepsilon n}}\end{aligned}$$

To simplify this expression, we show some separate bounds. Fix some $\varepsilon \in [-\frac{c^{3/2}\log n}{10n}, \frac{c^{3/2}\log n}{10n}]$, and $k = n(p+\varepsilon)$. We start by showing that

$$(p+\varepsilon)^k \leq e^{c\log(n)/5} p^k. \tag{11}$$

To show this inequality, observe that $(p+\varepsilon)^k \leq p^k(1+\varepsilon/p)^k \leq p^k e^{k\varepsilon/p}$ with $k\varepsilon/p = \frac{np\varepsilon}{p} + \frac{\varepsilon^2 n}{p} \leq \frac{c^{3/2}\log n}{10} + \frac{c^3\log^2 n}{100n^2}\frac{n}{p} \leq 2\frac{c\log n}{10}$.

Moreover, it also hold that

$$1-p-\varepsilon \leq (1-p)(1+2|\varepsilon|). \tag{12}$$

Indeed, since $p \leq 1/2$, $1-p-\varepsilon \leq 1-p-2|\varepsilon| + 2p|\varepsilon| = (1-p)(1+2|\varepsilon|)$.

Our last preliminary inequality is the following. Note that $|\varepsilon|n \leq c\sqrt{c}\log n \leq c\log n$. We have

$$(1+2|\varepsilon|)^{n-k} \leq (1+2|\varepsilon|)^n \leq e^{2n|\varepsilon|} \leq e^{2c\log n}. \tag{13}$$

Thus, using Equation 11, Equation 12 and Equation 13, we can simplify the Stirling approximation:

$$\begin{aligned}
&\binom{n}{np+\varepsilon n} p^{np+\varepsilon n}(1-p)^{n-np-\varepsilon n} \\
&\quad \geq (1-o(1))\frac{1}{2\sqrt{\pi pn}} \cdot \frac{(1-p)^{n-np-\varepsilon n}p^{np+\varepsilon n}}{(1-p-\varepsilon)^{n-np-\varepsilon n}(p+\varepsilon)^{np+\varepsilon n}} \\
&\quad \underset{(a)}{\geq} (1-o(1))\frac{1}{2\sqrt{\pi c\log n}} \cdot \frac{(1-p)^{n-np-\varepsilon n}p^{np+\varepsilon n}}{(1-p)^{n-np-\varepsilon n}\cdot(1+2|\varepsilon|)^{n-np-\varepsilon n}p^{np+\varepsilon n}e^{c\log n/5}} \\
&\quad \underset{(b)}{\geq} (1-o(1))\frac{1}{2\sqrt{\pi c\log n}} \cdot \frac{1}{e^{2c\log n + c\log n/5}} \\
&\quad \geq (1-o(1))\frac{1}{2\sqrt{\pi c\log n}} \cdot \frac{1}{e^{4c\log n}} \geq \frac{1}{n^{8c}},
\end{aligned}$$

where (a) uses Equation 11 and Equation 12, and (b) uses Equation 13.

■

Proof [Proof of Theorem 3] For simplicity assume we have exactly $2n$ nodes, and that nq is an integer. Let also assume that the two communities are drawn uniformly at random: each node has probability $1/2$ to be in each of them (with the restriction that $|C_1| = |C_2| = n$). We say a node is *confusing* if it has exactly

- nq incoming edges from nodes of its community,
- nq incoming edges from nodes of the other community,
- nq outgoing edges to nodes of its community and
- nq outgoing edges to nodes of the other community.

We fix some probability events. Let G be the observed graph. Given a vertex u, let $\mathcal{C}_u$ be the event that u is confusing, and given a set of vertices S, let $\text{Com}(S)$ be the community assignment of vertices from S.

Since all nodes have same p_u, q_u, it holds that:

$$\mathbb{P}\left[\, G \mid u \in C_1, v \in C_2, \mathcal{C}_u, \mathcal{C}_v, \text{Com}(V \setminus \{u,v\}) \,\right] = \mathbb{P}\left[\, G \mid u \in C_2, v \in C_1, \mathcal{C}_u, \mathcal{C}_v, \text{Com}(V \setminus \{u,v\}) \,\right].$$

Hence, using Bayes formula, we get:

$$\mathbb{P}\left[\, u \in C_1, v \in C_2 \mid G, \mathcal{C}_u, \mathcal{C}_v, \text{Com}(V \setminus \{u,v\}) \,\right] = \mathbb{P}\left[\, u \in C_2, v \in C_1 \mid G, \mathcal{C}_u, \mathcal{C}_v, \text{Com}(V \setminus \{u,v\}) \,\right]$$

This probability is $1/2$ when there are $n-1$ other vertices both in C_1 and C_2. This means that knowing that u and v have exactly the same number of neighbors in each community does not inform us in any way on their respective community. Now, we can show using Lemma 18 that there exist confusing nodes with high probability.

The number of outgoing (incoming) edges to nodes of its own community follows $\text{BIN}(n-1,p)$, and, by assumption of the theorem,

$$p - q \le \sqrt{p}\frac{c_p}{20}\sqrt{\frac{\log n}{n}} \le c_p^{3/2}\frac{\log n}{20n}.$$

This implies that $nq \in [np - \frac{c_p\sqrt{np\log n}}{10}, np]$ and we can apply Lemma 18: $\mathbb{P}\left[\, \text{BIN}(n-1,p) = nq \,\right] \ge 1/n^{8c_p} \ge n^{-1/10}$. The number of edges toward (from) the other community follows a binomial $\text{BIN}(n,q)$, with $q = c_q \log n/n$ and $c_q < c_p \le 1/40$. Applying the lemma directly shows $\mathbb{P}\left[\, \text{BIN}(n,q) = nq \,\right] \ge 1/n^{8c_q} \ge n^{-1/10}$. Hence, since all the edges considered are independent, a node is confusing with probability at least $n^{-4/10}$.

Given a pair of vertices u, v that do not share edges and any assignment of the other vertices to communities, with $n-1$ vertices in each, we therefore have that both u and v are confusing with probability $n^{-8/10}$ – note that since u and v do not share edges, the events are independent.

Partition the vertex set into n pairs, and let A_i be the event that the nodes in the i-th pair are both confusing. The variables A_i are 8-read w.r.t. to the definition given in Theorem 14, since A_i only depends on edges adjacent to vertices of the i-th pair, and so A_i and A_j involve only 8 common edges. Hence, we can apply the concentration bound of Theorem 14 to say that there exist a confusing pair with probability $1 - \exp(-\gamma n)$, for some constant γ.

Thus, with high probability there exists two confusing nodes from two different communities. Even knowing the graph G, those two nodes have equal probability of being in each community: therefore, any algorithm that only observes the graph G and assign communities to those vertices must fail on half of the assignments.

So any algorithm fails with probability $1/2$ on the graph randomness.

Note that it is easy to boost this probability to $1 - o(1)$, by identifying a larger number of confusing pairs instead of a single one: with k pairs, the probability of failure becomes $1 - 1/2^k$. ■