Proof of the Achievability Conjectures for the General Stochastic Block Model

EMMANUEL ABBE
Program in Applied and Computational Mathematics
and Department of Electrical Engineering
Princeton University

COLIN SANDON
Princeton University, USA

Abstract

In a paper that initiated the modern study of the stochastic block model (SBM), Decelle, Krzakala, Moore, and Zdeborová, backed by Mossel, Neeman, and Sly, conjectured that detecting clusters in the symmetric SBM in polynomial time is always possible above the Kesten-Stigum (KS) threshold, while it is possible to detect clusters information theoretically (i.e., not necessarily in polynomial time) below the KS threshold when the number of clusters k is at least 4. Massoulié, Mossel et al., and Bordenave, Lelarge, and Massoulié proved that the KS threshold is in fact efficiently achievable for $k = 2$, while Mossel et al. proved that it cannot be crossed at $k = 2$. The above conjecture remained open for $k \geq 3$.

This paper proves the two parts of the conjecture, further extending the results to general SBMs. For the efficient part, an approximate acyclic belief propagation (ABP) algorithm is developed and proved to detect communities for any k down to the KS threshold in quasi-linear time. Achieving this requires showing optimality of ABP in the presence of cycles, a challenge for message-passing algorithms. The paper further connects ABP to a power iteration method on a nonbacktracking operator of generalized order, formalizing the interplay between message passing and spectral methods. For the information-theoretic part, a nonefficient algorithm sampling a typical clustering is shown to break down the KS threshold at $k = 4$. The emerging gap is shown to be large in some cases, making the SBM a good case study for information-computation gaps.
© 2017 Wiley Periodicals, Inc.

Contents

Communications on Pure and Applied Mathematics, Vol. LXXI, 1334–1406 (2018)

1 Introduction

The stochastic block model (SBM) is a statistical model of networks with communities, and a natural model for studying various questions in machine learning, statistics, and complexity theory. In particular, the model serves as a test bed for clustering and community detection algorithms, commonly used in the context of social networks [66], protein-to-protein interactions networks [28], gene expressions [31], recommendation systems [54], medical prognosis [74], DNA folding [23], image segmentation [71], natural language processing [11], and more. The model also captures typical phenomena occurring in the inference of combinatorial structures, such as discrepancies of statistical and computational thresholds.

The SBM emerged independently in multiple scientific communities. The block model terminology, which seems to have dominated in the recent years, comes from the machine learning and statistics literature [17, 30, 40, 49, 52, 68, 73, 76, 77], while the model is typically called the planted partition model in theoretical computer science [19,22,24,33,37,51,56], and the inhomogeneous random graphs model in the mathematics literature [18]. Although the model was defined as far back as the 1980s, it resurged in recent years due in part to the following conjecture established first in [34], and backed in [61], from deep but nonrigorous statistical physics arguments:

CONJECTURE 1. *Let (X, G) be drawn from SBM(n, k, a, b); i.e., X is uniformly drawn among partitions of $[n]$ into k balanced groups (the clusters or communities), and G is a random graph on the vertex set $[n]$ where edges are placed independently with probability a/n inside the clusters and b/n across. Define* $\mathrm{SNR} = \frac{(a-b)^2}{k(a+(k-1)b)}$ *and say that an algorithm detects communities if it takes G as an input and outputs a clustering $\widehat{X}$ that is positively correlated with X with high probability. Then*:

(i) *Irrespective of k, if* SNR > 1*, it is possible to detect communities in polynomial time*; *i.e., the Kesten-Stigum* (*KS*) *threshold can be achieved efficiently.*

(ii) *If $k \geq 4$* (*the conjecture requires $k \geq 5$ when imposing the constraint that $a > b$*)*, it is possible to detect communities information-theoretically* (*i.e., not necessarily in polynomial time*) *for some* SNR *strictly below* 1.

We prove this conjecture here. (The proof was first posted in [5].) The problem was settled already for the case of $k = 2$: It was proved in [55, 60] that the KS threshold can be achieved efficiently for $k = 2$, with an alternative proof later given in [20], and [61] shows that no information-computation gap takes place for $k = 2$ with a tight converse. It was also shown in [20] that for SBMs with multiple communities satisfying certain constraints (i.e., for balanced communities and with the requirement that μ_k in theorem 5 of [20] is a simple eigenvalue), the KS threshold can be achieved efficiently. Yet [20] does not resolve Conjecture 1 for $k \geq 3$.

An interesting challenge raised by part (i) of Conjecture 1 is that standard clustering methods, commonly used in applications, fail to achieve the KS threshold. This includes spectral methods based on the adjacency matrix or Laplacians, as well as SDPs. For standard spectral methods, a first issue is that the fluctuations in the node degrees produce high-degree nodes that disrupt the eigenvectors from concentrating on the clusters. A classical trick is to trim such high-degree nodes [29, 32, 44, 75], throwing away some information, but this does not suffice to achieve the KS threshold. SDPs are a natural alternative, but they also stumble[1] before the KS threshold [44, 59], focusing on the most likely rather than typical clusterings. As we shall show in this paper, and as already investigated in [20, 53] for two communities, a linearized BP algorithm, or equivalently a spectral algorithm on a nonbacktracking matrix, provides a solution to the conjecture.

The classical nonbacktracking matrix B of a graph was introduced by Hashimoto [47] to study the Ihara zeta function, with the identity $\det(I - zB) = \frac{1}{\zeta(z)}$, where ζ is the Ihara zeta function of the graph. In particular, the poles of the Ihara zeta function are the reciprocal of the eigenvalues of B. Studying the spectrum of a graph thus implies properties on the location of the Ihara zeta function. The matrix is further used to define the graph Riemann hypothesis in [50], generalizing notions of Ramanujan graphs to nonregular graphs. The operator that we study is a natural extension of the classical nonbacktracking operator of Hashimoto, where we prohibit not only standard backtracks but also finite cycles. The approach is closely related to [20, 53].

In their original paper [34], Decellel, Krzakala, Moore, and Zdeborová conjecture that belief propagation (BP) achieves the KS threshold, and in fact, gives the the optimal accuracy in the reconstruction of the communities. However, the main issue when applying BP to the SBM is the classical one: the presence of cycles in the graph makes the behavior of the algorithm difficult to understand, and BP is susceptible to settling down in the wrong fixed points.[2] This is a long-standing challenge in the realm of message-passing algorithms for graphical models. Moreover, achieving the KS threshold requires precisely running BP to an extent where the graph is not even treelike, thus precluding simple tricks. No method is currently known to analyze BP with random initialization, as discussed in [60]. We develop here an alternative approach using a linearized version of belief propagation, which is more amenable for the analysis. This also gives a plan to study BP with random initializations on sparse graphs: take a first round with linearized BP at large depth to show convergence to a nontrivial solution, and then improve the obtained configuration with classical BP at short depth (see Section 5 for further discussions on this).

[1] The recent results of [57] on robustness to monotone adversaries suggest that SDPs can in fact not achieve the KS threshold.

[2] Empirical studies of BP on loopy graphs still give positive results in various scenarios, e.g., [64].

The paper also proves part (ii) of Conjecture 1, crossing the KS threshold at $k = 4$ using a nonefficient algorithm that samples a typical clustering (i.e., a clustering having the right proportions of edges inside and across clusters). Note the "information-computation gap" is used here for the gap between the KS and information-theoretic thresholds, which is the gap between the computational and information-theoretic thresholds only under nonformal evidence [34]. However, the IT bound that results from our analysis gives a gap to the KS threshold that can be very significant (see Remark 2.15), making the SBM a good case study for such gap phenomena.

1.1 Our Results Descriptions

The main contributions are the following:

(1) An approximate acyclic belief propagation (ABP) algorithm is developed and shown to detect communities down to the KS threshold with complexity $O(n \log n)$, proving part (i) of Conjecture 1. A more general result applying to arbitrary (possibly asymmetrical) SBMs with a generalized notion of detection and KS threshold is also developed. The complexity of ABP is either comparable or improved compared to prior algorithms for $k = 2$ [20, 55, 60], while ABP achieves universally the KS threshold (see Theorem 2.7).

(2) An algorithm that samples a clustering with typical volumes and cuts is shown to detect communities below the KS threshold at $k = 4$, proving part (ii) of Conjecture 1.

Along the way, the following are also obtained:

(3) A connection between ABP and a power iteration method on a nonbacktracking operator is developed, extending the operator of [47] to higher-order nonbacktracks, and formalizing the interplay described in [53] between linearized BP and nonbacktracking operators.

(4) An information-theoretic (IT) bound is derived for the symmetric SBM. For $a = 0$, it is shown that detection is information-theoretically solvable if $b > ck \ln k + o_k(1)$, $c \in [1, 2]$, showing that the gap between the KS and the IT thresholds can be large, as the KS threshold is $b > k(k-1)$. Our bound interpolates the optimal threshold at $a = 0$ and is conjectured to be tight in the scaling of small b for any k and in the scaling of large k.

(5) An efficient algorithm is shown to learn the parameters a, b and the number of communities k in the symmetric SBM down to the KS threshold.

To achieve the KS threshold, we rely on a linearized version of BP that can handle cycles. The simplest linearized version of BP[3] is to simply repeatedly update beliefs about a vertex's community based on its neighbor's suspected communities while ignoring the part of that belief that results from the beliefs about that

[3] Different forms of approximate message-passing algorithms have been studied for dense graphs, such as the AMP developed in [36] for compressed sensing.

vertex's community to prevent a feedback loop. However, this only works ideally if the graph is a tree. The correct response to a cycle would be to discount information reaching the vertex along either branch of the cycle to compensate for the redundancy of the two branches. However, due to computational issues we simply prevent information from cycling around small cycles in order to limit feedback. We also add steps where a multiple of the beliefs in the previous step are subtracted from the beliefs in the current step to prevent the beliefs from settling into an equilibrium where vertices' communities are sytematically misrepresented in ways that add credibility to each other. We refer to Section 2.1 for a complete description of the algorithm and Section 3 for further intuition on how it performs.

The fact that ABP is equivalent to a power iteration method on a nonbacktracking operator results from its linearized form, as pointed out first in [53]. This provides an intriguing synergy between message passing and spectral algorithms. It further allows us to interpret the obstructions of spectral methods through the lens of BP. The risk of obtaining eigenvectors that concentrate on singular structures (e.g., high-degree nodes for the Laplacian) is related to the risk that BP settles down in wrong fixed points (e.g., due to cycling around high-degree nodes). Rather than removing such obstructions, ABP mitigates the feedback coming from the loops by avoiding backtracks of higher order. In addition to simplifying the proofs, higher-order nonbacktracks are likely to be helpful in real networks, where short loops are more frequent.

To cross the KS threshold information theoretically, we rely on a nonefficient algorithm that samples a typical clustering. Upon observing a graph drawn from the SBM, the algorithm builds the set of all partitions of the n nodes that have a typical fraction of edges inside and across clusters, and then samples a partition uniformly at random from that set. The analysis of the algorithm reveals three different regimes that reflect three layers of refinement in the bounds on the typical set's size. In a first regime, no bad clustering (i.e., partition of the nodes that classifies close to $1/k$ of the vertices correctly) is typical with high probability based on a union bound, and the algorithm samples only good clusterings with high probability. This allows us to cross the KS threshold for $k = 5$ when $a = 0$ but does not give the right bound at $b = 0$; see [8]. In a second regime, the large number of treelike components in the graph is exploited, finding some bad clusterings to be typical but unlikely to be sampled. This gives a regime where the algorithm succeeds with the right bound at $b = 0$, but not the right approximation at small b. To address the latter, a finer estimate on the typical set's size is obtained by also exploiting parts of the giant that are treelike. Finally, we tighten our estimates on the typical set's size by taking into account vertices that are not saturated, i.e., whose neighbors do not cover all communities. The final bound crosses the KS threshold at $k = 4$, interpolates the optimal threshold at $a = 0$, and is conjectured to be tight in the scaling for small b, small a, and large k. Further details are in Section 4.

The learning of the parameters a, b, k is done similarly as for the case $k = 2$ [61]. Note that learning the parameters when k is unknown was previously settled only for diverging degrees [6] with related results in [21].

1.2 Related Literature

Several methods were proved to succeed down to the KS threshold for two communities [20, 55, 60], all influential in the development of ABP. The first is based on a spectral method from the matrix of self-avoiding walks (entry (i, j) counts the number of self-avoiding walks of moderate size between vertices i and j) [55], the second on counting weighted nonbacktracking walks between vertices [60], and the third on a spectral method with the matrix of nonbacktracking walks between directed edges [20]. [4]

The first method has a complexity of $O(n^{1+\varepsilon})$, $\varepsilon > 0$, while the second method affords a lesser complexity of $O(n \log^2 n)$ but with a large constant (see discussion in [60]). These two methods were the first to achieve the KS threshold for two communities. The third method is based on an elegant analysis of the spectrum of the nonbacktracking operator and allows going beyond the SBM with two communities, requiring however certain assumptions on the SBM parameters to obtain a result for detection (the precise condition is the requirement on μ_k being a simple eigenvalue of M in [20, theorem 5], and the balanced requirement on the community sizes), thus falling short of proving Conjecture 1(i) for $k \geq 3$ (since the second eigenvalue in this case has multiplicity at least 2). Note that a certain amount of symmetry is needed to make the detection problem interesting. For example, if the communities have different average degrees, detection becomes trivial. Thus the symmetric model SBM(n, k, a, b) is in a sense the most challenging model for detection.

The nonbacktracking operator was proposed first for the SBM in [53], also described as a linearization of BP. Note that the nonbacktracking operator suffers from an increase in dimension, as the derived matrix scales with the number of edges rather than vertices (specifically $2|E| \times 2|E|$, where $|E|$ is the number of edges).[5] Nonbacktracking spectral methods were also developed recently for the problem of detecting a single planted community [46].

Our results are closest to [20, 60], while diverging in several key parts. A few technical expansions in the paper are similar to those carried in [60], such as the weighted sums over nonbacktracking walks and the SAW decomposition from [60], which are similar to our compensated nonbacktracking walk counts and standard decomposition. Our definitions are developed to cope with general SBMs rather than the 2-symmetric case, in particular to compensate for the dominant eigenvalues in the latter setting, which is delicate due to the numerous potentially

[4] Related ideas relying on shortest paths were also considered in [16].

[5] The nonbacktracking matrix is also not normal and has thus a complex spectrum; an interesting heuristic based on the Bethe Hessian operator was proposed in [70] to address the dimensionality and normality issues.

close eigenvalues. Our algorithm complexity is also slightly reduced by a logarithmic factor.

Our algorithm is also closely related to [20], which focuses on extracting the eigenvectors of the standard nonbacktracking operator. However, our proof technique is different than the one in [20], so that we can cope with the setting of Conjecture 1 (i.e., multiplicity of eigenvalues). Also, we do not proceed with the extraction of eigenvectors, but implement the algorithm in a message-passing fashion. This avoids building the nonbacktracking matrix whose dimension grows with the number of edges. Note that from a spectral point of view, the power iteration method that we use is not relying on a traditional deflation method that subtracts the dominant eigenvector. Such an approach is likely to work in the symmetric SBM, but in the general SBM, we rely on a different approach that subtracts large eigenvalues times the identity matrix. Another difference from [20] is that we rely on nonbacktracking operators of higher orders r. While $r = 2$ is arguably the simplest implementation and may suffice for the sole purpose of achieving the KS threshold, a larger r may be beneficial in practice. For example, an adversary may add triangles for which ABP with $r = 2$ would fail while larger r would succeed. Finally, the approach of ABP can be extended beyond the linearized setting to improve the algorithm's accuracy.

For the information-theoretic part, a few papers have studied bounds and information-computation tradeoffs for SBMs with a growing number of communities [27], two unbalanced communities [65], and a single community [58]. No results seem known for the symmetric SBM and Conjecture 1(b). Shortly after this paper was posted, [13] obtained bounds on the information-theoretic threshold in an independent effort using moments' methods. The upper bound in [13] crosses at $k = 5$ rather than $k = 4$ and does not interpolate to the giant component at $b = 0$. A lower bound matching the scaling in k is also obtained in [13].

1.3 Related Models

Exact recovery is a stronger recovery requirement than detection, which has long been studied for the SBM [1,10,17,19,22,24,26,27,30,33,37,51,56,68,73,75], and more recently in the lens of sharp thresholds [2,12,42,45,62,78]. The notion of exact recovery requires a reconstruction of the complete communities with high probability. It was proved in [2,62] that exact recovery has a sharp threshold for $\mathrm{SBM}(n, 2, a\log(n), b\log(n))$ at $|\sqrt{a}-\sqrt{b}| = 1$, which can be achieved efficiently. In [4], it was proved that for the general SBM with linear size communities, exact recovery has a sharp threshold at the CH divergence, and the threshold is proved to be efficiently achievable for any k (extended in [6] for unknown parameters). This extends the results of [75] that are optimal in the scaling without discussing the phase transition. When considering sublinear communities and the coarser regime of the parameters, [27] gives evidence that exact recovery can again have information-computation gaps. We also conjecture that similar phenomena can take place in the setting of [4] for exact recovery when k is larger than $\log(n)$.

Finally, many variants of the SBM can be studied, such as the labeled-block model [43, 48], the censored-block model [1, 3, 29, 69], the degree-corrected block model [52], overlapping block models [41], and more. While most of the fundamental challenges seem to be captured by the SBM already, these represent important extensions for applications.

2 Results

DEFINITION 2.1. For positive integers k, n, a probability distribution $p \in (0,1)^k$, and a $k \times k$ symmetric matrix $Q \in \mathbb{R}_+^{k\times k}$, define $\mathrm{SBM}(n, p, Q/n)$ as the *probability distribution over ordered pairs* (σ, G) *of an assignment of vertices to one of k communities and an n-vertex undirected graph* as follows. First, each vertex $v \in V(G)$ is independently assigned a community σ_v under the probability distribution p. Then, for every $v \neq v'$, an edge is drawn in G between v and v' with probability $\frac{Q_{\sigma_v, \sigma_{v'}}}{n}$, independently of other edges. We define $\Omega_i = \{v : \sigma_v = i\}$, $i \in [k]$.

We sometimes say that G is drawn under $\mathrm{SBM}(n, p, Q/n)$ without specifying σ. The SBM is called symmetric if p is uniform and if Q takes the same value on the diagonal and the same value outside the diagonal.

DEFINITION 2.2. (σ, G) is *drawn under* $\mathrm{SBM}(n, k, a, b)$ if $p_i = 1/k$, $Q_{i,i} = a$, and $Q_{i,j} = b$ for every $i, j \in [k]$, $i \neq j$.

Our goal is to find an algorithm that can distinguish between vertices from one community and vertices from another community in a nontrivial way:

DEFINITION 2.3. Let $\widehat{\sigma}$ be *an algorithm that takes a graph as input and outputs a partition of its vertices into two sets*. $\widehat{\sigma}$ solves detection or weak recovery in graphs drawn from $\mathrm{SBM}(n, p, Q/n)$ if there exists $\epsilon > 0$ such that the following holds. When (σ, G) is drawn from $\mathrm{SBM}(n, p, Q/n)$ and $\widehat{\sigma}$ divides its vertices into S and S^{c}, with probability $1 - o(1)$, there exist $i, j \in [k]$ such that $|\Omega_i \cap S|/|\Omega_i| - |\Omega_j \cap S|/|\Omega_j| > \epsilon$. Detection is solvable efficiently if the algorithm runs in polynomial time in n, and information-theoretically if no such complexity bound is obtained.

In other words, an algorithm solves detection if it divides the graph's vertices into two sets such that vertices from different communities have different probabilities of being assigned to one of the sets. An alternate definition, used in particular by Decelle et al. [34], says that an algorithm succeeds at detection if it divides the vertices into k sets and there exists $\epsilon > 0$ such that with high probability there exists an identification of the sets with the communities such that the algorithm classifies at least $\max p_i + \epsilon$ of the vertices correctly:

DEFINITION 2.4. An algorithm $\widehat{\sigma} : 2^{\binom{[n]}{2}} \to [k]^n$ is said to *solve max-detection in* $\mathrm{SBM}(n, p, Q)$ if for some $\varepsilon > 0$, $\mathbb{P}\{A(\sigma, \widehat{\sigma}) \geq \max_{i \in [k]} p_i + \varepsilon\} = 1 - o(1)$, where

$A(x,y) = \max_{\pi \in S_k} \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}(x_i = \pi(y_i))$ denotes the agreement between x and y.[6]

In the k community symmetric case, these definitions (detection and max-detection) are equivalent, but in general, this may not hold. The detection definition is satisfied by any algorithm that produces nontrivial amounts of evidence on what communities the vertices are in, while max-detection requires the algorithm to sometimes produce enough evidence to overcome the prior probability. This may not always be possible. In particular, the conjecture from [34] that max-detection is efficiently solvable above the KS threshold is formally incorrect (as shown by the following example), but the conjecture holds for the detection notion introduced in Definition 2.3 as shown in this paper.

To obtain a counterexample of the general conjecture from [34], let $k = 5$, $p = [\frac{1}{2}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}]$, Q be a 5×5 matrix such that $Q_{i,j}$ is ϵ if exactly one of i and j is equal to 1 and 2.2 otherwise for some small $\epsilon > 0$. Note that $\mathrm{SBM}(n, p, Q/n)$ is essentially a 2-community symmetric SBM in which one of the communities has been divided into four subcommunities. We have that $\lambda_1 = (2.2+\epsilon)/2$ and $\lambda_2 = (2.2-\epsilon)/2$, so for sufficiently small ϵ, we have that $\lambda_2^2 > \lambda_1$. That means that we can solve detection on the 2-community SSBM that this SBM emulates, and thus that we can solve detection according to Definition 2.3 on this SBM. However, no algorithm solves max-detection on this SBM, as shown by the following argument.

First of all, for small ϵ, more than 80% of the vertices in a graph drawn from this SBM will be in small components (i.e., not in the giant),[7] which means that it will be impossible to determine what communities they are in. Any classification of the vertices with accuracy greater than $\frac{1}{2}$ must classify more than $\frac{3}{8}$ of these vertices correctly. The only way to achieve this without knowing what communities these vertices are in is to classify the majority of them as being in community 1. However, that fixes the community that the majority of these vertices are assigned to as the one that will be identified with community 1.

One can divide the vertices in the main component of the graph into two sets such that the majority of the vertices from community 1 are in one set and the majority of the vertices from the other communities are in the other set. However, by symmetry there is no way to tell which set is which. Furthermore, the requirement that the set containing the majority of the vertices from other components be identified with community 1 prevents one from being able to compensate for this using the variability of π in the agreement definition. So there is no way to classify these vertices with more than $\frac{1}{2}$ expected accuracy without compromising the classification of the other vertices. Thus, max-detection is impossible on this SBM.

This example is degenerate in the sense that Q has columns that are equal. However, if we change Q so that two vertices in the same small community have an

[6] Permutations π are needed as only the partition is of interest and not the actual labels.

[7] This takes place with high probability.

edge between them with probability $(2.2 + 3\epsilon')/n$ and two vertices in different small communities have an edge between them with probability $(2.2 - \epsilon')/n$ for sufficiently small ϵ', there is still no algorithm that can determine which of the two apparent "communities" is community 1 with high probability, so the result still holds. On the other hand, if all communities have the same size, then there is no minimum amount of evidence required to overcome prior beliefs, and we have the following:

LEMMA 2.5. *Let $k > 0$, Q be a $k \times k$ symmetric matrix with nonnegative entries, p be the uniform distribution over k sets, and A be an algorithm that solves detection in graphs drawn from* $\mathrm{SBM}(n, p, Q/n)$*. Then A also solves max-detection, provided that we consider it as returning $k - 2$ empty sets in addition to its actual output.*

PROOF. Let (σ, G) be drawn from $\mathrm{SBM}(n, p, Q/n)$ and $A(G)$ return S and S'. There exists $\epsilon > 0$ such that with high probability there exist i and j such that $|\Omega_i \cap S|/|\Omega_i| - |\Omega_j \cap S|/|\Omega_j| > \epsilon$. So, if we map S to community i and S' to community j, the algorithm classifies at least

$$\begin{aligned}|\Omega_i \cap S|/n + |\Omega_j \cap S'|/n &= |\Omega_j|/n + |\Omega_i \cap S|/n - |\Omega_j \cap S|/n \\ &\geq 1/k + \epsilon/k - o(1)\end{aligned}$$

of the vertices correctly with high probability. □

Finally, note that our notion of weak recovery can also be explained via an agreement metric, which shows the distinction with Decelle et al.'s definition: weak recovery is solvable if an algorithm produces a community reconstruction $\widehat{X}$ such that its *normalized agreement* to the true communities X satisfies $\widetilde{A}(X, \widehat{X}) = 1/k + \Omega_n(1)$ with high probability, where

$$\widetilde{A}(x, y) = \max_{\pi \in S_k} \frac{1}{k} \sum_{i=1}^{k} \frac{\sum_{u \in [n]} \mathbb{1}(x_u = \pi(y_u), x_u = i)}{\sum_{u \in [n]} \mathbb{1}(x_u = i)}, \quad x, y \in [k]^n. \tag{2.1}$$

2.1 Achieving the KS Threshold Efficiently

We present first a result that applies to the general SBM. We next specify the result for symmetric SBMs and provide the ABP algorithm in the next section.

Given parameters p and Q for the SBM, let P be the diagonal matrix such that $P_{i,i} = p_i \ \forall i \in [k]$. Also, let $\lambda_1, \ldots, \lambda_h$ be the distinct eigenvalues of PQ in order of nonincreasing magnitude. Our results are in terms of the following notion of SNR:

DEFINITION 2.6. The signal-to-noise ratio of $\mathrm{SBM}(n, p, Q/n)$ is defined by

$$\mathrm{SNR} := \lambda_2^2/\lambda_1.$$

This paper shows that efficient detection is possible if SNR > 1. In the k community symmetric case SBM(n, k, a, b) where vertices are connected with probability a/n inside communities and b/n across, we have

$$\mathrm{SNR} = \frac{\left(\frac{a-b}{k}\right)^2}{\frac{a+(k-1)b}{k}} = \frac{(a-b)^2}{k(a+(k-1)b)},$$

which is the quantity in Conjecture 1.

THEOREM 2.7. *Let $p \in (0,1)^k$ with $\sum p = 1$, Q be a symmetric matrix with nonnegative entries, P be the diagonal matrix such that $P_{i,i} = p_i$, and $\lambda_1, \dots, \lambda_h$ be the distinct eigenvalues of PQ in order of nonincreasing magnitude. If $\lambda_2^2 > \lambda_1$, then there exist constants r, c, and $m = \Theta(\log(n))$ such that the ABP algorithm with these parameters solves detection in* SBM$(n, p, Q/n)$. *The algorithm can be run in $O(n \log n)$ time.*

The proof is in Section 6.1.

COROLLARY 2.8. *ABP solves detection in* SBM(n, k, a, b) *if*

$$\frac{(a-b)^2}{k(a+(k-1)b)} > 1 \tag{2.2}$$

and can be run in $O(n \log n)$ time.

Remark 2.9. Our definition of detection also extends to deciding whether an algorithm distinguishes between two specific communities in the sense that there exists $\epsilon > 0$ such that the fraction of vertices from one of these communities assigned to S differs from the fraction of vertices from the other community assigned to S by at least ϵ with high probability. The right version of ABP can then distinguish between communities i and j if there exists an eigenvector w of PQ with an eigenvalue of magnitude greater than $\sqrt{\lambda_1}$ such that $w_i \neq w_j$.

Acyclic Belief Propagation (ABP) Algorithm

We present here two versions of our main algorithm: a simplified version ABP* that applies to the symmetric SBM and that can easily be implemented (see [7]), and the general version ABP that is used to prove Theorem 2.7. The general version has additional steps that are used to prove the theorem, but these can be removed for practical applications. The intuitions behind the algorithms are discussed in Section 3.3, and in Section 3.4 we show how the algorithms can be viewed as applying a power iteration method to a nonbacktracking operator $W^{(r)}$ of generalized order (where r denotes the order of the nonbacktracks). The algorithms below have a message passing implementation and correspond to a linearized version of belief propagation that mitigates cycles.

ABP$^*(G, m, r)$:

(1) For each adjacent v and v' in G, randomly draw $y^{(1)}_{v,v'}$ from a Gaussian distribution with mean 0 and variance 1. Assign $y^{(t)}_{v,v'}$ to a value of 0 for $t < 1$.

(2) For each $1 < t \leq m$, set

$$z^{(t-1)}_{v,v'} = y^{(t-1)}_{v,v'} - \frac{1}{2|E(G)|} \sum_{(v'',v''')\in E(G)} y^{(t-1)}_{v'',v'''}$$

for all adjacent v and v'. For each adjacent v, v' in G that are not part of a cycle of length r or less, set

$$y^{(t)}_{v,v'} = \sum_{\substack{v'':(v',v'')\in E(G),\\ v''\neq v}} z^{(t-1)}_{v',v''},$$

and for the other adjacent v, v' in G, let the other vertex in the cycle that is adjacent to v be v''', the length of the cycle be r', and set

$$y^{(t)}_{v,v'} = \sum_{\substack{v'':(v',v'')\in E(G),\\ v''\neq v}} z^{(t-1)}_{v',v''} - \sum_{\substack{v'':(v,v'')\in E(G),\\ v''\neq v', v''\neq v'''}} z^{(t-r')}_{v,v''}$$

unless $t = r'$, in which case, set

$$y^{(t)}_{v,v'} = \sum_{\substack{v'':(v',v'')\in E(G),\\ v''\neq v}} z^{(t-1)}_{v',v''} - z^{(1)}_{v''',v}.$$

(3) Set $y'_v = \sum_{v':(v',v)\in E(G)} y^{(m)}_{v,v'}$ for all $v \in G$. Return $(\{v : y'_v > 0\}, \{v : y'_v \leq 0\})$.

Remarks.

(1) In the $r = 2$ case, one does not need to find cycles, and one can exit step (2) after the second line. As mentioned above, we rely on a less compact version of the algorithm to prove the theorem, but expect that the above also succeeds at detection as long as $m > 2\ln(n)/\ln(\mathrm{SNR}) + \omega(1)$.

(2) What the algorithm does if (v, v') is in multiple cycles of length r or less is unspecified above, as there is no such edge with probability $1 - o(1)$ in the sparse SBM. This can be modified for more general settings. The simplest such

modification is to apply this adjustment independently for each such cycle, setting

$$y^{(t)}_{v,v'} = \sum_{\substack{v'':(v',v'')\in E(G),\\ v''\neq v}} z^{(t-1)}_{v',v''} - \sum_{r'=1}^{r} \sum_{v''':(v,v''')\in E(G)} C^{(r')}_{v''',v,v'} \sum_{\substack{v'':(v,v'')\in E(G),\\ v''\neq v, v''\neq v''''}} z^{(t-r')}_{v,v''},$$

where $C^{(r')}_{v''',v,v'}$ denotes the number of length r' cycles that contain v''', v, v' as consecutive vertices, substituting $z^{(1)}_{v''',v}$ for

$$\sum_{\substack{v'':(v,v'')\in E(G),\\ v''\neq v, v''\neq v''''}} z^{(t-r')}_{v,v''}$$

when $r' = t$. This will not exactly count r-nonbacktracking walks, but we believe that it gives a good enough approximation.

(3) The purpose of setting

$$z^{(t-1)}_{v,v'} = y^{(t-1)}_{v,v'} - \frac{1}{2|E(G)|} \sum_{(v'',v''')\in E(G)} y^{(t-1)}_{v'',v'''}$$

is to ensure that the average value of the $y^{(t)}$ is approximately 0, and thus that the eventual division of the vertices into two sets is roughly even. There is an alternate way of doing this in which we simply let $z^{(t-1)}_{v,v'} = y^{(t-1)}_{v,v'}$ and then compensate for any bias of $y^{(t)}$ towards positive or negative values at the end. More specifically, we define Y to be the $n \times m$ matrix such that for all t and v $Y_{v,t} = \sum_{v':(v',v)\in E(G)} y^{(t)}_{v,v'}$, and M to be the $m \times m$ matrix such that $M_{i,i} = 1$ and $M_{i,i+1} = -\lambda_1$ for all i, and all other entries of M are equal to 0. Then we set $y' = YM^{m'}e_m$, where $e_m \in \mathbb{R}^m$ denotes the unit vector with 1 in the m^{th} entry, and m' is a suitable integer.

The full version of the algorithm applying to the general SBM is as follows.

ABP$(G, m, r, c, (\lambda_1, \ldots, \lambda_h))$:

(1) Initialize:

(a) Set $s = 2$ unless $h > 2$ and $|\lambda_2| = |\lambda_3|$, in which case set $s = 3$.

(b) Set $\gamma = (1 - \lambda_1/\lambda_2^2)/2$.

(c) Set $l = \max((s-1)/\ln((1-\gamma)|\lambda_s|) + s - 1, 2(2r+1)(s-1))$.

(d) Assign each edge of G independently with probability γ to a set Γ. Then remove these edges from G.

(e) For every vertex $v \in G$, randomly draw x_v from a Gaussian distribution with mean 0 and variance 1.

(f) For each adjacent v and v', set $y^{(1)}_{v,v'} = x_{v'}$ and $y^{(t)}_{v,v'} = 0$ for all $t < 1$.

(2) Propagate:

(a) For each $1 \le t \le m$ and each adjacent $(v, v') \in E(G)$, set

$$y^{(t)}_{v,v'} = \sum_{\substack{v'':(v',v'')\in E(G),\\ v''\neq v}} y^{(t-1)}_{v',v''}$$

unless (v, v') is part of a cycle of length r or less. If it is, then let the other vertex in the cycle that is adjacent to v be v''', and the length of the cycle be r'.[8] Set

$$y^{(t)}_{v,v'} = \sum_{\substack{v'':(v',v'')\in E(G),\\ v''\neq v}} y^{(t-1)}_{v',v''} - \sum_{\substack{v'':(v,v'')\in E(G),\\ v''\neq v, v''\neq v'''}} y^{(t-r')}_{v,v''}$$

unless $t = r'$. In that case, set

$$y^{(t)}_{v,v'} = \sum_{\substack{v'':(v',v'')\in E(G),\\ v''\neq v}} y^{(t-1)}_{v',v''} - x_v.$$

(b) Set Y to be the $n \times m$ matrix such that for all t and v,

$$Y_{v,t} = \sum_{v':(v',v)\in E(G)} y^{(t)}_{v,v'}$$

(c) For each $s' < s$, set $M_{s'}$ to be the $m \times m$ matrix such that $M_{i,i} = 1$ and $M_{i,i+1} = -(1-\gamma)\lambda_{s'}$ for all i, and all other entries of $M_{s'}$ are equal to 0. Also, let $e_m \in \mathbb{R}^m$ be the vector with an m^{th} entry of 1 and all other entries equal to 0. Set

$$y^{(m)} = Y\Big(\prod_{s'<s} M_{s'}^{\lceil \frac{m-r-(2r+1)s'}{l} \rceil}\Big) e_m.$$

(d) For each v, set

$$y'_v = \sum_{v':(v,v')\in \Gamma} y^{(m)}_{v'}$$

and set y''_v to the sum of $y'_{v'}$ over all v' that have shortest paths to v of length $\lfloor \sqrt{\log\log n} \rfloor$.

(3) Assign:

(a) Set $c' = c \cdot \sqrt{\sum_{v\in G} (y''_v)^2/n}$. Create sets of vertices S_1 and S_2 as follows. For each vertex v, if $y''_v < -c'$, assign v to S_1. If $y''_v > c'$, then assign v to S_2. Otherwise, assign v to S_2 with probability $\frac{1}{2} + y''_v/2c'$ and S_1 otherwise.

(b) Return (S_1, S_2).

[8] What the algorithm does if (v, v') is in multiple cycles of length r or less is unspecified as there is no such edge with probability $1 - o(1)$ in the SBM. One can adapt this for more general models.

2.2 Crossing the KS Threshold Information-Theoretically

The following gives a region of the parameters in the symmetric SBM where detection can be solved information-theoretically.

THEOREM 2.10. *Let* $d := \frac{a+(k-1)b}{k}$, *assume* $d > 1$, *and let* $\tau = \tau_d$ *be the unique solution in* $(0,1)$ *of* $\tau e^{-\tau} = de^{-d}$, *i.e.,* $\tau = \sum_{j=1}^{+\infty} \frac{j^{j-1}}{j!}(de^{-d})^j$. *The typicality sampling algorithm* (*see below*) *detects communities in* $\mathrm{SBM}(n,k,a,b)$[9] *if*

$$\frac{a\ln a + (k-1)b\ln b}{k} - \frac{a+(k-1)b}{k}\ln\frac{a+(k-1)b}{k} > \min\biggl(\frac{1-\tau}{1-\tau k/(a+(k-1)b)}2\ln(k), \; 2\ln(k) - 2\ln(2)e^{-a/k}(1-(1-e^{-b/k})^{k-1})\biggr). \tag{2.3}$$

This bound strictly improves on the KS threshold for $k \geq 4$. In particular, for $k = 4$, $a = 0$, $b = 12$, i.e., at the KS threshold, the left-hand side of the inequality above is greater than 2.589, while the right-hand side is less than 2.576. Since both sides of the inequality are continuous in a and b, a slight modification of these parameters yields values below the KS threshold for which TSA solves detection.

COROLLARY 2.11. *Conjecture* 1(ii) *holds.*

Typicality Sampling Algorithm (TSA)

Define

$$\mathrm{Bal}(n,k,\varepsilon) = \left\{x \in [k]^n : \forall i \in [k], \frac{|\{u \in [n] : x_u = i\}|}{n} \in [1/k - \varepsilon, 1/k + \varepsilon]\right\}$$

and $\mathrm{Bal}(n,k) = \mathrm{Bal}(n,k,\log n/\sqrt{n})$, the set of vectors in $[k]^n$ with an asymptotically uniform fraction of components in each community.

Given an n-vertex graph G and $\delta > 0$, the algorithm draws $\widehat{\sigma}_{\mathrm{typ}}(G)$ uniformly at random in

$$\begin{aligned} T_\delta(G) = \Biggl\{ x \in \mathrm{Bal}(n,k,\delta) : & \\ & \sum_{i=1}^{k} \left| \{G_{u,v} : (u,v) \in \binom{[n]}{2} \text{ s.t. } x_u = i, x_v = i\} \right| \geq \frac{an}{2k}(1-\delta), \\ & \sum_{\substack{i,j \in [k],\\ i<j}} \left| \left\{ G_{u,v} : (u,v) \in \binom{[n]}{2} \text{ s.t. } x_u = i, x_v = j \right\} \right| \\ & \qquad \leq \frac{bn(k-1)}{2k}(1+\delta) \Biggr\}, \end{aligned}$$

[9] Setting $\delta > 0$ small enough gives the existence of $\varepsilon > 0$ for detection.

where the above assumes that $a > b$; flip the two inequalities in the case $a < b$.

Remark 2.12. Note that (2.3) simplifies to

$$\frac{1}{2\ln k}\left(\frac{a\ln a + (k-1)b\ln b}{k} - d\ln d\right) > \frac{1-\tau}{1-\tau/d} =: f(\tau, d), \tag{2.4}$$

and since $f(\tau, d) < 1$ when $d > 1$ (which is needed for the presence of the giant), detection is already solvable in $\mathrm{SBM}(n,k,a,b)$ if

$$\frac{1}{2\ln k}\left(\frac{a\ln a + (k-1)b\ln b}{k} - d\ln d\right) > 1. \tag{2.5}$$

As we shall see in Lemma 6.26, the above corresponds to the regime where there is no bad clustering that is typical with high probability. However, the above bound is not tight in the extreme regime of $b = 0$, since it reads $a > 2k$ as opposed to $a > k$, and only crosses the KS threshold at $k = 5$. However, in the asymmetric case, the counterpart of the above bound already allows us to cross the KS threshold for $k = 2$; for example, we obtain that detection is possible below the KS threshold for the asymmetric SBM given by $p = (\frac{1}{10}, \frac{9}{10})$ and $Q = (0, 81; 81, 72)$.

Defining $a_k(b)$ as the unique solution of

$$\frac{1}{2\ln k}\left(\frac{a\ln a + (k-1)b\ln b}{k} - d\ln d\right) = \min\left(f(\tau, d), 1 - \frac{e^{-a/k}(1-(1-e^{-b/k})^{k-1})\ln(2)}{\ln(k)}\right)$$

and simplifying the bound in Theorem 2.10 gives the following:

COROLLARY 2.13. *Detection is solvable*

$$\textit{in } \mathrm{SBM}(n,k,0,b) \quad \textit{if} \quad b > \frac{2k\ln k}{(k-1)\ln\frac{k}{k-1}} f(\tau, b(k-1)/k), \tag{2.6}$$

$$\textit{in } \mathrm{SBM}(n,k,a,b) \quad \textit{if} \quad a > a_k(b), \quad \textit{where } a_k(0) = k. \tag{2.7}$$

Remark 2.14. Note that (2.7) approaches the optimal bound given by the presence of the giant at $b = 0$, and we further conjecture that $a_k(b)$ gives the correct first-order approximation of the information-theoretic bound for small b.

Remark 2.15. Note that the $k\ln k$ scaling in (2.6) improves significantly on the KS threshold given by $b > k(k-1)$ at $a = 0$. Relatedly, note that the k-colorability threshold for Erdős-Rényi graphs grows as $2k\ln k$ [9].

Remark 2.16. We also believe that the above gives the correct scaling in k for $a = 0$; i.e., that for $b < (1-\varepsilon)k\ln(k) + o_k(1)$, $\varepsilon > 0$, detection is information-theoretically impossible. To see this, consider $v \in G$, $b = (1-\epsilon)k\ln(k)$, and assume that we know the communities of all vertices more than $r = \ln(\ln(n))$ edges away from v. For each vertex that is r edges away from v, there will be approximately k^ϵ communities in which it has no neighbors. Then vertices $r-1$

edges away from v have approximately $k^{\epsilon}\ln(k)$ neighbors that are potentially in each community, with approximately $\ln(k)$ fewer neighbors suspected of being in its community than in the average other community. At that point, the noise has mostly drowned out the signal and our confidence that we know anything about the vertices' communities continues to degrade with each successive step towards v.

In [13, 14], a different approach than the one described in the previous remark is developed based on a contiguity argument and estimates from [9], showing that the scaling in k is in fact tight.

2.3 Learning the Model

To estimate the parameters, we count cycles of slowly growing length as already done in [61] for $k = 2$, using nonbacktracking walks to approximate the count.

LEMMA 2.17. *If* SNR > 1, *there exists a consistent and efficient estimator for the parameters* a, b, k *in* $\mathrm{SBM}(n, k, a, b)$.

3 Achieving the KS Threshold Efficiently: Proof Technique

Recall the parameters: k and n are positive integers, $p \in (0,1)^k$ with $\sum p_i = 1$, and Q is a $k \times k$ symmetric matrix with nonnegative entries. Let $\Omega_1, \dots, \Omega_k$ be the communities and P be the $k \times k$ diagonal matrix such that $P_{i,i} = p_i$ for each i. Now, consider the 2-community symmetric stochastic block model. In this case, $k = 2$, $p = [\frac{1}{2}, \frac{1}{2}]$, $Q_{i,j}$ is a if $i = j$ and b otherwise for some a, b. Let $\lambda_1 = \frac{a+b}{2}$ be the average degree of a vertex in a graph drawn from this model, and $\lambda_2 = \frac{a-b}{2}$ be the other eigenvalue of PQ. Throughout this section we say f is approximately g or $f \approx g$ when $|f - g| = o(|f + g|)$ with probability $1 - o(1)$.

Our goal is to determine which of v's vertices are in each community with an accuracy that is nontrivially better than that attained by random guessing. Obviously, the symmetry between communities ensures that we can never tell whether a given vertex is in community 1 or community 2, so the best we can hope for is to divide the vertices into two sets such that there is a nontrivial difference between the fraction of vertices from community 1 that are assigned to the first set and the fraction of vertices from community 2 that are assigned to the first set.

3.1 Amplifying Random Guesses and Nonbacktracking Walks

Consider dividing the vertices into two sets at random. The difference between the fraction of the vertices from community 1 assigned to the first set and the fraction of the vertices from community 2 assigned to the first set will typically have a magnitude on the order of $1/\sqrt{n}$ (by the central limit theorem). For the rest of this section, we will consider the difference between these fractions to be fixed.

Once we have even such weak information on which vertex is in which community, we can try to improve our classification of a given vertex by factoring in our knowledge of what communities the nearby vertices are in. Given a vertex v and

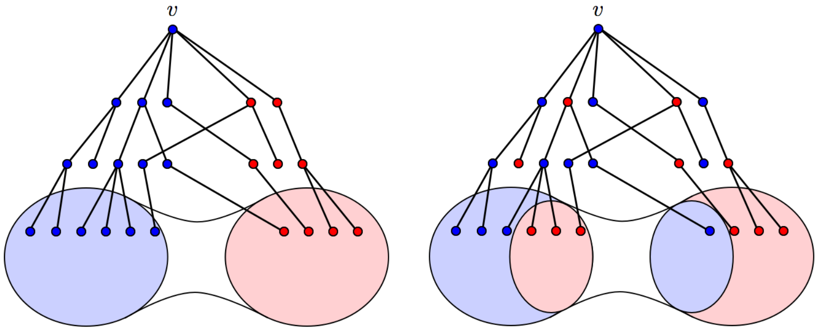

FIGURE 3.1. The left figure shows the neighborhood of vertex v pulled from the SBM graph at depth $c \log_{\lambda_1} n$, $c < \frac{1}{2}$, which is a tree with high probability. If one had an educated guess about each vertex's label, of good enough accuracy, then it would be possible to amplify that guess by considering only such small neighborhoods (deciding with the majority at the leaves). However, we do not have such an educated guess. We thus initialize our labels purely at random, obtaining a small advantage of roughly $\sqrt{n}$ vertices by luck (i.e., the central limit theorem), in either an agreement or disagreement form. This is illustrated in agreement form in the right figure. We next attempt to amplify that lucky guess by exploiting the information of the SBM graph. Unfortunately, the graph is too sparse to let us amplify that guess by considering treelike or even loopy neighborhoods; the vertices would have to be exhausted. This takes us to considering walks.

a small integer t, the expected number of vertices t edges away from v is approximately $(\frac{a+b}{2})^t$, and the expected difference between the number of these vertices in the same community as v and the number of them in the other community is approximately $(\frac{a-b}{2})^t$. So we could classify the vertices with an accuracy of

$$\frac{1}{2} + \theta\left(\left(\frac{(a-b)^2}{2(a+b)}\right)^{t/2} \Big/ \sqrt{n}\right)$$

based on the sets the vertices t edges away from them were assigned to. That suggests that if $(a-b)^2 > 2(a+b)$, then for large enough t we would be able to classify the vertices with accuracy $\frac{1}{2} + \Omega(1)$. Unfortunately, that would require t large enough that the graph would run out of vertices before we got t edges away from the vertex we were trying to classify.

An obvious way to solve the problem caused by running out of vertices would be to simply count the walks of length t from v to vertices in S_1 or S_2. Recall that a *walk* is a series of vertices such that each vertex in the walk is adjacent to the next, and a *path* is a walk with no repeated vertices. The last vertex of such a walk will be adjacent to an average of approximately $a/2$ vertices in its community outside the walk and $b/2$ vertices in the other community outside the walk. However, it will also be adjacent to the second-to-last vertex of the walk, and maybe some of the other vertices in the walk as well. As a result, the number of walks of length t from v to vertices in S_1 or S_2 cannot be easily predicted in terms of v's community. So, the numbers of such walks are not useful for classifying vertices.

We could deal with this issue by counting paths of length t from v to vertices in S_1 and S_2.[10] The expected number of paths of length t from v is approximately $(\frac{a+b}{2})^t$, and the expected difference between the number that end in vertices in the same community as v and the number that end in the other community is approximately $(\frac{a-b}{2})^t$. The problem with this is that counting all of these paths is inefficient.

The compromise we use is to count nonbacktracking walks ending at v, i.e., walks that never repeat the same edge twice in a row. We can efficiently determine how many nonbacktracking walks of length t there are from vertices in S_i to v. Furthermore, most nonbacktracking walks of a given length that is logarithmic in n are paths, so it seems reasonable to expect that counting nonbacktracking walks instead of paths in our algorithm will have a negligible effect on the accuracy.

More precisely, that suggests the following approach. Define $y_{v,v'}^{(t)}$ to be the number of nonbacktracking walks of length t that start at vertices in S_2 and end in the directed edge (v', v) minus the number of nonbacktracking walks of length t that start at vertices in S_1 and end in (v', v). Also, define $y_v^{(t)}$ to be the overall difference between the number of nonbacktracking walks of length t from vertices in S_2 to v and the number of nonbacktracking walks of length t from vertices in S_1 to v. Their values can be efficiently computed by means of the following procedure:

```
(1) For every $(v, v') \in E(G)$:
        If $v' \in S_2$, set $y_{v,v'}^{(1)} = 1$
        Otherwise, set $y_{v,v'}^{(1)} = -1$
(2) For every $1 < t \leq m$ and $(v, v') \in E(G)$:
        Set $y_{v,v'}^{(t)} = \sum_{v'':(v'',v') \in E(G), v'' \neq v} y_{v',v''}^{(t-1)}$
(3) For every $(v, v') \in E(G)$ and $v \in G$
        Set $y_v^{(m)} = \sum_{v':(v,v') \in E(G)} y_{v,v'}^{(m)}$
```

One way of viewing this algorithm is that $y_{v,v'}^{(t)}$ represents our current belief about what community v' is in, disregarding any information derived from the fact that it is next to v. We start with fairly unconfident beliefs about the vertices' communities, and then derive more and more confident beliefs about the vertices' communities by taking our beliefs about their neighbors' communities into account.

[10] This type of approach is considered in [16].

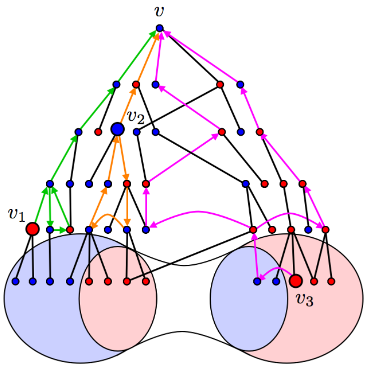

FIGURE 3.2. This figure extends Figure 3.1 to a larger neighborhood. The ABP algorithm amplifies the belief of vertex v by considering all the walks of a given length that end at it. To avoid being disrupted by backtracking or cycling the beliefs on short loops, the algorithm considers only walks that do not repeat the same vertex within r steps, i.e., r-nonbacktracking walks. For example, when $r = 3$ and when the walks have length 7, the green walk starting at vertex v_1 is discarded, whereas the orange walk starting at the vertex v_2 is counted. Note also that the same vertex can lead to multiple walks, as illustrated with the two magenta walks from v_3. Since there are approximately equally many such walks between any two vertices, if the majority of the vertices were initially classified as blue, this is likely to classify all of the vertices as blue. We hence need a compensation step to prevent the classification from becoming biased towards one community.

3.2 Compensating for the Average Value

The problem with this plan is that the average value of $y^{(1)}_{v,v'}$ will not be exactly 0. It will also tend to have an absolute value on the order of $1/\sqrt{n}$. That means that the average value over all $(v, v') \in E(G)$ of $y^{(t)}_{v,v'}$ will have an absolute value of $\Theta((\frac{a+b}{2})^t/\sqrt{n})$. So, the degrees of v and the nearby vertices will tend to affect the value of $y^{(m)}_v$ much more than the community of v will, which would render attempts to classify v based on $y^{(m)}_v$ ineffective.

Remark 3.1. The simple way to fix this would be to add a step where we subtract the average value of $y^{(t)}$ from every element of $y^{(t)}$ so its sum is 0 for every t like we did in the version of ABP in Section 2.1. However, this does not extend easily to the general case, and we want a solution that does.

In order to prevent this, we need to stop the average value of $y^{(t)}_{v,v'}$ from getting too large. It will tend to multiply by roughly $\frac{a+b}{2}$ each time t increases by 1, so the average value of $y^{(t)}_{v,v'} - \frac{a+b}{2} y^{(t-1)}_{v,v'}$ will probably be much smaller than the average value of $y_{v,v'}^{(t)}$. So, if we pick some $1 < i \le m$ and redefine $y_{v,v'}^{(i)}$ so that

$$y_{v,v'}^{(i)} = -\frac{a+b}{2} y_{v,v'}^{(i-1)} + \sum_{v'':(v'',v')\in E(G), v''\neq v} y_{v',v''}^{(i-1)} \tag{3.1}$$

for all $(v, v') \in E(G)$, the average value of $y_{v,v'}^{(i)}$ will be much smaller than it would have been, while the difference between the average values over v' in different communities of $y_{v,v'}^{(i)}$ will be roughly the same.

However, the average value of $y_{v,v'}^{(i)}$ will still be nonzero, and if we continue to set

$$y_{v,v'}^{(t)} = \sum_{\substack{v'':(v'',v')\in E(G),\\ v''\neq v}} y_{v',v''}^{(t-1)}$$

for all $t > i$, the average value of $y_{v,v'}^{(t)}$ would resume increasing in magnitude faster than the average difference between $y_{v,v'}^{(t)}$ for different communities of v'. This creates the risk that it would still eventually get too large. So, in order to actually fix the problem, it may be necessary to repeat the step where its magnitude is reduced. More precisely, we may have to choose several indices $t_0, t_1, \dots t_{m'}$ and redefine $y_{v,v'}^{(t_i)}$ for each i so that

$$y_{v,v'}^{(t_i)} = -\frac{a+b}{2} y_{v,v'}^{(t_i-1)} + \sum_{v'':(v'',v')\in E(G), v''\neq v} y_{v',v''}^{(t_i-1)} \tag{3.2}$$

for every $(v, v') \in E(G)$.

Once we have made these modifications, it will be the case that for sufficiently large m, the average value for v' in community 1 of $y_{v,v'}^{(m)}$ will differ from the average value for v' in community 2 of $y_{v,v'}^{(m)}$ by a constant multiple of the standard deviation of $y_{v,v'}^{(m)}$. Then, we define $y_v^{(m)} = \sum_{v':(v,v')\in E(G)} y_{v,v'}^{(m)}$ in order to simplify it to a function of one vertex that is still correlated with the vertex's community. Finally, we randomly assign each vertex to a community with a probability that scales linearly with $y_v^{(m)}$ because it is easier to prove that the algorithm works that way.

3.3 Vanilla ABP

We present first a simplified version of our algorithm. Our proof relies on a modified version described below, but this version captures the essence of our algorithm while avoiding technicalities required for the proof.

ABP*(G, m, m', r, λ_1):

(1) For each adjacent v and v' in G, randomly draw $y^{(1)}_{v,v'}$ from a Gaussian distribution with mean 0 and variance 1. Also, consider $y^{(t)}_{v,v'}$ as having a value of 0 whenever $t < 1$.

(2) For each $1 < t \le m$, and each adjacent v and v' in G, set

$$y^{(t)}_{v,v'} = \sum_{\substack{v'':(v',v'')\in E(G),\\ v''\neq v}} y^{(t-1)}_{v',v''}$$

unless (v, v') is part of a cycle of length r or less.[11] If it is, then let the other vertex in the cycle that is adjacent to v be v''', and the length of the cycle be r', and set

$$y^{(t)}_{v,v'} = \sum_{\substack{v'':(v',v'')\in E(G),\\ v''\neq v}} y^{(t-1)}_{v',v''} - \sum_{\substack{v'':(v,v'')\in E(G),\\ v''\neq v, v''\neq v''''}} y^{(t-r')}_{v,v''}$$

unless $t = r'$, in which case, set

$$y^{(t)}_{v,v'} = \sum_{\substack{v'':(v',v'')\in E(G),\\ v''\neq v}} y^{(t-1)}_{v',v''} - y^{(1)}_{v''',v}.$$

(3) Set Y to be the $n \times m$ matrix such that for all t and v,

$$Y_{v,t} = \sum_{v':(v',v)\in E(G)} y^{(t)}_{v,v'}.$$

(4) Set M to be the $m \times m$ matrix such that $M_{i,i} = 1$ and $M_{i,i+1} = -\lambda_1$ for all i, and all other entries of M are 0. Also, let $e_m \in \mathbb{R}^m$ be the vector with an m^{th} entry of 1 and all other entries equal to 0. Set

$$y' = YM^{m'}e_m.$$

(5) Return $(\{v : y'_v > 0\}, \{v : y'_v \le 0\})$.

Remark 3.2. In the $r = 2$ case, one does not need to find cycles and one can exit step 2 after the second line. In general, when running this algorithm, one should use variables that are accurate to within a factor of less than $(\lambda_2/\lambda_1)^m$. As mentioned above, we rely on a less compact version of the algorithm to prove the theorem, but expect that the above also succeeds at detection as long as $m > 2\ln(n)/\ln(\text{SNR}) + \omega(m')$ and $m' > m\ln(\lambda_1^2/\lambda_2^2)/(\ln(n) - \omega(1))$.

Remark 3.3. This version of ABP is mostly the same as the one in Section 2.1. However, the previous version compensates for biases in $y^{(1)}$ by subtracting the average value of $y^{(t)}$ from each entry in $y^{(t)}$ for every t in order to prevent it from

[11] See Remark (2) on page 1345 for how to modify the algorithm in the case of multiple cycles.

ever becoming significantly biased. This version compensates for biases by a variant of the method explained in the previous subsection instead. More specifically, it uses the method explained in the previous subsection except that it calculates all of the $y^{(t)}$ without using any form of compensation and then adjusts the results in order to essentially apply the compensation retroactively.

Implementation details. Note that ABP* has several differences from the algorithm outlined in previous sections. First of all, we initialize the $y^{(0)}_{v,v'}$ using a random value that is drawn from a normal distribution because a probability distribution that is a multidimensional normal distribution is easier to analyze than a probability distribution that is evenly distributed over the vertices of an n-dimensional hypercube. Second, we require that our walks never repeat the same vertex within r steps for some r, rather than merely requiring that they not backtrack. This allows us to use the expected number of walks between vertices in our analysis without worrying about the tiny probability that there is a dense tangle in the graph with a huge number of nonbacktracking walks between its vertices. Making this modification to the algorithm requires adding an extra part to the recursion step where walks that just repeated a vertex are cancelled out, specifically the second half of step 3 of the algorithm above. The resulting algorithm is called the *acyclic belief propagation* algorithm because it counts walks that do not contain any small cycles. Third, we move all of the recursion steps that compensate for the average value to the end of the algorithm. This is possible because the operation that takes $y^{(t-1)}$ as input and outputs a list that has a value of

$$-\frac{a+b}{2}y^{(t-1)}_{v,v'} + \sum_{\substack{v'':(v'',v')\in E(G),\\ v''\neq v}} y^{(t-1)}_{v',v''}$$

for each $(v,v') \in E$ commutes with the one that simply outputs a list that has a value of

$$\sum_{\substack{v'':(v'',v')\in E(G),\\ v''\neq v}} y^{(t-1)}_{v',v''}$$

for each $(v,v') \in E$. The algorithm generates $y^{(m)}$ by applying these two operations in some sequence to $y^{(1)}$, so we can calculate it by applying the later operation $m-m'$ times and then applying the former operation m' times. Actually, the algorithm takes this one step further by applying the latter operation m times and then calculating how the result would have changed if it had been applied to the former the appropriate number of times, but it still has the same result.

The full Acyclic Belief Propagation algorithm also has a few differences from ABP* that make it easier to prove that it works. In particular, we randomly select a small fraction of the graph's edges at the beginning of the algorithm. Then we require one specific step of each nonbacktracking walk to use one of the selected edges, and all of their other steps to use edges that have not been selected. Since the selected edges are nearly independent of the rest of the graph, this allows us to more easily prove that the values of $y^{(t-1)}_{v',v''}$ for v' adjacent to v will not become dependent in a way that disrupts the algorithm.

3.4 Spectral View of ABP

An alternative perspective on this algorithm is the following. Assume for the moment that there are exactly $n/2$ vertices in each community, and let M be the expected adjacency matrix, the matrix such that $M_{v,v'}$ is a/n if v and v' are in the same community and b/n if they are not. This matrix has an eigenvector whose entries are all 1 with eigenvalue $\frac{a+b}{2}$, an eigenvector whose entries are ± 1 with their signs determined by the relevant vertices' communities that have eigenvalue $\frac{a-b}{2}$ and all of their other eigenvalues are 0.

Now, let M' be the graph's actual adjacency matrix. The result above suggests that the second eigenvector of M' may have entries that are correlated with the vertices' communities. The problem with this reasoning is that while M' has an expected value of M, $(M')^2$ has an expected value of roughly $M^2 + \frac{a+b}{2}I$ because for every i, j, $M'_{i,j} = 1 \iff M'_{j,i} = 1$, with the result that $E[\sum_j M'_{i,j} \cdot M'_{j,i}]$ is very different from $\sum_j E[M'_{i,j}] \cdot E[M'_{j,i}]$. In other words, the square of the adjacency matrix counts walks of length 2 and has an expected value that is significantly different from the square of the expected adjacency matrix due to backtracking.

In order to avoid this issue, we define the graph's nonbacktracking walk matrix W as a matrix over the vector space with an orthonormal basis consisting of a vector for each directed edge in the graph. $W_{(v_1,v_2),(v'_1,v'_2)}$ is defined to be 1 if $v'_2 = v_1$ and $v_2 \neq v'_1$ and 0 otherwise. In other words, it has a 1 for every case where one directed edge leads to another that is not the same edge in the other direction.

Now, let $w \in R^{2|E(G)|}$ be the vector whose entries are all 1, and $w' \in R^{2|E(G)|}$ be the vector such that $w'_{(v_0,v_1)}$ is 1 if v_0 is in community 1 and -1 if v_0 is in community 2. As mentioned before, for a small t and a random $(v, v') \in E(G)$, there will be an average of approximately $(\frac{a+b}{2})^t$ directed edges t edges in front of (v, v'), and approximately $(\frac{a-b}{2})^t$ more of these edges will have ending vertices in the same community as v' than in the other community on average. So, $w \cdot W^t w \approx 2|E(G)|(\frac{a+b}{2})^t$ and $w' \cdot W^t w' \approx 2|E(G)|(\frac{a-b}{2})^t$. That strongly suggests that W has eigenvectors that are correlated with w and w' that have eigenvalues of approximately $\frac{a+b}{2}$ and $\frac{a-b}{2}$, respectively. It also seems plausible that W's other eigenvalues have relatively small magnitudes.

If this is true, then one can gain information on which vertices of G are in each community from the second eigenvector of W. One could simply calculate the second eigenvector of W directly. However, it is significantly faster to pick a random

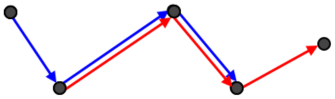

FIGURE 3.3. These two paths (blue and red) lead to an entry of 1 in the $W^{(4)}$ matrix.

vector w'' and then compute $W^m w''$ for some suitable m. The resulting vector will be approximately a linear combination of W's main eigenvectors. Unfortunately, it will be much closer to being a multiple of its first eigenvector than its second. If we multiply $(W - \frac{a+b}{2}I)$ by the resulting vector, such as in (3.1), the component of the vector that is proportional to the first eigenvector will be mostly cancelled out. However, since its eigenvalue is not exactly $\frac{a+b}{2}$, it will not be cancelled out completely and might still be too large. Luckily, if we instead multiply $(W - \frac{a+b}{2}I)^{m'}$ by the resulting vector for suitable m', such as in (3.2), the component of the vector that is proportional to the first eigenvalue will be essentially cancelled out, leaving a vector that is approximately a multiple of W's second eigenvector and thus correlated with G's communities. We believe that this would succeed in detecting communities in the SBM, but in order to make it easier to prove that our algorithm works we actually use r-nonbacktracking walks. This corresponds to using the graph's r-nonbacktracking walk matrix, which is defined as follows.

DEFINITION 3.4. For any r, the graph's *r-nonbacktracking walk matrix*, $W^{(r)}$, is a matrix over the vector space with an orthonormal basis consisting of a vector for each directed path of length $r-1$ on the graph. $W^{(r)}_{(v_1,v_2,\dots,v_r),(v'_1,v'_2,\dots,v'_r)}$ is 1 if $v'_{i+1} = v_i$ for each $1 \le i < r$ and $v'_1 \ne v_r$; otherwise it is 0. In other words, $W^{(r)}$ maps a path of length $r-1$ to the sum of all paths resulting from adding another element to the end of the path and deleting its first element.

Performing these calculations is essentially what the acyclic belief propagation algorithm does. From this perspective, the algorithm roughly translates to the following:

(1) Choose $y^{(1)}$ randomly such that each element is independently drawn from a normal distribution.
(2) For each $1 < t \le m$, let $y^{(t)} = W^{(r)} y^{(t-1)}$.
(3) Change $y^{(m)}$ to $(W^{(r)} - \frac{a+b}{2}I)^{m'} y^{(m-m')}$, where $m' = \lceil \frac{m-3r-1}{l} \rceil$.
(4) Set $y'_v = \sum_{v_1,\dots,v_{r-1}} y^{(m)}_{(v_1,\dots,v_{r-1},v)}$ for every $v \in G$. Return $(\{v : y'_v > 0\}, \{v : y'_v \le 0\})$.

Even though $r = 2$ might suffice to achieve the KS threshold in the SBM, the use of a larger r might help for other graph models, e.g., those having more short cycles.

3.5 ABP for the General SBM

Now, consider a graph G drawn from $\mathrm{SBM}(n, p, Q/n)$ with arbitrary p and Q. Also, let $\lambda_1, \dots, \lambda_h$ be the distinct eigenvalues of PQ in order of nonincreasing magnitude. If the parameters are such that vertices from different communities have different expected degrees, then one can detect communities by simply dividing its vertices into those with above-average degrees and those with below-average degrees. So, assume that the expected degree of a vertex is independent of its community. Detecting communities in the general case runs into some obstacles that do not apply in the 2-community symmetric case. First of all, it is much less clear that assigning vertices to sets randomly is a useful start. Also, even if we did have reasonable preliminary guesses of which community each vertex was in, it is not obvious how to determine a vertex's community based on the alleged communities of the vertices a fixed distance from it.

For the moment, assume that for each vertex v, we have a vector x_v such that we believe v is in community i with probability $p_i + x_v \cdot e_i$ for each i, where all elements of x_v are small. Furthermore, assume that x_v is generated independently of v's neighbors. The correct belief about the probability that v is in each community once its neighbors are taken into account is

$$p + x_v + \frac{1}{\lambda_1} \sum_{v':(v,v') \in E[G]} PQ x_{v'}$$

up to nonlinear terms in the x's. So, given m small enough that the set of vertices within m edges of v is a tree, the correct belief about what community v is in once all of the vertices within m edges of v are taken into account is

$$p + \sum_{0 \le m' \le m} \lambda_1^{-m'} \sum_{v':d(v,v')=m'} (PQ)^{m'} x_{v'}$$

up to nonlinear terms in the x's. So, the logical belief about the probability that v is in each community based only on the preliminary guesses concerning the vertices m edges away from v is

$$p + \lambda_1^{-m} \sum_{v':d(v,v')=m} (PQ)^m x_{v'}.$$

Conveniently, this expression is linear, so if w is an eigenvector of PQ with eigenvalue λ_i for $i \neq 1$, then

$$\begin{aligned} E[w \cdot P^{-1} e_{\sigma_v}] &\approx w \cdot P^{-1} \Big(p + \lambda_1^{-m} \sum_{v' : d(v,v') = m} (PQ)^m x_{v'} \Big) \\ &= \lambda_1^{-m} \lambda_i^m \sum_{v' : d(v,v') = m} w \cdot P^{-1} x_{v'}. \end{aligned}$$

In particular, this means that we only need an initial estimate for $w \cdot P^{-1} e_{\sigma_{v'}}$ for every vertex in the graph, rather than needing a full set of beliefs about the vertices' communities. Any random guesses we make will probably have correlation $\pm\Omega(1/\sqrt{n})$ with $w \cdot P^{-1} e_{\sigma_{v'}}$, so we can use them as a starting point.

Unfortunately, just like in the 2-community symmetric case, the graph will run out of vertices before m becomes large enough to amplify our beliefs enough. However, switching from a sum over all vertices v' that are m edges away from v to a sum over all nonbacktracking walks of length m ending in a vertex v' fixes this problem the same way it does in the 2-community symmetric case. Likewise, we can still compute this sum by randomly dividing G's vertices between two sets S_1 and S_2 and then using the following algorithm:

```
(1) For every $(v, v') \in E(G)$ :
        If $v' \in S_2$, set $y^{(1)}_{v,v'} = 1$
        Otherwise, set $y^{(1)}_{v,v'} = -1$
(2) For every $1 < t \leq m$ and $(v, v') \in E(G)$ :
        Set $y^{(t)}_{v,v'} = \sum_{v'' : (v'', v') \in E(G), v'' \neq v} y^{(t-1)}_{v',v''}$
(3) For every $(v, v') \in E(G)$ and $v \in G$
        Set $y^{(m)}_v = \sum_{v' : (v,v') \in E(G)} y^{(m)}_{v,v'}$
```

However, needing to compensate for the average value is a special case of a considerably more complicated phenomenon. The average value over all $(v, v') \in E(G)$ of $y^{(1)}_{v,v'} \cdot w_{\sigma_{v'}}$ will typically have a magnitude of $\Theta(1/\sqrt{n})$, and for general t the average value over all $(v, v') \in E(G)$ of $y^{(t)}_{v,v'} \cdot w_{\sigma_{v'}}$ will typically have a magnitude of $\Theta(|\lambda_i^t|/\sqrt{n})$. Now, let w' be an eigenvector of PQ with an eigenvalue of $\lambda_{i'}$ that has greater magnitude than λ_i. Then the average value over all $(v, v') \in E(G)$ of $y^{(t)}_{v,v'} \cdot w'_{\sigma_{v'}}$ will typically have a magnitude of $\Theta(|\lambda_{i'}^t|/\sqrt{n})$. Since this grows faster than $y^{(t)}_{v,v'} \cdot w_{\sigma_{v'}}$ does, it will eventually become large enough to disrupt efforts to estimate $w_{\sigma_{v'}}$ using $y^{(t)}_{v,v'}$ the same way the average value of $y^{(t)}$ did in the 2-community symmetric case. In fact, the issue with the average value is just the subcase of this when $i' = 1$. So, in order to deal with this, we need to compensate for each eigenvalue, $\lambda_{i'}$, of PQ with magnitude greater than λ_i by choosing several indices $t_{0,i'}, t_{1,i'}, \ldots t_{m',i'}$ and redefining $y^{(t_{j,i'})}_{v,v'}$ for each j

so that

$$y_{v,v'}^{(t_{j,i'})} = -\lambda_{i'} y_{v,v'}^{(t_{j,i'}-1)} + \sum_{v'':(v'',v')\in E(G),v''\neq v} y_{v',v''}^{(t_{j,i'}-1)}$$

for every $(v, v') \in E(G)$. Assuming that this is done, $y_{v,v'}^{(1)}$ has a variance of approximately 1, and then $y_{v,v'}^{(t)}$ has a variance of roughly λ_1^t. It becomes possible to determine which community v is in with accuracy nontrivially greater than that obtained by guessing randomly based on $y_{v,v'}^{(t)}$ when the expected difference between its values for v in different communities is within a constant factor of its standard deviation. In other words, t needs to be large enough that $|\lambda_i^t|/\sqrt{n}$ is significant relative to $\sqrt{\lambda_1^t}$. If $\lambda_1 \geq \lambda_i^2$ then this will never happen, so the algorithm requires

$$\lambda_i^2 > \lambda_1.$$

The general acyclic belief propagation algorithm is almost the same as the 2-community symmetric version. However, it takes a list of eigenvectors as input instead of just λ_1. Also, the step compensating for larger eigenvalues is changed from "Set $y^{(m)} = YM^{\lceil \frac{m-3r-1}{l} \rceil} e_m$ where M is the matrix such that $M_{i,i} = 1$ for all i, $M_{i,i+1} = -(1-\gamma)\lambda_1$ for all i, and all other entries of M are 0" to "Set $y^{(m)} = Y \prod_{s'<s} M_{s'}^{\lceil \frac{m-r-(2r+1)s'}{l} \rceil} e_m$ where $M_{s'}$ is the matrix such that $(M_{s'})_{i,i} = 1$ for all i, $(M_{s'})_{i,i+1} = -(1-\gamma)\lambda_{s'}$ for all i, and all other entries of $M_{s'}$ are 0." Its effectiveness is described by the following theorem.

THEOREM 3.5. *Let $p \in (0,1)^k$ with $\sum p = 1$, Q be a symmetric matrix with nonnegative entries, P be the diagonal matrix such that $P_{i,i} = p_i$, and $\lambda_1, \dots, \lambda_h$ be the eigenvalues of PQ in order of nonincreasing magnitude. If $\lambda_2^2 > \lambda_1$, then there exist constants ϵ, r, c and $m = \Theta(\log(n))$ such that when ABP is run on these parameters and a random $G \in \mathrm{SBM}(n, p, Q/n)$, with probability $1 - o(1)$ there exist σ and σ' such that the difference between the fraction of vertices from community σ that are in S_1 and the fraction of vertices from community σ' that are in S_1 is at least ϵ. The algorithm can be run in $O(n \log n)$ time.*

Remark 3.6. Let $s = 3$ if $|\lambda_2| = |\lambda_3|$ and $s = 2$ otherwise. This theorem could alternately have stated that there exists ϵ such that for any two communities σ and σ' such that there exists an eigenvector w of PQ with eigenvalue λ_s such that $w_\sigma \neq w_{\sigma'}$, the expected difference between the fraction of vertices from community σ that are in S_1 and the fraction of vertices from community σ' that are in S_1 is at least ϵ.

There are also a couple of other variants of these ideas that may be useful for community detection. For instance, if we pick $r \approx \ln(\ln(n))$ and then define

Σ to be the $n \times n$ symmetric matrix such that $\Sigma_{v,v'}$ is the number of nonbacktracking walks of length r between v and v', we suspect that Σ's eigenvector of second-largest magnitude will have entries that are correlated with the corresponding vertices' communities. As in the standard case, we expect that we could get an approximation of this eigenvector by taking a random vector w and then computing $(\Sigma - \lambda_1' I)^{m'} \Sigma^{m-m'} w$ for suitable m and m' where λ_1' is an estimate of Σ's largest eigenvalue.

We can compute Σ as follows. First, let $\Sigma^{(t)}$ be the $n \times n$ matrix such that $\Sigma^{(t)}_{v,v'}$ is the number of nonbacktracking walks of length t between v and v'. Then $\Sigma^{(0)} = I$, $\Sigma^{(1)}$ is the graph's adjacency matrix, and $\Sigma^2_{v,v'}$ is equal to the number of shared neighbors v and v' have for all v and v'. For every $t > 2$, we have that $\Sigma^{(t)} = \Sigma^{(1)} \cdot \Sigma^{(t-1)} - D \cdot \Sigma^{(t-2)}$, where D is the diagonal matrix such that $D_{v,v}$ is one less than the degree of v for all v. This can be used to efficiently compute $\Sigma = \Sigma^{(r)}$.

Also, instead of prohibiting repeating a vertex within r steps, we could address the issue of tangles by dividing G's edges between sets $E_0, \dots, E_{m'}$ for suitable m' such that most of the edges are assigned to E_0 and the rest are assigned to one of the others at random. Then we count nonbacktracking walks with the restriction that edge r of the walk must be from E_1, edge $2r$ must be from E_2, and so on, while all other edges must be from E_0 for suitable r. The periodic prohibitions on using edges from E_0 would force the walk to leave any tangle it had been in, while the fact that most of the edges are chosen from E_0 prevents the restriction from reducing the number of walks too severely.

4 Crossing the KS Threshold: Proof Technique

Recall that the algorithm samples a typical clustering uniformly at random in the typical set

$$
\begin{aligned}
T_\delta(G) = \Bigg\{ & x \in \mathrm{Bal}(n,k,\delta) : \\
& \sum_{i=1}^{k} \left| \left\{ G_{u,v} : (u,v) \in \binom{[n]}{2} \text{ s.t. } x_u = i, x_v = i \right\} \right| \geq \frac{an}{2k}(1-\delta), \\
& \sum_{i,j \in [k], i<j} \left| \left\{ G_{u,v} : (u,v) \in \binom{[n]}{2} \text{ s.t. } x_u = i, x_v = j \right\} \right| \\
& \qquad \leq \frac{bn(k-1)}{2k}(1+\delta) \Bigg\},
\end{aligned}
$$

where the previous two inequalities apply to the case $a > b$ and are flipped if $a < b$. A first question is to estimate the likelihood that a bad clustering—i.e., one that has an overlap that is close to $1/k$—belongs to the typical set. This means the probability that a clustering which splits each of the true clusters into k roughly equal-sized groups belonging to each community still manages to keep the right proportions of edges inside and across the clusters. This is unlikely to take place, but we need a small enough exponent on this rare event probability.

The number of edges that are contained in the clusters of a bad clustering is roughly distributed as the sum of two binomial random variables,

$$E_{\text{in}} \mathrel{\dot\sim} \operatorname{Bin}\left(\frac{n^2}{2k^2}, \frac{a}{n}\right) + \operatorname{Bin}\left(\frac{(k-1)n^2}{2k^2}, \frac{b}{n}\right), \tag{4.1}$$

where we use $\dot\sim$ to emphasize that this is an approximation since the bad clustering may not be perfectly balanced. Note that the expectation of the above distribution is $\frac{n}{2k}\frac{a+(k-1)b}{k}$. In contrast, the true clustering would have a distribution given by $\operatorname{Bin}(\frac{n^2}{2k}, \frac{a}{n})$, which would give an expectation of $\frac{an}{2k}$. In turn, the number of edges that are crossing the clusters of a bad clustering is roughly distributed as

$$E_{\text{out}} \mathrel{\dot\sim} \operatorname{Bin}\left(\frac{n^2(k-1)}{2k^2}, \frac{a}{n}\right) + \operatorname{Bin}\left(\frac{n^2(k-1)^2}{2k^2}, \frac{b}{n}\right), \tag{4.2}$$

which has an expectation of $\frac{n(k-1)}{2k}\frac{a+(k-1)b}{k}$. In contrast, the true clustering would have the above replaced by $\operatorname{Bin}(\frac{n^2(k-1)}{2k}, \frac{b}{n})$, and an expectation of $\frac{bn(k-1)}{2k}$.

Thus, we need to estimate the rare event that the binomial sum deviates from its expectations. While there is a large list of bounds on binomial tail events, the number of trials here is quadratic in n, and the success bias decays linearly in n, which requires particular care to ensure tight bounds. We derive these, obtaining that for a bad clustering x_{bad}, $\mathbb{P}\{x_{\text{bad}} \in T_\delta(G) \mid x_{\text{bad}} \in B_\epsilon\}$ behaves when ε, δ are arbitrarily small as

$$\exp\left(-\frac{n}{k}A\right)$$

where

$$A := \frac{a+b(k-1)}{2}\ln\frac{k}{a+(k-1)b} + \frac{a}{2}\ln a + \frac{b(k-1)}{2}\ln b.$$

Using the fact that $|T_\delta(G)| \geq 1$ with high probability, since the planted clustering is typical with high probability, and from a union bound on k^n possible bad clusterings, one obtains

$$\begin{aligned}\mathbb{P}\{\widehat{X}(G) \in B_\epsilon\} = \mathbb{E}_G \frac{|T_\delta(G) \cap B_\epsilon|}{|T_\delta(G)|} &\leq \mathbb{E}_G|T_\delta(G) \cap B_\epsilon| + o(1) \\ &\leq k^n \cdot \mathbb{P}\{x_{\text{bad}} \in T_\delta(G) | x_{\text{bad}} \in B_\epsilon\} + o(1).\end{aligned}$$

Checking when the above upper bound vanishes already gives a regime that crosses the KS threshold when $k \geq 5$, and scales properly in k when $a = 0$. However, it does not interpolate the correct behavior of the information-theoretic bound in the extreme regime of $b = 0$ and does not cross at $k = 4$. In fact, for

$b = 0$, the union bound requires $a > 2k$ to imply no bad typical clustering with high probability, whereas as soon as $a > k$, an algorithm that simply separates the two giants in $\mathrm{SBM}(n, k, a, 0)$ and assigns communities uniformly at random for the other vertices solves detection. Thus when $a \in (k, 2k]$, the union bound is loose. To remedy this, we next take into account the topology of the SBM graph to tighten our bound on $|T_{\delta}(G)|$.

Since the algorithm samples a typical clustering, we only need the number of bad and typical clusterings to be small compared to the total number of typical clusterings in expectation. We can thus get a tighter bound on the probability of error of the TS algorithm by obtaining a tighter bound on the typical set size than simply 1. We proceed here with three levels of refinements to bound the typical set size. First we exploit the large fraction of nodes that are in treelike components outside of the giant. Conditioned on being on a tree, the SBM labels are distributed as in a broadcasting problem on a (Galton-Watson) tree. Specifically, for a uniformly drawn root node X, each edge in the tree acts as a k-ary symmetric channel. Thus, labelling the nodes in the trees according to the above distribution and freezing the giant to the correct labels leads to a typical clustering with high probability. The resulting bound matches the giant component bound at $b = 0$ but is unlikely to scale properly for small b.

To improve on this, we next take into account the vertices in the giant that belong to planted trees and follow the same program as above, except that the root node (in the giant) is now frozen to the correct label rather than being uniformly drawn. This gives a bound claimed tight at the first-order approximation when b is small. Finally, we also take into account vertices that are not saturated, i.e., whose neighbors do not cover all communities and who can thus be swapped without affecting typicality. The latter allows us to cross at $k = 4$.

The technical estimates to obtain the desired bound on the typical set size are as follows. Let T be the isolated number and M be the total number of edges that belong to trees (both isolated and planted in the giant). Let Z be the number of vertices than are not saturated (as defined above). Similarly to the Erdős-Rényi case [38], we show the following concentration results taking place in probability:

$$T/n \xrightarrow{(p)} \frac{\tau}{d}\Big(1 - \frac{\tau}{2}\Big), \tag{4.3}$$

$$M/n \xrightarrow{(p)} \frac{\tau^2}{2d}, \tag{4.4}$$

$$Z/n \xrightarrow{(p)} e^{-a/k}(1 - (1 - e^{-b/k})^{k-1}), \tag{4.5}$$

where τ is as in Theorem 2.10. Using entropic bounds to estimate how many typical assignments can be obtained from the above three concentration results, we obtain our bound on the typical set size that implies Theorem 2.10.

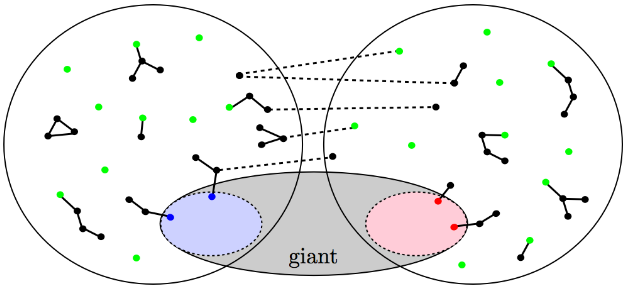

FIGURE 4.1. Illustration of the topology of SBM(n,k,a,b) for $k = 2$. A giant component covering the two communities takes place when $d = \frac{a+(k-1)b}{k} > 1$; a linear fraction of vertices belong to isolated trees (including isolated vertices), and a linear fraction of vertices in the giant are on planted trees. The following is used to estimate the size of the typical set in Theorem 6.28. For isolated trees, sample a bit uniformly at random for a vertex (green vertices) and propagate the bit according to the symmetric channel with flip probability $b/(a+(k-1)b)$ (plain edges do not flip whereas dashed edges flip). For planted trees, do the same but freeze the root bit to its true value.

5 Conclusions and Open Problems

5.1 Impossibility Statements

Further conjectures were made in [34] concerning impossibility statements:

CONJECTURE 2. *Under the same model as in Conjecture* 1*, irrespective of the value of* k*, if* SNR < 1*, it is impossible to detect communities in polynomial time.*

This is proved for $k = 2$ in [61], in which a reduction is used on the reconstruction problem on trees [63]. The same program as in [61] is likely to extend to $k = 3$, at least for large degrees, as in this case it is shown in [72] that it is impossible to detect below the KS threshold for the reconstruction problem on trees.

For $k \geq 4$, the problem is much harder given the results in the current paper (i.e., the fact that ruling out all methods does not suffice due to the gap emerging). While [34] provides evidence towards this conjecture, proving formally such a conjecture would require computational lower bounds that seem currently out of reach. Similar statements apply to related models such as planted clique or planted community [35, 46, 58] problems. To back up the current "evidence" on the impossibility of detecting efficiently below the KS threshold, one may rely on further reductions as is done in [15, 27] or subclasses of algorithms as is done in [39] with statistical query algorithms. Finding the exact expression of the IT threshold at all SNRs also seems to be an interesting problem.

5.2 Accuracy

As mentioned in Section 1.1, solving the detection problem above the KS threshold also gives a path to achieving optimal accuracy in the communities reconstruction. Since ABP solves detection as soon as SNR > 1, it can then be enhanced to achieve better accuracy using full BP, using our graph-splitting approach from [4]. This is relatively straightforward for two communities. For more than two communities, this first requires converting the two sets that are correlated with their communities into a nontrivial assignment of a belief to each vertex.

There are several ways one could do that, such as by making the following modifications to the algorithm. First, one would change the definition of Y so that Y is a $2|E(G)| \times m$ matrix such that for all $(v', v) \in E(G)$, $Y_{(v',v),t} = y^{(t)}_{(v',v)}$. Then, rather than dividing G's vertices into those that have positive and negative values of y', divide its directed edges into those that have positive and negative values of y'. Then use the frequencies of vertices with given numbers of edges with positive or negative values of y', and edge densities between these vertices, to assign a probability distribution for the communities of the starting vertices of edges with given signs of y'. Finally, use these as the starting probabilities for a belief propagation algorithm of depth $\ln(n)/3\ln(\lambda_1)$.

For the general SBM, this requires some conditions, and we make the following conjecture.

CONJECTURE 3. *Let $k \in \mathbb{Z}_+$, $p \in (0,1)^k$, Q be a $k \times k$ symmetric matrix with nonnegative entries, and G be drawn under* SBM$(n, p, Q/n)$. *Furthermore, assume that for any two communities i and j, there exists an eigenvector w of PQ with eigenvalue λ_2 such that $w_i \neq w_j$. Then there exist $c, r \in \mathbb{Z}^+$ and $m : \mathbb{Z}^+ \to \mathbb{Z}^+$ such that with the above modifications,* ABP$(G, m(n), r, c, (\lambda_1, \ldots, \lambda_h))$ *classifies G's vertices with an accuracy that is assymptotically at least as good as any other polynomial-time algorithm.*

If there exist communities i and j such that $w_i = w_j$ whenever w is an eigenvector of PQ with eigenvalue λ_2, then the original use of ABP will not distinguish between these communities. We believe that one could classify vertices with optimal accuracy on such a graph by using the beliefs resulting from this algorithm as a starting point for another layer of ABP and possibly going through several sucessive layers.

5.3 Extensions

Many variants of the SBM can be studied, such as the labeled block model [43], the censored block model [1,3,29,69], the degree-corrected block model [52], overlapping block models [41], and more. While many of the fundamental challenges seem to be captured by the SBM, these represent important extensions for applications. Another important extension would be to tackle sublinear size communities, which is likely to raise new challenges (e.g., the planted clique problem).

6 Proofs

6.1 Achieving the KS Threshold

The proof works for the general model SBM$(n, p, Q/n)$. Recall that $P = \operatorname{diag}(p)$, and $\lambda_1, \dots, \lambda_h$ are the distinct eigenvalues of PQ in order of decreasing magnitude.

DEFINITION 6.1. For any r and m, an *r-nonbacktracking walk of length m* is a sequence of vertices $v_0, \dots, v_m$ such that $v_i \neq v_j$ whenever $|i - j| \leq r$ and v_i is adjacent to v_{i+1} for all i.

DEFINITION 6.2. Given $r, m > 0$ and vertices v and v', let $W_{m[r]}(v, v')$ be the number of r-nonbacktracking walks of length m from v to v'.

DEFINITION 6.3. Given $r, m > 0$, graph G, an assignment of a real number x_v to every vertex v, a multiset of real numbers S, and a vertex v, let

$$W_{m/S[r]}(x, v) = \sum_{T \subseteq S} \prod_{y \in T} (-y) \sum_{v' \in G} x_{v'} W_{m-|T|[r]}(v', v). \tag{6.1}$$

In other words, $W_{m/\varnothing[r]}(x, v)$ is the sum over all r-nonbacktracking walks of length m ending at v of the values of x at their initial vertices, and for $S \neq \varnothing$ and $y \in S$, we have that

$$W_{m/S[r]}(x, v) = W_{m/(S\setminus\{y\})[r]}(x, v) - y \cdot W_{m-1/(S\setminus\{y\})[r]}(x, v).$$

Note that the ABP algorithm sets $Y_{v,t}$ equal to $W_{t/\varnothing[r]}(x, v)$ for all $0 < t \leq m$ and $v \in G$ in step 2(b). Then in step 2(c), it sets $y_v^{(m)}$ equal to $W_{m/S[r]}(x, v)$, where S is the multiset containing $\lceil \frac{m-r-(2r+1)s'}{l} \rceil$ copies of $\lambda_{s'}$ for every $s' < s$.

Further intuition on (6.1) will be provided with Definition 6.5. The plan is to select the x_v independently according to a normal distribution, and then compute $W_{m/S[r]}(x, v)$ for appropriate S, m, and r. The assignment of x will inevitably have slightly different average values in different communities, and under the right conditions, these differences will be amplified to the point of allowing differentiation between communities with asymptotically nonzero advantage.

PROOF OF THEOREM 2.7. When ABP splits Γ off of G, the remaining graph is still drawn from the SBM, albeit with connectivity $(1-\gamma)Q$. The formula for γ ensures that if $\lambda_2^2 > \lambda_1$, then $((1-\gamma)\lambda_2)^2 > ((1-\gamma)\lambda_1)$. Now, let $m'' = \lfloor \sqrt{\log\log n} \rfloor + 1$. For each $v \in G$ and $t > 0$, let $N_t(v)$ be the set of all vertices t edges away from v, $N_{\leq t}(v)$ be the subgraph of G induced by the vertices within t edges of v, and $T_v = \{v' : \exists v'' \in N_{m''}(v) \mid (v', v'') \in \Gamma\}$. Unless the original graph has a cycle in $N_{\leq m''}(v) \cup T_v$, we have that $y''_v = \sum_{v' \in T_v} y_{v'}^{(m)}$.

Now, let w be an eigenvector of PQ with eigenvalue λ_i and magnitude 1. Also, for every $v \in G$ and $t \geq 0$, let $W_t(v) = \sum_{v' \in N_t(v)} w_{\sigma_{v'}} / p_{\sigma_{v'}}$. Given a fixed value of $N_{\leq t}(v)$, $v' \in N_t(v)$, a fixed value of $\sigma_{v'}$, and a community j, the expected number of vertices not in $N_{\le t}(v)$ that are in community j and adjacent to v' is

$$\left(1-\frac{|N_{\le t}(v)|}{n}\right)(1-\gamma)e_j\cdot PQe_{\sigma_{v'}}.$$

So, we can show by induction on t that for any $t\ge 0$, $n'>0$, and $z\in\mathbb{R}$, we have

$$E[W_{m''}(v)|W_{m''-t}(v)=z,|N_{m''-t}(v)|=n']=\\ (1-\gamma)^t\lambda_i^t z+O\left(\frac{((1-\gamma)\lambda_1)^{2t}(n')^2}{n}\right).$$

Similarly, for any $t\ge 0$, $0<n'\le \ln^{2m''-2t}(n)$, and $z\in\mathbb{R}$, we have

$$\mathrm{Var}\big[W_{m''}(v)\mid W_{m''-t}(v)=z,|N_{m''-t}(v)|=n'\big]=\\ O\left(\sum_{t'=1}^{t} n'(1-\gamma)^{t+t'}\lambda_1^{t-t'}\lambda_i^{2t'}\right).$$

Now, given nonintersecting graphs N' and N'' with $|N'|,|N''|=O(\ln^{2m''}(n))$, $v\in N'$, and $v'\in N''$, the event that the subgraph of G induced by $V(N')$ is N' is independent of the event that the subgraph of G induced by $V(N'')$ is N''. Also, there are no edges between these two graphs with probability $1-O(|N'|\cdot |N''|/n)$ regardless of what form these graphs take. Furthermore, for any $v''\notin N'\cup N''$, regardless of the value of $G\backslash\{v''\}$ there are no edges from v'' to $V(N')$ with probability $1-\lambda_1|N'|/n+O(|N'|^2/n^2)$, there are no edges from v'' to $V(N'')$ with probability $1-\lambda_1|N''|/n+O(|N''|^2/n^2)$, and there are no edges from v'' to either with probability $1-\lambda_1(|N'|+|N''|)/n+O((|N'|+|N''|)^2/n^2)$. So, we have that

$$\begin{aligned}&\big|P[N_{m''}(v)=N',N_{m''}(v')=N'']\\ &\quad -P[N_{m''}(v)=N']\cdot P[N_{m''}(v')=N'']\big|\\ &\qquad =O(|N'|\cdot|N''|/n)P[N_{m''}(v)=N',N_{m''}(v')=N'']\end{aligned}$$

Furthermore, the probability that $N_{m''}(v)$ and $N_{m''}(v')$ intersect is $O(\lambda_1^{2m''}/n)$. Also, $|W_{m''}(v)|\le n\cdot\max_i|w_i/p_i|$ for all v, and with probability $1-o(n^{-20})$, every vertex in G has degree less than $\ln^2(n)$, so $W_{m''}(v)=O(\ln^{2m''}(n))$ for all v. So, for $v\ne v'$, the correlation between $W_{m''}(v)$ and $W_{m''}(v')$ is $o(1/\sqrt{n})$, and the correlation between $W^2_{m''}(v)$ and $W^2_{m''}(v')$ is also $o(1/\sqrt{n})$. Therefore, with probability $1-o(1)$, the average value of $W_{m''}(v)$ over all v in community j is $(1-\gamma)^{m''}\lambda_i^{m''}(w_j/p_j+o(1))$ for all j. Also, there exists constant δ such that $\sum_{v\in G}(W_{m''}(v))^2\le n\delta\sum_{t=1}^{m''}(1-\gamma)^{m''+t}\lambda_1^{m''-t}\lambda_i^{2t}$ with probability $1-o(1)$.

On another note, for each v, we have that $y_v^{(m)} = W_{m/\{c_i\}}(x, v)$, so by Lemma 6.25,

$$\operatorname{Var}[W_{m/\{c_i\}}(x, v)] = O\biggl(\prod_{j=1}^{m} ((1-\gamma)\lambda_s - c_j)^2\biggr),$$

and the empirical variance of $\{y_v^{(m)}\}$ is $O(\prod_{j=1}^{m}((1-\gamma)\lambda_s - c_j)^2)$ with probability $1 - o(1)$. Now, let $\overline{y}$ be the vector such that $\overline{y}_i = \sum_{v\in\Omega_i} y_v^{(m)}/n$ for all i. By Lemma 6.23, we know that with probability $1-o(1)$, the component of $\overline{y}$ on PQ's eigenspace of eigenvalue λ_s has magnitude $\Omega(\frac{1}{\sqrt{n}}\prod|(1-\gamma)\lambda_s - c_j|)$, while all of its components on PQ's other eigenspaces have magnitudes of

$$O\biggl(\frac{1}{\log(n)\sqrt{n}}\prod|(1-\gamma)\lambda_s - c_j|\biggr).$$

Now, let $y_v''' = \sum_{v'\in T_v} y_v^{(m)}$ for all v, and recall that $y_v'' = y_v'''$ unless the original graph has a cycle in $N_{\le m''}(v) \cup T_v$. Observe that

$$\begin{aligned}\sum_{v\in G} w_{\sigma_v} y_v'''/p_{\sigma_v} &= \sum_{\substack{v,v',v''\in G: v'\in N_{m''}(v),\\ (v',v'')\in\Gamma}} w_{\sigma_v} y_{v''}^{(m)}/p_{\sigma_v}\\ &= \sum_{(v',v'')\in\Gamma} W_{m''}(v')\cdot y_{v''}^{(m)}.\end{aligned}$$

For each $v, v' \in G$ without an edge between them,

$$P[(v,v')\in\Gamma \mid \sigma_v = j, \sigma_{v'} = j'] = (1 + O(1/n))\gamma Q_{j,j'}/n.$$

So, there exists $\delta_w > 0$ such that for a fixed G, σ, and x, with probability $1 - o(1)$, it will be the case that

$$E\biggl[\sum_{v\in G} w_{\sigma_v} y_v'''/p_{\sigma_v}\biggr] = \gamma(1-\gamma)^{m''}\lambda_i^{m''+1}(w\cdot P^{-1}\overline{y} + o(1/\sqrt{n}))n$$

and

$$\begin{aligned}\operatorname{Var}\biggl[\sum_{v\in G} w_{\sigma_v} y_v'''/p_{\sigma_v}\biggr] &= (\delta_w/n)\sum_{v',v''}\bigl(W_{m''}(v')\cdot y_{v''}^{(m)}\bigr)^2\\ &\le n\delta_w^2\sum_{t=1}^{m''}(1-\gamma)^{m''+t}\lambda_1^{m''-t}\lambda_i^{2t}\cdot\prod_{j=1}^{m}((1-\gamma)\lambda_s - c_j)^2\\ &= o\Biggl(\biggl(\gamma(1-\gamma)^{m''}\lambda_s^{m''+1}\cdot\sqrt{n}\prod_{j=1}^{m}((1-\gamma)\lambda_s - c_j)^2\biggr)^2\Biggr).\end{aligned}$$

This means that with probability $1 - o(1)$, we have that

$$\operatorname{Var}\biggl[\sum_{v\in G} w_{\sigma_v} y_v'''/p_{\sigma_v}\biggr] = o\bigl(\gamma^2(1-\gamma)^{2m''}\lambda_s^{2m''+2}\|\overline{y}\|_2^2 n^2\bigr).$$

In particular, if we choose an orthonormal eigenbasis for PQ, $w_1, \dots, w_k$, then with probability $1 - o(1)$ this will hold for all w_i in the basis, which implies that with probability $1 - o(1)$, we have that

$$\sum_{v \in \Omega_j} y_v'''/n = \gamma(1-\gamma)^{m''}\big((PQ)^{m''+1}\overline{y}\big)_j + o\big(\gamma(1-\gamma)^{m''}\lambda_s^{m''+1}\|\overline{y}\|_2\big)$$

for all $j \in [k]$. Also, for fixed G and x and with probability $1 - o(1)$ we have

$$\begin{aligned}\sum_{v \in G}(y_v''')^2 &= \sum_{v,v_2,v_3,v_2',v_3' \in G: v_2,v_2' \in N_{m''}(v),(v_2,v_3)\in\Gamma,(v_2',v_3')\in\Gamma} y_{v_3}^{(m)} \cdot y_{v_3'}^{(m)} \\ &= (1+o(1))\sum_{v \in G}\Big(\sum_{v_2 \in N_{m''}(v)} \gamma(Q\overline{y})_{\sigma_{v_2}}\Big)^2 \\ &\quad + (1+o(1))\gamma(1-\gamma)^{m''}\lambda_1^{m''+1}\sum_{v \in G}\big(y_v^{(m)}\big)^2 \\ &\le (1+o(1))n\sum_{t=1}^{m''}\gamma^2(1-\gamma)^{m''+t}\lambda_1^{m''-t}\lambda_s^{2t+2}\|\overline{y}\|_2^2/(\min p_i) \\ &\quad + O\big(\gamma(1-\gamma)^{m''}\lambda_1^{m''+1}\|\overline{y}\|_2^2 n\big) \\ &= (1+o(1))n \cdot \frac{(1-\gamma)\lambda_s^2}{\min p_i[(1-\gamma)\lambda_s^2 - \lambda_1]}\gamma^2(1-\gamma)^{2m''}\lambda_s^{2m''+2}\|\overline{y}\|_2^2.\end{aligned}$$

With probability $1 - o(1)$, there are only $O(\ln(n))$ vertices $v \in G$ such that $y_v'' \neq y_v'''$. With probability $1 - o(1)$, no such vertex has more than $\ln^{2m''+2}(n)$ vertices within $m''+1$ edges of it, and no such vertex has more than one cycle within $m''+1$ edges of it. Assuming this holds, these differences are small enough that the results above still hold if we substitute y_v'' for y_v'''. Among other things, this means that we can choose σ_1 and σ_2 such that the average value of y_v'' for $v \in \Omega_{\sigma_1}$ differs from the average value of y_v'' for $v \in \Omega_{\sigma_2}$ by at least $(1-o(1))\gamma(1-\gamma)^{m''}\lambda_s^{m''+1}\|\overline{y}\|_2$. Now, let a be the difference between the average value of y_v'' for $v \in \Omega_{\sigma_1}$ and the average value of y_v'' for $v \in \Omega_{\sigma_2}$. Also, let b be the average value of $(y_v'')^2$. There exists a constant δ' such that $b \le \delta' a^2$ with probability $1 - o(1)$. The average value of $\frac{1}{2} + y_v''/(C\sqrt{\sum(y_{v'}'')^2/n})$ for $v \in \Omega_{\sigma_1}$ differs from its average value for $v \in \Omega_{\sigma_2}$ by $\frac{a}{C\sqrt{b}}$. For any v,

$$\min\Bigg(|y_v''/(C\sqrt{\textstyle\sum(y_{v'}'')^2/n})| - \frac{1}{2}, 0\Bigg) \le 2(y_v''/(C\sqrt{b}))^2$$

So, the average value of $\min(|y_v''/(C\sqrt{\sum(y_{v'}'')^2/n})| - \frac{1}{2}, 0)$ for v in any community is at most $2/(C^2 \min p_i)$. So the probability that a vertex from community σ_1 is assigned to group 2 differs from the probability that a vertex from community σ_2

is assigned to group 2 by at least $\frac{a}{C\sqrt{b}} - 4/(C^2 \min p_i)$. For a sufficiently large constant C and small constant $\epsilon > 0$, this will be greater than ϵ with probability $1 - o(1)$, as desired.

ABP initializes in $O(n)$ time. Computing the $y^{(t)}$ takes $O(n \log n)$ expected time. The eigenvalue compensation step can be carried out in $O(n \log n)$ time if

$$\Big(\prod_{s'<s} M_{s'}^{\lceil \frac{m-r-(2r+1)s'}{l} \rceil}\Big) e_m$$

is computed first and then Y is multiplied by it, and computing y'' takes $o(n \log n)$ time. The assignment step also takes $O(n)$ time, so the entire algorithm can be run in $O(n \log n)$ time. □

Preliminaries

Our first step in proving the lemmas used above is to reexpress $W_{m/S[r]}(x, v)$ as a sum over r-nonbacktracking walks of an expression that we will eventually be able to bound. Then we will establish some properties of r-nonbacktracking walks that we will need.

DEFINITION 6.4. For any $r \geq 1$ and series of vertices $v_0, \dots, v_m$, let $W_r((v_0, \dots, v_m))$ be 1 if $v_0, \dots, v_m$ is an r-nonbacktracking walk and 0 otherwise.

DEFINITION 6.5. For any $r \geq 1$, series of vertices $v_0, \dots, v_m$, and series of real numbers $c_0, \dots, c_m$, let $W_{(c_0,\dots,c_m)[r]}((v_0, \dots, v_m))$ be the sum, over all subseries $i_0, \dots, i_{m'}$ of $0, \dots, m$, of

$$\left(\prod_{i \notin (i_0,\dots,i_{m'})} (-c_i/n)\right) \cdot W_r((v_{i_0}, v_{i_1}, \dots, v_{i_{m'}})).$$

Note that for any $c_0, \dots, c_m$ consisting of the elements of S and 0's with $c_0 = c_m = 0$,

$$W_{m/S[r]}(x, v) = \sum_{v_0,\dots,v_m \in G : v_m = v} [x_{v_0} \cdot W_{(c_0,\dots,c_m)[r]}((v_0, \dots, v_m))].$$

DEFINITION 6.6. Given a series of vertices $v_0, \dots, v_m$, its walk graph is the graph consisting of every vertex in the series with an edge between each pair of vertices that is adjacent in the series. Note that the walk graph has a number of vertices equal to the number of distinct vertices in $v_0, \dots, v_m$, a number of edges equal to the cardinality of $\{\{v_i, v_{i+1}\} : 0 \leq i < m\}$, and every vertex in it other than v_0 or v_m must have degree at least 2.

LEMMA 6.7. *Consider the complete graph K_n. For any $m, b, w, e \geq 0$, there are at most $2n^{m-w+1}(m+1)^{4(w-e)}2^{2b(w-e)}$ walks of length m in K_n that have $m + 1 - w$ distinct vertices, $m - e$ distinct edges, no vertex repeated more than b times, and no edge repeated twice in a row.*

PROOF. Choose such a walk, and let H be its walk graph. Also, let the distinct vertices in the walk be $v_1, \dots, v_{m+1-w}$, in order of first appearance. There are at most n^{m-w+1} possible choices of which vertices these are. For any i such that v_i is not adjacent to v_{i+1} in the walk, the first edge in the walk after v_i and the first edge leading to v_{i+1} must both have not appeared previously in the walk. Otherwise, the edge between v_i and v_{i+1} must not have appeared previously. So, there are at most $m-e-(m-w) = w-e$ indices i such that v_{i+1} is not adjacent to v_i. There are at most $(m+1)^{w-e}$ choices of which indices have this property, and $m+1$ choices of which vertex the first edge leading to each such v_{i+1} has on its other side. That accounts for $m-w$ of H's edges, so there are at most $(m+1)^{2(w-e)}$ possibilities of what the others are. Thus, there are at most $n^{m-w+1}(m+1)^{4(w-e)}$ possible graphs H.

Now, let $v'_0, \dots, v'_m$ be the complete walk in question. Given the values of v'_i and v'_{i-1}, the only possible values of v'_{i+1} are the vertices that are adjacent to v'_i other than v'_{i-1}. Furthermore, since no vertex appears in the walk more than b times, that means that the number of possible walks for a given H is at most

$$2 \prod_{v \in H:\text{degree}(v)>1} (\text{degree}(v)-1)^b \leq 2 \prod_{v \in H:\text{degree}(v)>1} 2^{b(\text{degree}(v)-2)} \leq 2^{1+2b(w-e)}.$$

□

DEFINITION 6.8. Given a sequence of vertices $v_0, \dots, v_m$, a *fresh segment* is a consecutive subsequence $v_a, v_{a+1}, \dots, v_b$ such that no interior vertex of the segment has degree greater than 2 in the walk graph of $v_0, \dots, v_m$, no interior vertex of the segment is equal to v_m, and the walk $v_0, \dots, v_a$ does not traverse any of the segment's edges. Call the segment nonrepeated if none of its edges occur in the walk $v_b, \dots, v_m$ and repeated otherwise.

LEMMA 6.9 (Walk decomposition lemma). *Let $v_0, \dots, v_m$ define a walk in K_n that has $m+1-w$ distinct vertices and $m-e$ distinct edges. There exists a set of at most $3(w-e)+1$ fresh segments of the walk that are disjoint except at end vertices, and such that every edge in the walk is in one of these segments. Also, every end vertex of one of these segments is either v_0, v_m, or a vertex that is repeated in the walk.*

PROOF. Let G' be the walk graph of $v_1, \dots, v_m$. Let S consist of v_1, v_m, and all vertices in G' of degree greater than 2. The total degree of all vertices in S is at most $6(w-e)+2$, so G' consists of these vertices and at most $3(w-e)+1$ paths between them that do not contain any vertex from S. Once the walk goes onto one of these paths, the facts that it is nonbacktracking, v_m is not in the path, and no vertex on the path has degree greater than 2 forces it to continue to the end of the path. The walk must traverse each of these paths for the first time at some point. Therefore, the set of series of vertices corresponding to the first traversals of these paths is the desired set of fresh segments. □

DEFINITION 6.10. Given a sequence of vertices $v_0, \dots, v_m$, let its *standard decomposition* be the collection of fresh segments generated by the above construction for $v_0, \dots, v_m$.

DEFINITION 6.11. Given a collection of fresh segments, let its *measures* be d, d', t, t', where d is the number of nonrepeated fresh segments in the collection, t is the sum of their lengths, d' is the number of repeated fresh segments in the collection, and t' is the sum of their lengths. Also, let the *measures of a series of vertices* denote the measures of its standard decomposition.

The Shard Decomposition

The next step in the proof is to establish an upper bound on

$$|E[W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))]|$$

when $(c_0,\dots,c_m), (v_0,\dots,v_m)$ satisfy some special properties. Then we will show that $W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))$ can always be expressed as a linear combination of shards, expressions of the form $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))$ that have the aforementioned properties. After that, we will show that if $(c_0,\dots,c_m)$ is chosen correctly, the magnitudes of the expected values of many of the shards of $W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))$ must be fairly small.

LEMMA 6.12. *Let $r \geq 1$ and $c_0,\dots,c_m$ be a series of real numbers with $c_0 = c_m = 0$. Also let $v_0,\dots,v_m$ be a series of vertices with standard decomposition $v_{a_1},\dots,v_{b_1}$, $v_{a_2},\dots,v_{b_2},\dots,v_{a_d},\dots,v_{b_d}$, ordered so that $b_i \leq a_{i+1}$ for each i. Assume that for any i, j such that v_i occurs elsewhere in the series and $|i-j| \leq r$, $c_j = 0$. Then,*

$$|\mathbb{E}[W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))]| \leq \\ n^{-\sum_{i=1}^{d} b_i - a_i} (\min p_i)^{-d} \prod_{i=1}^{d} \sum_{j=1}^{k} \prod_{i'=a_i}^{b_i-1} |\lambda_j - c_{i'}|.$$

PROOF. For any i such that $c_i \neq 0$, i is at least r away from every element of the series that is repeated. So, deleting v_i for all i in any subset of $\{i : c_i \neq 0\}$ has no effect on whether $v_1,\dots,v_m$ is r-nonbacktracking. If $v_1,\dots v_m$ is not r-nonbacktracking, then the expected value of $W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))$ is 0, and the conclusion holds. Now, consider the case where $v_1,\dots,v_m$ is r-nonbacktracking. Since any subsequence of $v_0,\dots,v_m$ resulting from deleting v_i for which $c_i \neq 0$ is also r-nonbacktracking, the restriction that a vertex cannot repeat within r steps is irrelevant and

$$W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m)) = W_{(c_0,\dots,c_m)[0]}((v_0,\dots,v_m)).$$

Also, v_{a_i} and v_{b_i} are all repeated vertices except possibly v_0 and v_m. So, $c_{a_i} = 0$ and $c_{b_i} = 0$ for all i. That implies that

$$\begin{aligned} &W_{(c_0,\dots,c_m)[0]}((v_0,\dots,v_m)) \\ &\quad = W_{(c_0,\dots,c_{a_1})[0]}((v_0,\dots,v_{a_1})) \cdot W_{(c_{a_1},\dots,c_{b_1})[0]}((v_{a_1},\dots,v_{b_1})) \\ &\qquad \cdot W_{(c_{b_1},\dots,c_{a_2})[0]}((v_{b_1},\dots,v_{a_2})) \cdot W_{(c_{a_2},\dots,c_{b_2})[0]}((v_{a_2},\dots,v_{b_2})) \\ &\qquad \cdot \dots \cdot W_{(c_{b_d},\dots,c_m)[0]}((v_{b_d},\dots,v_m)). \end{aligned}$$

If v_i is not part of one of the fresh segments, then v_i is a repetition of a vertex that is in one of them, so $c_i = 0$. Thus,

$$\begin{aligned} W_{(c_0,\dots,c_{a_1})[0]}((v_0,\dots,v_{a_1})) \cdot W_{(c_{b_1},\dots,c_{a_2})[0]}((v_{b_1},\dots,v_{a_2})) \\ \cdot \dots \cdot W_{(c_{b_d},\dots,c_m)[0]}((v_{b_d},\dots,v_m)) \end{aligned}$$

is either 0 or 1. If it is 0, then there is some v_i, v_{i+1} that are not part of any of the fresh segments and do not have an edge between them. By the previous lemma, there must exist $1 \leq i \leq d$ and $a_i \leq j \leq b_i - 1$ such that $\{v_i, v_{i+1}\} = \{v_j, v_{j+1}\}$, and since the vertices are repeated, $c_j = c_{j+1} = 0$. So, the lack of an edge between them means that $W_{(c_{a_i},\dots,c_{b_i})[0]}((v_{a_i},\dots,v_{b_i})) = 0$. Either way,

$$W_{(c_0,\dots,c_m)[0]}((v_0,\dots,v_m)) = \prod_{i=1}^{d} W_{(c_{a_i},\dots,c_{b_i})[0]}((v_{a_i},\dots,v_{b_i})).$$

The fresh segments in a standard decomposition only intersect at their endpoints, so for a fixed assignment of communities to the endpoints,

$$\begin{aligned} &|\mathbb{E}[W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))]| \\ &\quad = \prod_{i=1}^{d} |\mathbb{E}[W_{(c_{a_i},\dots,c_{b_i})[0]}((v_{a_i},\dots,v_{b_i}))]| \\ &\quad = n^{-\sum b_i - a_i} \prod_{i=1}^{d} \left| e_{\sigma_{v_{a_i}}} P^{-1} \prod_{i'=a_i}^{b_i-1} (PQ - c_{i'} I) e_{\sigma_{v_{b_i}}} \right| \\ &\quad \leq n^{-\sum b_i - a_i} (\min p_i)^{-d} \prod_{i=1}^{d} \sum_{j=1}^{k} \prod_{i'=a_i}^{b_i-1} |\lambda_j - c_{i'}|. \qquad \square \end{aligned}$$

Of course, for general $c_0, \dots, c_m$ and $v_0, \dots, v_m$, the condition that $c_i = 0$ whenever there is a repeated vertex v_j with $|i - j| \leq r$ may not be satisfied. We can deal with that by using the fact that for arbitrary $c_0, \dots, c_m$, $v_0, \dots, v_m$, r, and i,

$$
\begin{aligned}
&W_{(c_0,\dots,c_{i-1},c_i,c_{i+1},\dots,c_m)[r]}((v_0,\dots,v_{i-1},v_i,v_{i+1},\dots,v_m)) \\
&\quad = W_{(c_0,\dots,c_{i-1},0,c_{i+1},\dots,c_m)[r]}((v_0,\dots,v_{i-1},v_i,v_{i+1},\dots,v_m)) \\
&\qquad - \frac{c_i}{n} W_{(c_0,\dots,c_{i-1},c_{i+1},\dots,c_m)[r]}((v_0,\dots,v_{i-1},v_{i+1},\dots,v_m)).
\end{aligned}
$$

So, for any $c_0,\dots,c_m$ with $c_0 = c_m = 0$, $v_0,\dots,v_m$, and r, we can express $W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))$ as a linear combination of expressions that the above lemma applies to by means of the following algorithm.

Path-sum-conversion($W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))$):

(1) If there exist i and j such that $v_i = v_j$, $c_i \neq 0$, and $c_j = 0$, return

$$
\begin{aligned}
&\text{path-sum-conversion}(W_{(c_0,\dots,c_{i-1},0,c_{i+1},\dots,c_m)[r]}((v_0,\dots,v_{i-1},v_i,v_{i+1},\dots,v_m))) \\
&- \frac{c_i}{n}\text{path-sum-conversion}(W_{(c_0,\dots,c_{i-1},c_{i+1},\dots,c_m)[r]}((v_0,\dots,v_{i-1},v_{i+1},\dots,v_m)))
\end{aligned}
$$

(2) Otherwise, if there exist $j \neq j'$ such that $v_j = v_{j'}$, and i such that $0 < |i-j| \leq r$ and $c_i \neq 0$, return

$$
\begin{aligned}
&\text{path-sum-conversion}(W_{(c_0,\dots,c_{i-1},0,c_{i+1},\dots,c_m)[r]}((v_0,\dots,v_{i-1},v_i,v_{i+1},\dots,v_m))) \\
&- \frac{c_i}{n}\text{path-sum-conversion}(W_{(c_0,\dots,c_{i-1},c_{i+1},\dots,c_m)[r]}((v_0,\dots,v_{i-1},v_{i+1},\dots,v_m)))
\end{aligned}
$$

(3) Otherwise, if there exist $i \neq j$ such that $v_i = v_j$, $c_i \neq 0$, and $c_j \neq 0$, return

$$
\begin{aligned}
&\text{path-sum-conversion}(W_{(c_0,\dots,c_{i-1},0,c_{i+1},\dots,c_m)[r]}((v_0,\dots,v_{i-1},v_i,v_{i+1},\dots,v_m))) \\
&- \frac{c_i}{n}\text{path-sum-conversion}(W_{(c_0,\dots,c_{i-1},c_{i+1},\dots,c_m)[r]}((v_0,\dots,v_{i-1},v_{i+1},\dots,v_m)))
\end{aligned}
$$

(4) Otherwise, if there exist $i \neq j$ such that $|i-j| \leq r$, and $v_i = v_j$, return 0

(5) Otherwise, return $W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))$

If there is more than one i satisfying the conditions of the case under consideration, the algorithm should choose one according to some rule, such as always using the smallest. Also, note that this algorithm exists as a proof technique allowing us to replace any expression of the form $W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m)))$ with a sum of expressions of the form $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'})))$ that the previous lemma applies to. We would never actually run it.

LEMMA 6.13. Path-sum-conversion($W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))$) *always terminates. Furthermore, if there are no i, j such that $c_i \neq 0$, $c_j \neq 0$, and $|i-j| \leq r$, then for any $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))$ that appears in the expression it outputs and any v_i that was deleted in going from $W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m)))$ to $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))$, at least one of the following holds. There exists i' such that $v'_{i'} = v_i$, or there exists j such that $|i-j| \leq r$, v_j was not deleted in going from $W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m)))$ to $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))$, and v_j appears more than once in the list $(v'_0,\dots,v'_{m'})$. Also, if $c_0 = c_m = 0$,* *then for any* $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))$ *that appears in the expression it outputs,* $c'_0 = c'_{m'} = 0$.

PROOF. First, note that if $c_i = 0$, then v_i will never be deleted. The theorem follows from a combination of induction on the number of nonzero elements in $(c_0,\dots,c_m)$ and case analysis of the circumstances under which v_i can be deleted. □

DEFINITION 6.14. $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))$ is a *level x shard* of

$$W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))$$

if $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))$ is one of the expressions in the sum resulting from Path-sum-conversion$(W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m)))$ and exactly x of the vertices that were deleted in the process of converting

$$W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m)) \quad \text{to} \quad W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))$$

are repetitions of vertices in $(v'_0,\dots,v'_{m'})$. For a given $W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))$, Shar$(W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m)))$ denotes the set of all of its shards, and

$$\mathrm{Shar}_x\big(W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))\big)$$

denotes the set of its level x shards.

DEFINITION 6.15. Let an *eigenvalue approximation* be an ordered tuple $\lambda'_1,\dots,\lambda'_{s-1}$ such that $|\lambda_i - \lambda'_i| \le |\lambda_i - \lambda'_{i'}|$ for all i and i'. Call such a tuple Λ*-bounded* if $|\lambda'_i| \le \Lambda$ and $|\lambda_i| \le \Lambda$ for all i. Also, let the error of such an approximation be $\max_i |\lambda_i - \lambda'_i|$.

DEFINITION 6.16. Given integers r, l, a tuple $\lambda'_1,\dots,\lambda'_{s-1}$, and a sequence $c_0,\dots,c_m$, let the sequence be an $l[r]$*-cycle* of $(\lambda'_1,\dots,\lambda'_{s-1})$ if and only if $(2r+1)(s-1) \le l$, $c_i = \lambda'_j$ for all i and j such that $i < m - r$ and the remainder of i when divided by l is $(2r+1)j$, and $c_i = 0$ whenever there is no j such that the above holds. Note that the first and last elements of an $l[r]$-cycle are always 0, and that if $(2r+1)(s-1) \le l$, there exists a unique $l[r]$-cycle of length m for every m.

LEMMA 6.17. *Let* $\lambda'_1,\dots,\lambda'_{s-1}$ *be a* Λ*-bounded eigenvalue approximation and* $c_0,\dots,c_m$ *be an* $l[r]$*-cycle of this approximation for some* $l, r > 0$. *Then, let* $v_0,\dots,v_m$ *be a series of vertices, and let* $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))$ *be a level* x *shard of* $W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m)))$. *Finally, let* d, d', t, t' *be the measures of*

$(v'_0, \dots, v'_{m'})$ and $t'' = \lceil \frac{t-d(l+2r+1)}{l} \rceil$. Then

$$\begin{aligned}
&|\mathbb{E}(W_{(c'_0,\dots,c'_{m'})[r]}((v'_0, \dots, v'_{m'}))| \\
&\qquad \leq n^{-t-t'} k^{d+d'} (\min p_i)^{-d-d'} \lambda_1^{t'+\max(t-(2x+d)(s-1)-l\cdot\max(t''-2x,0),0)} \\
&\qquad\quad \cdot (\lambda_1 + \Lambda)^{t-\max(t-(2x+d)(s-1)-l\cdot\max(t''-2x,0),0)-l\cdot\max(t''-2x,0)} \\
&\qquad\quad \cdot \left(\max_{1 \leq j \leq k} |\lambda_j|^{l-s+1} \prod_{i=1}^{s-1} |\lambda_j - \lambda'_i| \right)^{\max(t''-2x,0)} .
\end{aligned}$$

PROOF. Let $v'_{a_1}, \dots, v'_{b_1}$, $v'_{a_2}, \dots, v'_{b_2}$, $\dots$, $v'_{a_d}, \dots, v'_{b_d}$ be the nonrepeated fresh segments in the standard decomposition of $(v'_0, \dots, v'_{m'})$, and $v'_{a'_1}, \dots, v'_{b'_1}$, $v'_{a'_2}, \dots, v'_{b'_2}$, $\dots$,$v'_{a'_{d'}}, \dots, v'_{b'_{d'}}$ be the repeated fresh segments in its standard decomposition. For arbitrary $1 \leq i \leq d'$ and $a'_i \leq i' \leq b'_i$, we have that $v'_{i'}$ either is a repeated vertex or is within one step of one. So, $c'_{i'} = 0$. We know from a previous lemma that

$$\begin{aligned}
&|\mathbb{E}[W_{(c'_0,\dots,c'_{m'})[r]}((v'_0, \dots, v'_{m'}))]| \\
&\quad \leq n^{-t-t'} (\min p_i)^{-d-d'} \left(\prod_{i=1}^{d} \sum_{j=1}^{k} \prod_{i'=a_i}^{b_i-1} |\lambda_j - c'_{i'}| \right) \left(\prod_{i=1}^{d'} \sum_{j=1}^{k} \prod_{i'=a'_i}^{b'_i-1} |\lambda_j - c'_{i'}| \right) \\
&\quad \leq n^{-t-t'} k^{d+d'} (\min p_i)^{-d-d'} \lambda_1^{t'} \left(\prod_{i=1}^{d} \max_{1 \leq j \leq k} \prod_{i'=a_i}^{b_i-1} |\lambda_j - c'_{i'}| \right).
\end{aligned}$$

Now, for each $0 \leq i \leq m'$, define $f(i)$ such that $v_{f(i)}$ corresponds to v'_i. Also, for each $0 \leq i \leq m$, define $g(i)$ so that $v'_{g(i)}$ corresponds to v_i, or v_{i+1} if v_i has been deleted. Also, for each $1 \leq i \leq d$, let y_i be the largest integer such that $g(f(a_i + r + 1) + l \cdot y_i) \leq b_i - r$, or 0 if $a_i + 2r \geq b_i$. Note that for any $i' \geq i$, $i' - i \geq g(i') - g(i)$. As a result,

$$\begin{aligned}
\sum_{i=1}^{d} y_i &\geq \frac{1}{l} \sum_{i=1}^{d} g(f(a_i + r + 1) + l \cdot y_i) - g(f(a_i + r + 1)) \\
&\geq \frac{1}{l} \sum_{i=1}^{d} b_i - r - l - (a_i + r + 1) = \frac{t - d(2r + l + 1)}{l},
\end{aligned}$$

because $g(f(a_i + r + 1) + l \cdot (y_i + 1)) > b_i - r$. So, $\sum_{i=1}^{d} y_i \geq t''$.

Next, note that $c'_j = 0$ whenever $a_i \leq j \leq a_i + r$ or $b_i - r \leq j \leq b_i$ for some i. Also, for any $1 \leq i \leq d$ and $1 \leq j \leq y_i$, it is the case that

$$g(f(a_i + r + 1) + l \cdot j) - g(f(a_i + r + 1) + l \cdot (j - 1)) = l$$

unless one of the vertices $v_{f(a_i+r+1)+l\cdot(j-1)}, \dots, v_{f(a_i+r+1)+l\cdot j-1}$ was deleted and

$$\{c'_{g(f(a_i+r+1)+l\cdot(j-1))}, \dots, c'_{g(f(a_i+r+1)+l\cdot j-1)}\}$$

consists of one copy of $\lambda'_{h''}$ for each h'' and $l-(s-1)$ copies of 0 unless one of the vertices $v_{f(a_i+r+1)+l\cdot(j-1)}, \dots, v_{f(a_i+r+1)+l\cdot j-1}$ was deleted or one of $c_{f(a_i+r+1)+l\cdot(j-1)}, \dots, c_{f(a_i+r+1)+l\cdot j-1}$ was set to 0 as a result of the corresponding vertex being within distance r of a vertex that was repeated somewhere else in the walk. The entirety of the block is at least r away from any vertex that is repeated in $(v'_0, \dots v'_{m'})$, so the second case is only possible if the other copy of that vertex was subsequently deleted. So, at most x blocks fall under each of these two cases, which means that there are at least $\max(t''-2x, 0)$ uncorrupted blocks. Each corrupted block has at most one c'_i with a value of $\lambda'_{h''}$ for each h'', and there is at most one c'_j with a value of $\lambda'_{h''}$ between $c'_{g(f(a_i+r+1)+l\cdot y_i)}$ and c'_{b_i} for each i and h''. So,

$$\begin{aligned}
&\prod_{i=1}^{d} \max_{1\le j\le k} \prod_{i'=a_i}^{b_i-1} |\lambda_j - c'_{i'}| \le \\
&\qquad \left(\max_{1\le j\le k} |\lambda_j|^{l-s+1} \prod_{i=1}^{s-1} |\lambda_j - \lambda'_i|\right)^{\max(t''-2x,0)} \\
&\qquad \cdot \lambda_1^{\max(t-(2x+d)(s-1)-l\cdot\max(t''-2x,0),0)} \\
&\qquad \cdot (\lambda_1+\Lambda)^{t-\max(t-(2x+d)(s-1)-l\cdot\max(t''-2x,0),0)-l\cdot\max(t''-2x,0)}. \qquad \square
\end{aligned}$$

LEMMA 6.18. *Let $c_0, \dots, c_m$ be a sequence of real numbers with at most y nonzero elements, and $v'_0, \dots v'_{m'}$ be vertices. For any integer x, there are at most $3^y(m+1)^x n^{m-m'-x}$ pairs of a sequence of vertices $v_1, \dots, v_m$ and a sequence of real numbers $c'_0, \dots, c'_{m'}$ such that $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0, \dots, v'_{m'}))$ is a level x shard of $W_{(c_0,\dots,c_m)[r]}((v_0, \dots, v_m))$.*

PROOF. Path-sum-conversion only deletes vertices that have a nonzero corresponding element of $c_0, \dots, c_m$, so there are at most 3^y possibilities for where the vertices that were deleted originally were, and which of them were copies of vertices in $v'_0, \dots, v'_{m'}$. There are at most $(m+1)^x$ possibilities for the identities of the x deleted vertices that are repetitions of vertices in $(v'_0, \dots, v'_{m'})$ and $n^{m-m'-x}$ possiblilities for the remaining $m-m'-x$ deleted vertices. Each possible choice of what vertices were deleted from where corresponds to at most one possible value of $(v_0, \dots, v_m)$, and there is only one shard that can result from deleting the designated vertices when running Path-sum-conversion$(W_{(c_0,\dots,c_m)[r]}((v_0, \dots, v_m)))$. □

LEMMA 6.19. *Let $\lambda'_1, \dots, \lambda'_{s-1}$ be a Λ-bounded eigenvalue approximation and $c_0, \dots, c_m$ be an $l[r]$-cycle of this approximation for some l and $r > 0$. Then, let $v_0, \dots, v_m$ be a series of vertices, and let $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0, \dots, v'_{m'}))$ be a level x shard of $W_{(c_0,\dots,c_m)[r]}((v_0, \dots, v_m)))$ such that $v'_0, \dots, v'_{m'}$ is nonbacktracking. The absolute value of the coefficient of $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0, \dots, v'_{m'}))$ in the sum resulting from* Path-sum-conversion$(W_{(c_0,\dots,c_m)[r]}((v_0, \dots, v_m)))$ *is at most $(\Lambda/n)^{m-m'}$.*

PROOF. First, note that since $r \geq 1$, for any $i < i'$ such that $c_i \neq 0$ and $c_{i'} \neq 0$, it must be the case that $i' - i \geq 3$. One can show that there is only one subseries of vertices with nonzero c_i that can be deleted from $v_0, \dots, v_m$ to yield $v'_0, \dots, v'_{m'}$. That means that there is only one route through which Path-sum-conversion$(W_{(c_0,\dots,c_m)[r]}((v_0, \dots, v_m)))$ arrives at $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0, \dots, v'_{m'}))$. Every time the algorithm deletes a vertex, it multiplies the expression's coefficient by $-c_i/n$, where i is that vertex's original index in $v_0, \dots, v_m$. So, the conclusion holds because $|-c_i/n| \leq \Lambda/n$ for all i. □

Bounding the Variance of $W_{m/S}$

The previous section provides the key techniques we will need to prove bounds. Now we need to apply them. We need to prove that expressions involving $W_{m/S}$ are within a certain range with high probability rather than merely computing their expected values, which requires us to bound their variances. That, in turn, will require us to bound the expected values of expressions of the form

$$W_{(c_0,\dots,c_m)[r]}((v_0, \dots, v_m)) \cdot W_{(c_0,\dots,c_m)[r]}((v''_0, \dots, v''_m)).$$

In order to convert that to something more familiar, let $u_1, \dots, u_r$ be some new vertices that are adjacent to every vertex in r, and then note that for any $c_0, \dots, c_{m_1}$, $c''_0, \dots, c''_{m_2} \in \mathbb{R}$ and $v_0, \dots, v_{m_1}, v''_0, \dots, v''_{m_2} \in G$,

$$\begin{aligned}&W_{(c_0,\dots,c_{m_1})[r]}((v_0, \dots, v_{m_1})) \cdot W_{(c''_0,\dots,c''_m)[r]}((v''_0, \dots, v''_{m_2})) = \\ &\quad W_{(c_0,\dots,c_{m_1},0,\dots,0,c''_{m_2},c''_{m_2-1},\dots,c''_0)[r]}((v_0, \dots, v_{m_1}, u_1, u_2, \dots, u_r, v''_{m_2}, v''_{m_2-1}, \dots, v''_0))\end{aligned}$$

because all of the vertices are connected to the u_i, and they create enough distance between the v_i and the v''_i that they will never be within r of each other.

Next we will establish a series of bounds on the expected values of these expressions, starting with the following.

LEMMA 6.20. *Let $\lambda'_1, \dots, \lambda'_{s-1}$ be a Λ-bounded eigenvalue approximation with error at most $2\Lambda(\min_i |\lambda_s - \lambda_i|/4\Lambda)^{s-1}|\lambda_s/\Lambda|^{l-s+1}$ and $c_0, \dots, c_{m_1}$ and $c''_0, \dots, c''_{m_2}$ be $l[r]$-cycles of this approximation for some l and r. Then, let*

$$\left(c'''_0, \dots, c'''_{m_1+m_2+r+1}\right) = \left(c_0, \dots, c_{m_1}, 0, \dots, 0, c''_{m_2}, c''_{m_2-1}, \dots, c''_0\right),$$

where there are r 0s in the middle. Next, let $v_0, \dots, v_{m_1}, v''_0, \dots, v''_{m_2} \in G$ and $W_{(c'_0,\dots,c'_{m'}[r])}((v'_0, \dots, v'_{m'}))$ be a level x shard of

$$W_{(c'''_0,\dots,c'''_{m_1+m_2+r+1})[r]}\big((v_0, \dots, v_m, u_1, u_2, \dots, u_r, v''_m, v''_{m-1}, \dots, v''_0)\big).$$

Define w and e such that there are $m' + 1 - w$ distinct vertices in $v'_0, \dots, v'_{m'}$ and $m' - e$ distinct unordered pairs $\{v'_i, v'_{i+1}\}$, and let $m'' = m' - r - 1$. Then $m_1 + m_2 - m'' \le x + 6(w - e) + 2 + e/r$. Also, if $(\Lambda/|\lambda_s|)^{l-s+1} \ge 2^{s-1}$ and $\lambda_s^2 > \Lambda$, then

$$\begin{aligned}
&|\mathbb{E}(W_{(c'_0,\dots,c'_{m'})[r]}((v'_0, \dots, v'_{m'}))| \\
&\quad \le n^{e-m''}(2^{s-1}k/\min p_i)^{3w-3e+2}|\lambda_s|^{m''}(2\Lambda/|\lambda_s|)^{(s-1)m''/l} \\
&\qquad \cdot (\Lambda/\lambda_s^2)^e\big((\Lambda/|\lambda_s|)^{l-s+1}2^{1-s}\big)^{\frac{(3w-3e+2)(l+2r+1)}{l}}(\Lambda/|\lambda_s|)^{2(l-s+1)x}
\end{aligned}$$

and also

$$\begin{aligned}
&|\mathbb{E}(W_{(c'_0,\dots,c'_{m'})[r]}((v'_0, \dots, v'_{m'}))| \\
&\quad \le n^{e-m''}(k/(\min p_i))^{3(w-e)+2}2^{(2x+3(w-e)+2)(s-1)} \\
&\qquad \cdot \left(\max_{s\le j\le h}|\lambda_j|^{l-s+1}\prod_{i=1}^{s-1}|\lambda_j - \lambda'_i|\right)^{m''/l} \\
&\qquad \cdot \left(\max_{s\le j\le h}(\lambda_j^2/\Lambda)^{l-s+1}\prod_{i=1}^{s-1}(\lambda_j - \lambda'_i)^2/\Lambda\right)^{-e/l} \\
&\qquad \cdot \left(\max_{s\le j\le h}(|\lambda_j|/\Lambda)^{l-s+1}\prod_{i=1}^{s-1}|\lambda_j - \lambda'_i|/\Lambda\right)^{-\frac{(3w-3e+2)(l+2r+1)}{l}-2x}.
\end{aligned}$$

PROOF. First, consider the standard decomposition of $(v'_0, \dots, v'_{m'})$. The only vertices that could have been deleted are vertices within r of the edge of a fresh segment with nonzero corresponding values of c'''_i, vertices that are not in a fresh segment with nonzero corresponding values of c'''_i, and the x vertices that were deleted as copies of vertices in $v'_0, \dots, v'_{m'}$. There are at most $2e$ indices i such that (v'_i, v'_{i+1}) is not in a fresh segment and there are no i and i' such that $|i - i'| \le 2r$, $c'''_i \ne 0$, and $c'''_{i'} \ne 0$. So, there are at most $6(w - e) + 2 + e/r$ vertices with nonzero c'''_i that are not in a fresh segment and farther than r from its edge.

Now, remove $u_1, \dots, u_r$ from whichever fresh segment they are in, splitting it into two fresh segments on either side of $u_1, \dots, u_r$ if necessary. Let d, d', t, t' be the measures of the resulting set of fresh segments, and $t'' = \lceil \frac{t-d(l+2r+1)}{l} \rceil$. By the walk decomposition lemma, $d + d' \le 3(w - e) + 2$. Also, note that every edge in one of the repeated fresh segments is repeated later in the walk. So, $t + 2t' \le m''$. Every edge except those that involve the u_i appears exactly once in a fresh segment, so $t + t' = m'' - e$ and $t \geq m'' - 2e$. On another note, for every $j < s$, we have that

$$|\lambda_j|^{l-s+1} \prod_{i=1}^{s-1} |\lambda_j - \lambda_i'| \leq \Lambda^{l-s+1} |\lambda_j - \lambda_j'| (2\Lambda)^{s-2} \leq$$

$$|\lambda_s|^{l-s+1} (\min_i |\lambda_s - \lambda_i|/2)^{s-1} \leq |\lambda_s|^{l-s+1} \prod_{i=1}^{s-1} |\lambda_s - \lambda_i'|.$$

Also, for every $j \geq s$, we have that

$$|\lambda_j|^{l-s+1} \prod_{i=1}^{s-1} |\lambda_j - \lambda_i'| \leq |\lambda_s|^{l-s+1} (2\Lambda)^{s-1}.$$

That means that if $(\Lambda/|\lambda_s|)^{l-s+1} \geq 2^{s-1}$ and $\lambda_s^2 > \Lambda$ then

$$\begin{aligned}
&|\mathbb{E}(W_{(c_0',\ldots,c_{m'}')[r]}((v_0',\ldots,v_{m'}'))| \\
&\quad \leq n^{e-m''} (k/(\min p_i))^{3(w-e)+2} 2^{(2x+3(w-e)+2)(s-1)} \\
&\qquad \cdot \left(\max_{s \leq j \leq h} |\lambda_j|^{l-s+1} \prod_{i=1}^{s-1} |\lambda_j - \lambda_i'| \right)^{m''/l} \\
&\qquad \cdot \left(\max_{s \leq j \leq h} (\lambda_j^2/\Lambda)^{l-s+1} \prod_{i=1}^{s-1} (\lambda_j - \lambda_i')^2/\Lambda \right)^{-e/l} \\
&\qquad \cdot \left(\max_{s \leq j \leq h} (|\lambda_j|/\Lambda)^{l-s+1} \prod_{i=1}^{s-1} |\lambda_j - \lambda_i'|/\Lambda \right)^{-\frac{(3w-3e+2)(l+2r+1)}{l} - 2x}.
\end{aligned}$$

Alternatively,

$$\begin{aligned}
&|\mathbb{E}(W_{(c_0',\ldots,c_{m'}')[r]}((v_0',\ldots,v_{m'}'))| \\
&\quad \leq n^{-t-t'} k^{d+d'} (\min p_i)^{-d-d'} \lambda_1^{t' + \max(t-(2x+d)(s-1)-l\cdot\max(t''-2x,0),0)} \\
&\qquad \cdot (\lambda_1 + \Lambda)^{t - \max(t-(2x+d)(s-1)-l\cdot\max(t''-2x,0),0) - l\cdot\max(t''-2x,0)} \\
&\qquad \cdot \left(\max_{1 \leq j \leq k} |\lambda_j|^{l-s+1} \prod_{i=1}^{s-1} |\lambda_j - \lambda_i'| \right)^{\max(t''-2x,0)} \\
&\quad \leq n^{e-m''} (2^{s-1} k/\min p_i)^{3w-3e+2} |\lambda_s|^{m''} (2\Lambda/|\lambda_s|)^{(s-1)m''/l} \\
&\qquad \cdot (\Lambda/\lambda_s^2)^e \left((\Lambda/|\lambda_s|)^{l-s+1} 2^{1-s} \right)^{\frac{(3w-3e+2)(l+2r+1)}{l}} (\Lambda/|\lambda_s|)^{2(l-s+1)x}.
\end{aligned}$$

□

DEFINITION 6.21. $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))$ is a degree z shard for $(\sigma_1,\sigma_2,\sigma_3,\sigma_4)$, m_1, m_2, r, $(c_0,\dots,c_{m_1})$, and $(c''_0,\dots,c''_{m_2})$ if there exists $x \in \mathbb{Z}$ and $v_0,\dots,v_{m_1},v''_0,\dots,v''_{m_2}$ such that $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))$ is a level x shard of

$$W_{(c_0,\dots,c_{m_1},0,\dots,0,c''_{m_2},c''_{m_2-1},\dots,c''_0)[r]}((v_0,\dots,v_{m_1},u_1,\dots,u_r,v''_{m_2},v''_{m_2-1},\dots,v''_0)),$$

the number of distinct sets $\{v'_i,v'_{i+1}\}$ minus the number of distinct vertices in $v'_0,\dots,v'_{m'}$ is $z-x-1$, and v_0, v_{m_1}, v''_0, and v''_{m_2} are in communities σ_1, σ_2, σ_3, and σ_4, respectively.

LEMMA 6.22 (Degree bound lemma). *For any functions m_1 and m_2 of n such that $m_1,m_2 = O(\log n)$, there exists r_0 such that for any $r \ge r_0$, $z \ge 0$, and Λ, s, l such that $(\Lambda/|\lambda_s|)^{l-s+1} \ge 2^{s-1}$, $\lambda_s^2 > \Lambda > \lambda_1$, and $l \ge (2r+1)(s-1)$, the following holds: Let $\lambda'_1,\dots,\lambda'_{s-1}$ be a Λ-bounded eigenvalue approximation and $c_0,\dots,c_{m_1}$ and $c''_0,\dots,c''_{m_2}$ be $l[r]$-cycles of this approximation with error less than $2\Lambda(\min_i|\lambda_s-\lambda_i|/4\Lambda)^{s-1}|\lambda_s/\Lambda|^{l-s+1}$. For any $\sigma_1,\sigma_2,\sigma_3,\sigma_4$, the expected value of the sum of the absolute values of all shards of degree at least z for $(\sigma_1,\sigma_2,\sigma_3,\sigma_4)$, m_1, m_2, r, $(c_0,\dots,c_{m_1})$, and $(c''_0,\dots,c''_{m_2})$ times their coefficients is $O(n^{(15-5z)/6}\prod|\lambda_s-c_i|\cdot\prod|\lambda_s-c''_i|)$.*

PROOF. Let $m = m_1+m_2$. For any given x, w, m', and e with $x+w-e > z$ and $x \le m_1+m_2+r+1-m' \le x+6(w-e)+2+e/r$, the expected sum of the values of all level x shards for $(\sigma_{v_0},\sigma_{v_{m_1}},\sigma_{v''_0},\sigma_{v''_{m_2}})$, m_1, m_2, r, $(c_0,\dots,c_{m_1})$, and $(c''_0,\dots,c''_{m_2})$ with $m'+1$ vertices, $m'+1-w$ distinct vertices, and $m'-e$ distinct adjacent pairs of vertices has an absolute value of at most

$$\begin{aligned}
&[2n^{m'-w+1}(m+r+2)^{4(w-e)}2^{2\lceil\frac{m'+1}{r+1}\rceil(w-e)}] \\
&\quad\cdot(\Lambda/n)^{m+r+1-m'}\cdot[3^{m/r}(m+2)^x n^{m+1-m'-x}] \\
&\quad\cdot\big[n^{e-m'+r+1}(2^{s-1}k/\min p_i)^{3w-3e+2}|\lambda_s|^{m'-r-1}\cdot(2\Lambda/|\lambda_s|)^{(s-1)\cdot(m'-r-1)/l} \\
&\qquad\cdot(\Lambda/\lambda_s^2)^e\big((\Lambda/|\lambda_s|)^{l-s+1}2^{1-s}\big)^{\frac{(3w-3e+2)(l+2r+1)}{l}}(\Lambda/|\lambda_s|)^{2(l-s+1)x}\big] \\
&\quad\le|\lambda_s|^m n^{(14-5z)/6}
\end{aligned}$$

provided that $r > 12m/\log_2(n)$ and n is sufficiently large. There are at most $2(s-1)+(s-1)m/l$ indices i such that $c_i \ne 0$ or $c''_i \ne 0$, so

$$|\lambda_s|^{-m}\prod|\lambda_s-c_i|\prod|\lambda_s-c''_i| \ge (\min_{s'<s}|\lambda_s-\lambda_{s'}|/2)^{(s-1)m/l}.$$

So, as long as r_0 is large enough that $\frac{m}{2r+1}\ln(\min_{s'<s}|\lambda_s-\lambda_{s'}|/2) < \ln(n)/3$, then $|\lambda_s|^m n^{(14-5z)/6} \le n^{(15-5z)/6}\prod|\lambda_s-c_i|\cdot\prod|\lambda_s-c''_i|$. □

LEMMA 6.23. *Assume that m, Λ, $\lambda'_1,\dots,\lambda'_{s-1}$, l, r, and $(c_0,\dots,c_m)$ satisfy the conditions of the degree bound lemma. Also assume that $m = \Omega(\log n)$ and either*

$s = h'$ or $|\lambda_s| > |\lambda_{s+1}|$. Now, let w be an eigenvector of PQ with an eigenvalue of λ_j. If $j \neq s$, then with probability $1 - o(1)$ the average value over all v of $w_{\sigma_v} W_{m/\{c_i\}}(x, v)/p_{\sigma_v}$ is $O(\frac{1}{\ln(n)\sqrt{n}} \prod |\lambda_s - c_i|)$, and if $j = s$ then the average value over all v of $w_{\sigma_v} W_{m/\{c_i\}}(x, v)/p_{\sigma_v}$ is $\Omega(\frac{1}{\sqrt{n}} \prod |\lambda_j - c_i|)$ with probability $1 - o(1)$.

PROOF. Let $m''' = \lfloor \sqrt{\ln(n)} \rfloor$ unless $c_{\lfloor \sqrt{\ln(n)} \rfloor} \neq 0$, in which case let $m''' = \lfloor \sqrt{\ln(n)} \rfloor + 1$. The first step in proving the desired bounds is to justify approximating $\sum_{v \in \Omega_j} W_{m/\{c_i\}}(x, v)$ with

$$\sum_{v_0,\dots,v_{m'''} \in G} x_{v_0} \cdot W_{(c_0,\dots,c_{m'''})[r]}((v_0, \dots, v_{m'''}) e_{\sigma_{v_{m'''}}} \cdot \prod_{i=m'''+1}^{m} (PQ - c_i) e_j.$$

In order to do that, we observe that for any j,

$$E\Big[\sum_{v \in \Omega_j} W_{m/\{c_i\}}(x, v)$$

$$- \sum_{v_0,\dots,v_{m'''} \in G} x_{v_0} \cdot W_{(c_0,\dots,c_{m'''})[r]}((v_0, \dots, v_{m'''}) e_{\sigma_{v_{m'''}}} \cdot \prod_{i=m'''+1}^{m} (PQ - c_i) e_j)^2\Big]$$

is equal to

$$E\Big[\Big(\sum_{\substack{v_0,\dots,v_m,v'_0,\dots,v'_m \in G: \\ v_m, v'_m \in \Omega_j}}$$

$$x_{v_0} x_{v'_0} W_{(c_0,\dots,c_m,0,\dots,0,c_m,\dots,c_0)[r]}(v_0, \dots, v_m, u_1, \dots, u_r, v'_m, \dots, v'_0)$$

$$- x_{v_0} x_{v'_0} W_{(c_0,\dots,c_m,0,\dots,0,c_{m'''},\dots,c_0)[r]}(v_0, \dots, v_m, u_1, \dots, u_r, v'_{m'''}, \dots, v'_0)$$

$$e_{\sigma_{v'_{m'''}}} \cdot \prod_{i=m'''+1}^{m} \frac{1}{n}(PQ - c_i) e_j$$

$$- x_{v_0} x_{v'_0} W_{(c_0,\dots,c_{m'''},0,\dots,0,c_m,\dots,c_0)[r]}(v_0, \dots, v_{m'''}, u_1, \dots, u_r, v'_m, \dots, v'_0)$$

$$e_{\sigma_{v_{m'''}}} \cdot \prod_{i=m'''+1}^{m} \frac{1}{n}(PQ - c_i) e_j$$

$$+ x_{v_0} x_{v'_0} W_{(c_0,\dots,c_{m'''},0,\dots,0,c_{m'''},\dots,c_0)[r]}(v_0, \dots, v_{m'''}, u_1, \dots, u_r, v'_{m'''}, \dots, v'_0)$$

$$\Big(e_{\sigma_{v_{m'''}}} \cdot \prod_{i=m'''+1}^{m} \frac{1}{n}(PQ - c_i) e_j\Big)\Big(e_{\sigma_{v'_{m'''}}} \cdot \prod_{i=m'''+1}^{m} \frac{1}{n}(PQ - c_i) e_j\Big)\Big)\Big].$$

Note that $E[v_0 v'_0]$ is 1 if $v_0 = v'_0$ and 0 otherwise. So, the only $(v_0, \dots, v_m)$, $(v'_0, \dots, v'_m)$ that make a nonzero contribution to the expected value of the sum above are those for which $v_0 = v'_0$. Also, given any $(v_0, \dots, v_m)$, $(v'_0, \dots, v'_m)$

such that $v_{m'''+1}, \dots, v_m$ are distinct and $\{v_{m'''+1}, \dots, v_m\} \cap (\{v_0, \dots, v_{m'''}\} \cup \{v'_0, \dots, v'_m\}) = \varnothing$, then for any assignment of communities to $v_{m'''}, v_m$, and v'_m, and any possible value of the subgraph of G induced by $\{v_0, \dots, v_{m'''}\} \cup \{v'_0, \dots, v'_m\}$, the expected value of $W_{(c_0,\dots,c_m)[r]}(v_0, \dots, v_m)$ given these values is $W_{(c_0,\dots,c_{m'''})[r]}(v_0, \dots, v_{m'''}) e_{\sigma_{m'''}} \cdot \prod_{i=m'''+1}^{m} (PQ - c_i) e_j$. That means that the expected contribution to the sum above of the terms corresponding to such $(v_0, \dots, v_m), (v'_0, \dots, v'_m)$ is 0. By the same logic, all $(v_0, \dots, v_m), (v'_0, \dots, v'_m)$ such that $v'_{m'''+1}, \dots, v'_m$ are distinct and $\{v'_{m'''+1}, \dots, v'_m\} \cap (\{v'_0, \dots, v'_{m'''}\} \cup \{v_0, \dots, v_m\}) = \varnothing$ have an expected contribution of 0 to the sum above.

Now, let V be the set of all pairs of tuples $((v_0, \dots, v_m), (v'_0, \dots, v'_m))$ such that $v_0 = v'_0$, either $|\{v_{m'''+1}, \dots, v_m\}| < m - m'''$ or

$$\{v_{m'''+1}, \dots, v_m\} \cap (\{v_0, \dots, v_{m'''}\} \cup \{v'_0, \dots, v'_m\}) \neq \varnothing,$$

and either $|\{v'_{m'''+1}, \dots, v'_m\}| < m - m'''$ or $\{v'_{m'''+1}, \dots, v'_m\} \cap (\{v'_0, \dots, v'_{m'''}\} \cup \{v_0, \dots, v_m\}) \neq \varnothing$. Note that for every $v_0, \dots, v_{m'''}$ and $v'_0, \dots, v'_m$, there are at most $(2m+2)^2 n^{m-m'''-1}$ possible choices of $v_{m'''+1}, \dots, v_m$ such that

$$((v_0, \dots, v_m), (v'_0, \dots, v'_m)) \in V.$$

So,

$$\begin{aligned}
E\Bigg[& \sum_{\substack{((v_0,\dots,v_m),(v'_0,\dots,v'_m)) \in V: \\ v_m, v'_m \in \Omega_j}} \Bigg| x_{v_0} x_{v'_0} W_{(c_0,\dots,c_m,0,\dots,0,c_{m'''},\dots,c_0)[r]}(v_0, \dots, v_m, u_1, \dots, u_r, v'_{m'''}, \dots, v'_0) \\
& \qquad e_{\sigma_{v'_{m'''}}} \cdot \prod_{i=m'''+1}^{m} \frac{1}{n} (PQ - c_i) e_j \Bigg| \Bigg] \\
& \le [(2m+2)^2 n^{m-m'''-1}] \cdot \Bigg[n^{m'''-m} / (\min_i p_i) \max_{1 \le i' \le h} \prod_{i=m'''+1}^{m} |\lambda_{i'} - c_i| \Bigg] \\
& \quad \cdot E\Bigg[\sum_{\substack{v_0,\dots,v_m,v'_0,\dots,v'_{m'''} \in G: \\ v_m, v'_m \in \Omega_j}} |W_{(c_0,\dots,c_m,0,\dots,0,c_{m'''},\dots,c_0)[r]}(v_0, \dots, v_m, u_1, \dots, u_r, v'_{m'''}, \dots, v'_0)| \Bigg] \\
& = O\Bigg(n^{5/6} \prod_{i=1}^{m} (\lambda_s - c_i)^2 \Bigg)
\end{aligned}$$

where the equality follows from the degree bound lemma and the facts that $v_0 = v'_0$ implies that every shard of a W in this expression has nonzero degree,

$$\prod_{i=m'''}^{m} |\lambda_{i'} - c_i| = O\Bigg(\prod_{i=m'''}^{m} |\lambda_s - c_i| \Bigg)$$

for any i', and $m = O(\ln(n))$. The same logic applies to the analogous expressions for

$$W_{(c_0,\dots,c_{m'''},0,\dots,0,c_m,\dots,c_0)[r]}(v_0, \dots, v_{m'''}, u_1, \dots, u_r, v'_m, \dots, v'_0)$$

and

$$W_{(c_0,\ldots,c_{m'''},0,\ldots,0,c_{m'''},\ldots,c_0)[r]}(v_0,\ldots,v_{m'''},u_1,\ldots,u_r,v'_{m'''},\ldots,v'_0).$$

This implies that

$$\begin{aligned}
&E\Big[\Big(\sum_{v\in\Omega_j} W_{m/\{c_i\}}(x,v)\\
&\quad -\sum_{v_0,\ldots,v_{m'''}\in G} x_{v_0}\cdot W_{(c_0,\ldots,c_{m'''})[r]}((v_0,\ldots,v_{m'''})e_{\sigma_{v_{m'''}}}\cdot \prod_{i=m'''+1}^{m}(PQ-c_i)e_j\Big)^2\Big]\\
&\quad = E\Big[\sum_{\substack{((v_0,\ldots,v_m),(v'_0,\ldots,v'_m))\in V:\\ v_m,v'_m\in\Omega_j}} W_{(c_0,\ldots,c_m,0,\ldots,0,c_m,\ldots,c_0)[r]}(v_0,\ldots,v_m,u_1,\ldots,u_r,v'_m,\ldots,v'_0)\Big]\\
&\qquad + O\Big(n^{5/6}\prod_{i=1}^{m}(\lambda_s-c_i)^2\Big).
\end{aligned}$$

If the $W_{(c_0,\ldots,c_m,0,\ldots,0,c_m,\ldots,c_0)[r]}(v_0,\ldots,v_m,u_1,\ldots,u_r,v'_m,\ldots,v'_0)$ are broken down into weighted sums of shards, then every resulting shard has degree at least 1 for the same reason as in the previous cases, and all resulting shards of degree at least 2 have a combined contribution to the expected value of at most $n^{5/6}\prod_{i=1}^{m}(\lambda_s-c_i)^2$ by the degree bound lemma.

Now, let $W_{(c'_0,\ldots,c'_{m'})[r]}(v''_0,\ldots,v''_{m'})$ be a degree 1 shard of

$$W_{(c_0,\ldots,c_m,0,\ldots,0,c_m,\ldots,c_0)[r]}(v_0,\ldots,v_m,u_1,\ldots,u_r,v'_m,\ldots,v'_0)$$

for some $((v_0,\ldots,v_m),(v'_0,\ldots,v'_m))\in V$. Also, let H be the walk graph of $v''_0,\ldots,v''_{m'}$ with $u_1,\ldots,u_r$ removed. We know that $v''_0=v_0=v'_0=v''_{m'}$, so H is connected. That means that the only way that $W_{(c'_0,\ldots,c'_{m'})[r]}(v''_0,\ldots,v''_{m'})$ can be a degree 1 shard is if H is a tree and no vertex that was deleted in the process of converting $W_{(c_0,\ldots,c_m,0,\ldots,0,c_m,\ldots,c_0)[r]}(v_0,\ldots,v_m,u_1,\ldots,u_r,v'_m,\ldots,v'_0)$ to $W_{(c'_0,\ldots,c'_{m'})[r]}(v''_0,\ldots,v''_{m'})$ is repeated in $v''_0,\ldots,v''_{m'}$. So, there must exist some $t_0\geq 0$ such that $v''_i=v''_{m'-i}$ for all $i\leq t_0$, $v''_{t_0+1},\ldots,v''_{m'-t_0-1}$ are distinct, and $\{v''_0,\ldots,v''_{t_0}\}\cap\{v''_{t_0+1},\ldots,v''_{m'-t_0-1}\}=\varnothing$. Furthermore, since $((v_0,\ldots,v_m),(v'_0,\ldots,v'_m))\in V$, there exist i and i' such that $i>m''$ and $v_i=v'_{i'}$ or $v_i=v_{i'}$ with $i\neq i'$. Either neither of the repeated elements were deleted in the process of converting

$$W_{(c_0,\ldots,c_m,0,\ldots,0,c_m,\ldots,c_0)[r]}(v_0,\ldots,v_m,u_1,\ldots,u_r,v'_m,\ldots,v'_0)$$

to $W_{(c'_0,\ldots,c'_{m'})[r]}(v''_0,\ldots,v''_{m'})$, or they both were, in which case v_i must have been within r of a repeated vertex that was not deleted during the conversion. The only vertices that could have been deleted during the conversion are those with nonzero corresponding c_i, so either way, we have that $t_0 \geq \frac{2r}{2r+1}m'' - r$. Also, by Lemma 6.20, we have that

$$\begin{aligned}
&|E[W_{(c'_0,\dots,c'_{m'})[r]}(v''_0,\dots,v''_{m'})]| \\
&\quad \leq n^{t_0-m'+r+1}(k/(\min p_i))^5 2^{5(s-1)} \\
&\qquad \cdot \left(\max_{s\leq j\leq h} |\lambda_j|^{l-s+1}\prod_{i=1}^{s-1}|\lambda_j - \lambda'_i|\right)^{(m'-r-1)/l} \\
&\qquad \cdot \left(\max_{s\leq j\leq h} (\lambda_j^2/\Lambda)^{l-s+1}\prod_{i=1}^{s-1}(\lambda_j - \lambda'_i)^2/\Lambda\right)^{-t_0/l} \\
&\qquad \cdot \left(\max_{s\leq j\leq h} (|\lambda_j|/\Lambda)^{l-s+1}\prod_{i=1}^{s-1}|\lambda_j - \lambda'_i|/\Lambda\right)^{-\frac{5(l+2r+1)}{l}}.
\end{aligned}$$

For fixed values of t_0 and m', there are at most $2m \cdot n^{(m'-t_0-r)}$ possible values of $(v''_0,\dots,v''_{m'})$. Furthermore, for each possible value of $(v''_0,\dots,v''_{m'})$, there are at most $2^{t_0/r+1}n^{2m+r+1-m'}$ possible values of $v_0,\dots,v_m$, $v'_0,\dots,v'_m$, and $c'_0,\dots,c'_{m'}$ such that $W_{(c'_0,\dots,c'_{m'})[r]}(v''_0,\dots,v''_{m'})$ is a degree 1 shard of

$$W_{(c_0,\dots,c_m,0,\dots,0,c_m,\dots,c_0)[r]}(v_0,\dots,v_m,u_1,\dots,u_r,v'_m,\dots,v'_0),$$

and this shard has a coefficient with an absolute value of at most $(\Lambda/n)^{2m+r+1-m'}$. So, the combined contribution to the expected value from all degree 1 shards with a given value of t_0 and m' is at most

$$\begin{aligned}
&2^{t_0/r+2}m\Lambda^{2m+r+1-m'}n(2^{s-1}k/(\min p_i))^5 \\
&\quad \cdot \left(\max_{s\leq j\leq h} |\lambda_j|^{l-s+1}\prod_{i=1}^{s-1}|\lambda_j - \lambda'_i|\right)^{(m'-r-1)/l} \\
&\quad \cdot \left(\max_{s\leq j\leq h} (\lambda_j^2/\Lambda)^{l-s+1}\prod_{i=1}^{s-1}(\lambda_j - \lambda'_i)^2/\Lambda\right)^{-t_0/l} \\
&\quad \cdot \left(\max_{s\leq j\leq h} (|\lambda_j|/\Lambda)^{l-s+1}\prod_{i=1}^{s-1}|\lambda_j - \lambda'_i|/\Lambda\right)^{-\frac{5(l+2r+1)}{l}} \\
&\quad = O\left(n\ln^{-5}(n)\prod_{i=0}^{m}(\lambda_s - c_i)^2\right).
\end{aligned}$$

There are only $O(\ln^2(n))$ possible values of t_0 and m', so the combined contribution of all degree 1 shards is $O(n\ln^{-3}(n)\prod_{i=0}^{m}(\lambda_s - c_i)^2)$. Therefore, with probability $1 - o(1)$, the average value over all $v \in \Omega_\sigma$ of $W_{m/\{c_i\}}(x, v)$ is within $o(\frac{\sqrt{n}}{\log(n)} \prod |\lambda_s - c_i|)$ of

$$\frac{1}{p_\sigma} e_\sigma \cdot \prod_{i=m''+1}^{m} (PQ - c_i) \sum_{(v_0,\dots,v_{m'''})} x_{v_0} \cdot W_{(c_0,\dots,c_{m'''})[r]}((v_0, \dots, v_{m'''}))e_{\sigma_{v_{m''}}}.$$

If w is an eigenvector of PQ with eigenvalue λ_j, then this means that the average value of $w_{\sigma_v} W_{m/\{c_i\}}(x, v)/p_{\sigma_v}$ is within $o(\frac{1}{\log(n)} \prod |\lambda_s - c_i|)$ of

$$\frac{1}{n} \prod_{i=m''+1}^{m} (\lambda_j - c_i) \sum_{(v_0,\dots,v_{m'''})} x_{v_0} \cdot W_{(c_0,\dots,c_{m'''})[r]}((v_0, \dots, v_{m'''}))w_{\sigma_{v_{m''}}}/p_{\sigma_{v_{m''}}}.$$

For fixed G and σ but not fixed x, the probability distribution of this expression is a normal distribution with mean 0 and variance

$$\frac{1}{n^2} \prod_{i=m''+1}^{m} (\lambda_j - c_i)^2 \sum_{v_0} \Big(\sum_{(v_1,\dots,v_{m'''})} W_{(c_0,\dots,c_{m'''})[r]}((v_0, \dots, v_{m'''}))w_{\sigma_{v_{m''}}}/p_{\sigma_{v_{m''}}} \Big)^2.$$

With probability $1-o(1)$, there is no $v_0 \in G$ that has more than one cycle within m''' edges of it, there are $O(\lambda_1^{2m'''})$ vertices that have a cycle within m''' edges of them, and no vertex in the graph has degree greater than $\ln^2(n)$. This implies that

$$\sum_{v_0} \Big(\sum_{(v_1,\dots,v_{m'''})} W_{(c_0,\dots,c_{m'''})[r]}((v_0, \dots, v_{m'''}))w_{\sigma_{v_{m''}}}/p_{\sigma_{v_{m''}}} \Big)^2 = O(n \ln^{2m'''}(n)).$$

If $j \neq s$, then there exists $\epsilon > 0$ such that

$$\prod_{i=m''+1}^{m} (\lambda_j - c_i)^2 = O\Big(n^{-\epsilon} \prod_{i=m''+1}^{m} (\lambda_s - c_i)^2 \Big),$$

so the variance is $o(\frac{1}{n \log^2(n)} \prod (\lambda_s - c_i)^2)$. Thus, the specified average is

$$O\Big(\frac{1}{\log(n)\sqrt{n}} \prod |\lambda_s - c_i| \Big)$$

with probability $1 - o(1)$, as desired.

Now, consider the case where $j = s$. If there is no cycle in the portion of the graph within m'' edges of v_0, then every nonbacktracking walk of length m'' or less starting at v_0 is a path. So, conditioned on the absence of such cycles near v_0 and a fixed value of σ_{v_0}, we have that

$$\Big(\sum_{(v_1,\dots,v_{m'''})} W_{(c_0,\dots,c_{m'''})[r]}((v_0, \dots, v_{m'''}))w_{\sigma_{v_{m''}}}/p_{\sigma_{v_{m''}}} \Big)^2 = \\ (1 + o(1)) \prod_{i=1}^{m'''-1} (\lambda_s - c_i) w_{\sigma_{v_0}}/p_{\sigma_{v_0}}.$$

Thus the expected value of the square of this sum is $\Omega(\prod_{i=1}^{m'''-1}(\lambda_s - c_i)^2)$. Furthermore, for any v and v' there is no vertex within m'' edges of both v and v' with probability $1 - O(\lambda_1^{2m'''}/n)$. So, the joint probability distribution of the subgraphs of G within m'' edges of v and v' differs from the product of the individual distributions by $O(\lambda_1^{2m''}/n)$. So, with probability $1 - o(1)$ we have that

$$\sum_{v_0}\Big(\sum_{(v_1,\dots,v_{m'''})} W_{(c_0,\dots,c_{m'''})[r]}((v_0,\dots,v_{m'''}))w_{\sigma_{v_{m''}}}/p_{\sigma_{v_{m''}}}\Big)^2 = \Omega\Big(n\prod_{i=1}^{m'''-1}(\lambda_s - c_i)^2\Big).$$

Thus the average value over all v of $w_{\sigma_v}W_{m/\{c_i\}}(x,v)/p_{\sigma_v}$ is $\frac{1}{\sqrt{n}}\Omega(\prod|\lambda_s - c_i|)$ with probability $1 - o(1)$. □

LEMMA 6.24. *There exists a constant r_0 and $m_0 = \Theta(\log n)$ such that if m, Λ, $\lambda'_1,\dots,\lambda'_{s-1}$, l, r, and $(c_0,\dots,c_m)$ satisfy the conditions of the degree bound lemma, $r > r_0$, and $m > m_0$, then for any communities σ_1,σ_2,*

$$\Bigg|\mathbb{E}\sum_{\substack{v_0,\dots,v_m,v''_0,\dots,v''_m\in G:\\ v_m=v''_m,v_0=v''_0,\\ \sigma_{v_m}=\sigma_1,\sigma_{v_0}=\sigma_2}} W_{(c_0,\dots,c_m)[r]}((v_0,\dots,v_m))\cdot W_{(c_0,\dots,c_m)[r]}((v''_0,\dots,v''_m))\Bigg|$$
$$= O\Big(\Big(\sum_{i=s}^{h}\prod_{j=0}^{m}|\lambda_i - c_j|\Big)^2\Big).$$

PROOF. Let $W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))$ be a level x shard of

$$W_{(c_0,\dots,c_m,0,\dots,0,c_m,\dots,c_0)[r]}((v_0,\dots,v_m,u_1,\dots,u_r,v''_m,\dots,v''_0)),$$

and G' be the walk graph of $(v'_0,\dots,v'_{m'})$. $c_0 = c''_0 = 0$, so $v'_0 = v'_{m'}$ is the only vertex in G' that can have degree 1. Since $c_m = c''_m = 0$, $v_m \in G$ and it is adjacent to u_1, u_r, and v_{m-1}, meaning that v_m has degree greater than 2 and this shard has nonzero degree.

By the degree-bound lemma, the contribution to this expected value of all shards of degree greater than 2 is $O(\prod(\lambda_s - c_i)^2)$. Now, assume that this shard has degree less than 3, and let G'' be G' with $u_1,\dots,u_r$ and all of their edges removed. G'' is still connected, and the only vertices that might have degree 1 in it are v_0 and v_m.

First, consider the case in which G'' is a tree. It must be a path, so $v'_0,\dots,v'_{m'}$ simply consists of a path, followed by a cycle, followed by the same path in reverse. There are at most m/r nonzero elements of $(c_0,\dots,c_m,0.,\dots,0,c_m,\dots,c_0)$, so if I is the set of indices of vertices in $(v_0,\dots,v_m,u_1,\dots,u_r,v''_m,\dots,v''_0)$ that were deleted in the process of converting it to this shard, there are at most $2^{m/r}$ possible values of I. Also, $m' = 2m + r + 1 - |I|$ and for a given value of I, there are at most $n^{(2m+2-|I|)/2}$ possible values of $(v'_0,\dots,v'_{m'})$, each of which could have come from at most $n^{|I|}$ possible values of $(v_0,\dots,v_m)$ and $(v'_0,\dots,v'_m)$. The coefficient of the shard is at most $(\Lambda/n)^{|I|}$, and $c'_i = 0$ for all i. That means that

$W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))$ is 1 if each of the $m-|I|/2$ edges in G'' are also in G and 0 otherwise. So, $|\mathbb{E}[W_{(c'_0,\dots,c'_{m'})[r]}((v'_0,\dots,v'_{m'}))]| = O((\lambda_1/n)^{m-|I|/2})$. That means that the total contribution to the expected value of all such shards is $O(2^{m/r}\Lambda^{m/r}\lambda_1^{m-m/2r}n)$. So, as long as

$$m > \ln(n) \Big/ \bigg(\ln(\lambda_s^2/\lambda_1) - \ln(4\Lambda^2/\lambda_1)/2r + \frac{2(s-1)}{l}\ln\Big(\min_{i<s}|\lambda_i - \lambda_s|/2|\lambda_s|\Big)\bigg), \tag{6.2}$$

the contribution to the expected value of all such shards is

$$O\bigg(\Big(\sum_{i=s}^{h}\prod_{j=0}^{m}|\lambda_i - c_j|\Big)^2\bigg),$$

as desired.

That leaves the case where G'' is not a tree. The fact that the shard has degree less than 3 implies that G' has at most 1 more edge than it has vertices, so G'' has at least as many vertices as edges. So, if it is not a tree, it must have an equal number of edges and vertices, which means that it contains exactly one cycle. The only vertices in G'' that might have degree 1 are v_0 and v_m, so G'' consists of a cycle and two paths (possibly of length 0) that connect v_0 and v_m to the cycle. Now, let e be the length of the path from v_0 to the cycle, e' be the length of the path from v_m to the cycle, and $|I|$ be the set of all indices of vertices in $(v_0,\dots,v_m,u_1,\dots,u_r,v''_m,\dots,v''_0)$ that were deleted in the process of converting it to $(v'_0,\dots,v'_{m'})$. $m' = 2m+r+1-|I|$, and the shard will have a coefficient of magnitude at most $(\Lambda/n)^{|I|}$.

There are a few subcases to consider. First, consider the case in which some but not all of the edges in the cycle in G'' show up more than once in the walk defined by $(v'_0,\dots,v'_{m'})$. The only way for this to happen is if the edges on one side of the cycle are repeated and the edges on the other side are not. The only way for this to happen is if $(v'_0,\dots,v'_{m'})$ simply goes across the repeated branch of the cycle when it goes from v_0 to v_m and then goes around the cycle 1 and a fraction times on its way back or vice versa. So, each edge in the repeated branch shows up exactly 3 times in the walk defined by $(v'_0,\dots,v'_{m'})$.

Let e'' be the number of edges on the repeated side, and e''' be the number of edges on the nonrepeated side. $e''' = 2m-2e-2e'-3e''-|I| \le 2(m-e-e'-e'')-|I|$, and there are at most $(2e+2e'+3e''+2r)/2r$ indices of vertices that could be in I. For fixed values of e, e', e'', and I, there are at most $n^{2m-e-e'-2e''-|I|}$ possible labelings of the vertices in G'', each of which corresponds to at most two possible values of $(v'_0,\dots,v'_{m'})$. Each of those could be derived from at most $n^{|I|}$

possible values of $(v_0,\dots,v_m)$ and $(v_0'',\dots,v_m'')$. Also, the absolute value of the shard's expected value is at most

$$\begin{aligned}
&n^{e+e'+2e''-2m+|I|}(k/(\min p_i))^8 2^{8(s-1)}\\
&\cdot\left(\max_{s\le j\le h}|\lambda_j|^{l-s+1}\prod_{i=1}^{s-1}|\lambda_j-\lambda_i'|\right)^{(2m-|I|)/l}\\
&\cdot\left(\max_{s\le j\le h}(\lambda_j^2/\Lambda)^{l-s+1}\prod_{i=1}^{s-1}(\lambda_j-\lambda_i')^2/\Lambda\right)^{-(e+e'+2e'')/l}\\
&\cdot\left(\max_{s\le j\le h}(|\lambda_j|/\Lambda)^{l-s+1}\prod_{i=1}^{s-1}|\lambda_j-\lambda_i'|/\Lambda\right)^{-\frac{8(l+2r+1)}{l}}.
\end{aligned}$$

So, all such shards have a combined expected contribution of absolute value

$$O\left(\left(\sum_{i=s}^{h}\prod_{j=0}^{m}(\lambda_i-c_j)\right)^2\right).$$

One can prove that all shards for which no edge in the cycle in G'' shows up more than once in the walk defined by $(v_0',\dots,v_{m'}')$ and all shards for which every edge in the cycle in G'' shows up more than once in the walk defined by $(v_0',\dots,v_{m'}')$ have combined expected contributions of absolute value of

$$O\left(\left(\sum_{i=s}^{h}\prod_{j=0}^{m}(\lambda_i-c_j)\right)^2\right)$$

using similar reasoning. □

LEMMA 6.25. *There exists a constant r_0 and $m_0=\Theta(\log n)$ such that if m, Λ, $\lambda_1',\dots,\lambda_{s-1}'$, l, r, and $(c_0,\dots,c_m)$ satisfy the conditions of the degree bound lemma, $r>r_0$, and $m>m_0$, then for all v, we have that*

$$\operatorname{Var}[W_{m/\{c_i\}}(x,v)]=O\left(\prod(\lambda_s-c_i)^2\right).$$

PROOF. We have

$$\begin{aligned}
&W_{m/\{c_i\}}^2(x,v)=\\
&\sum_{\substack{v_0,\dots,v_m,v_0'',\dots,v_m''\in G:\\ v_m=v_m''=v}} x_{v_0}x_{v_0''}W_{(c_0,\dots,c_m)}((v_0,\dots,v_m))\cdot W_{(c_0,\dots,c_m)}((v_0'',\dots,v_m'')).
\end{aligned}$$

If $v_0 \neq v_0''$ then $E[x_{v_0} \cdot x_{v_0''}] = 0$, while if $v_0 = v_0''$ then $E[x_{v_0} \cdot x_{v_0''}] = 1$. So,

$$\begin{aligned} &E[W^2_{m/\{c_i\}}(x,v)] \\ &\quad = \sum_{\substack{v_0,\dots,v_m,v_0'',\dots,v_m''\in G: \\ v_0=v_0'', v_m=v_m''=v}} E[W_{(c_0,\dots,c_m)}((v_0,\dots,v_m)) \cdot W_{(c_0,\dots,c_m)}((v_0'',\dots,v_m''))] \\ &\quad = O\Big(\prod(\lambda_s - c_i)^2\Big) \end{aligned}$$

where the last equality follows by the previous lemma. □

6.2 Crossing the KS Threshold

For $x \in [k]^n$ and $\varepsilon > 0$, define the set of bad clusterings with respect to x as

$$B_\varepsilon(x) = \left\{ y \in [n]^k : \frac{1}{n} d_*(x,y) > 1 - \frac{1}{k} - \varepsilon \right\}, \tag{6.3}$$

where $d_*(x,y) = \min_{\pi \in S_k} d_H(x, \pi(y))$; d_H is the Hamming distance and $\pi(y)$ denotes the application of π to each component of y.

Recall that

$$\mathrm{Bal}(n,k,\varepsilon) = \left\{ x \in [k]^n : \forall i \in [k], \frac{|\{u \in [n] : x_u = i\}|}{n} \in \left[\frac{1}{k} - \varepsilon, \frac{1}{k} + \varepsilon\right] \right\}$$

and

$$\begin{aligned} T_\delta(G) = \Bigg\{ & x \in \mathrm{Bal}(n,k,\delta) : \\ & \sum_{i=1}^{k} \left| \left\{ G_{u,v} : (u,v) \in \binom{[n]}{2} \text{ s.t. } x_u = i, x_v = i \right\} \right| \geq \frac{an}{2k}(1-\delta), \\ & \sum_{i,j \in [k], i<j} |\{G_{u,v} : (u,v) \in \binom{[n]}{2} \text{ s.t. } x_u = i, x_v = j\}| \leq \frac{bn(k-1)}{2k}(1+\delta) \Bigg\}. \end{aligned}$$

Atypicality of a Bad Clustering

LEMMA 6.26. *Let $\varepsilon > 0$, $x, y \in \mathrm{Bal}(n,k,\delta)$ such that $y \in B_\epsilon(x)$ and $(\sigma, G) \sim \mathrm{SBM}(n,k,a/n,b/n)$. Then,*

$$\mathbb{P}\{y \in T_\delta(G) \mid \sigma = x\} \leq \exp\Big(-\frac{n}{k}A(\varepsilon,\delta)\Big)$$

where $A(\varepsilon,\delta)$ is continuous at $(\varepsilon,\delta) = (0,0)$ and

$$A(0,0) = \frac{a + b(k-1)}{2} \ln \frac{k}{(a+(k-1)b)} + \frac{a}{2}\ln a + \frac{b(k-1)}{2}\ln b.$$

Recall that $d := \frac{a+(k-1)b}{k}$.

COROLLARY 6.27. *Detection is solvable in* SBM$(n, k, \frac{a}{n}, \frac{b}{n})$ *if*

$$\frac{1}{2\ln k}\left(\frac{a\ln a + (k-1)b\ln b}{k} - d\ln d\right) > 1.$$

Note that for $a = 0$, the above bound for b is $\Theta(\ln k/k)$ times the bound given by the KS threshold. Further, this improves on the KS threshold for $k \geq 5$. However, for $b = 0$, the above bound gives $a > 2k$, which is worse than the KS threshold $a > k$. For the same reasons, it is loose for $k = 2$ and $a = 0$. In the next section, we improve the bound to capture the right behavior at the extremal regimes.

PROOF OF COROLLARY 6.27. Let $(\sigma, G) \sim \mathrm{SBM}(n, k, a/n, b/n)$ where k, a, b satisfy

$$\frac{1}{2\ln k}\left(\frac{a\ln a + (k-1)b\ln b}{k} - d\ln d\right) > 1. \tag{6.4}$$

Let $\widehat{\sigma}_\delta(G)$ be uniformly drawn in $T_\delta(G)$. We have

$$\begin{aligned}\mathbb{P}\{\widehat{\sigma}_\delta(G) \in B_\epsilon(\sigma)\} &= \mathbb{P}\{\widehat{\sigma}_\delta(G) \in B_\epsilon(\sigma), \sigma \in \mathrm{Bal}(n,k,\delta)\} + o(1)\\ &= \mathbb{E}\frac{|T_\delta(G) \cap B_\epsilon(\sigma) \mid \mathbb{1}(\sigma \in \mathrm{Bal}(n,k,\delta))}{|T_\delta(G)|} + o(1)\\ &\leq \mathbb{E}|T_\delta(G) \cap B_\epsilon(\sigma) \mid \mathbb{1}(\sigma \in \mathrm{Bal}(n,k,\delta)) + o(1)\end{aligned} \tag{6.5}$$

where we use the fact that $|T_\delta(G)| \geq 1$ with high probability, since $\sigma \in T_\delta(G)$ with high probability, and $\sigma \in \mathrm{Bal}(n,k,\delta)$ with high probability. Moreover,

$$\mathbb{P}\{\widehat{\sigma}_\delta(G) \in B_\epsilon(\sigma)\} \leq \sum_{y \in [k]^n} \sum_{\substack{x \in \mathrm{Bal}(n,k,\delta):\\ y \in B_\epsilon(x)}} \mathbb{P}\{y \in T_\delta(G) \mid \sigma = x\}\mathbb{P}\{\sigma = x\} + o(1).$$

Letting τ be the bound obtained from Lemma 6.26 on $\mathbb{P}\{y \in T_\delta(G) \mid \sigma = x\}$, we have

$$\mathbb{P}\{\widehat{\sigma}_\delta(G) \in B_\epsilon(\sigma)\} \leq k^n\tau + o(1). \tag{6.6}$$

We can now take $\delta > 0$ such that for a strictly positive ε, (6.4) implies that (6.6) vanishes. □

PROOF OF LEMMA 6.26. Assume that $a \geq b$ (the other case is treated similarly). Let $(\sigma, G) \sim \mathrm{SBM}(n, k, a, b)$ and $x, y \in \mathrm{Bal}(n,k,\delta)$ such that $y \in B_\varepsilon(x)$. Denote by $T_\delta^{(\mathrm{in})}(G)$ and $T_\delta^{(\mathrm{out})}(G)$ the sets of clusterings having typical fraction of edges inside and across clusters, respectively. We have

$$\mathbb{P}\{y \in T_\delta(G) \mid \sigma = x\} = \mathbb{P}\big\{y \in T_\delta^{(\mathrm{in})}(G) \mid \sigma = x\}\mathbb{P}\{y \in T_\delta^{(\mathrm{out})}(G) \mid \sigma = x\big\}, \tag{6.7}$$

with

$$\begin{aligned}
&\mathbb{P}\{y \in T_\delta^{(\mathrm{in})}(G) \mid \sigma = x\} \\
&\quad = \mathbb{P}\left\{\sum_{i=1}^{k} \left| \left\{ G_{u,v} : (u,v) \in \binom{[n]}{2} \text{ s.t. } y_u = i, y_v = i \right\} \right| \geq \frac{an}{2k}(1-\delta) \mid \sigma = x \right\} \\
&\quad \leq \mathbb{P}\left\{ \mathrm{Bin}\left(\frac{n(n-1)}{2k^2}(1+\xi), \frac{a}{n} \right) + \mathrm{Bin}\left(\frac{(k-1)n(n-1)}{2k^2}(1+\xi), \frac{b}{n} \right) \geq \frac{an}{2k}(1-\delta) \right\}
\end{aligned}$$

$$\begin{aligned}
&\mathbb{P}\{y \in T_\delta^{(\mathrm{out})}(G) | \sigma = x\} \\
&\quad = \mathbb{P}\left\{ \sum_{i,j \in [k], i<j} \left| \left\{ G_{u,v} : (u,v) \in \binom{[n]}{2} \text{ s.t. } y_u = i, y_v = j \right\} \right| \leq \frac{bn(k-1)}{2k}(1+\delta) \mid \sigma = x \right\} \\
&\quad \leq \mathbb{P}\left\{ \mathrm{Bin}\left(\frac{n^2(k-1)}{2k^2}(1-\xi), \frac{a}{n} \right) + \mathrm{Bin}\left(\frac{n^2(k-1)^2}{2k^2}(1-\xi), \frac{b}{n} \right) \leq \frac{bn(k-1)}{2k}(1+\delta) \right\},
\end{aligned}$$

where $\mathrm{Bin}(N,p)$ denotes a binomial random variable with N trials of bias p, all binomial random variables are independent, and $\xi \leq 2k^2(\varepsilon + \delta)$ (the latter gives a loose bound on the offset of the trial counts in the binomials due to the fact that x, y are not exactly but approximately balanced).

Since the dependence on ε, δ is continuous, we next proceed without the ξ-terms to prove the lemma. We have

$$\begin{aligned}
&\mathbb{P}\left\{ \mathrm{Bin}\left(\frac{n(n-1)}{2k^2}, \frac{a}{n} \right) + \mathrm{Bin}\left(\frac{(k-1)n(n-1)}{2k^2}, \frac{b}{n} \right) \geq \frac{an}{2k} \right\} \\
&\quad \leq O(n^2) \max_{cn \in [0, \frac{n^2}{2k^2}]} \mathbb{P}\left\{ \mathrm{Bin}\left(\frac{n^2}{2k^2}, \frac{a}{n} \right) = cn \right\} \mathbb{P}\left\{ \mathrm{Bin}\left(\frac{(k-1)n^2}{2k^2}, \frac{b}{n} \right) = \frac{an}{2k} - cn \right\} \\
&\quad \leq \exp\left(-\frac{n}{k}\left(\frac{a+(k-1)b}{2k} + \frac{a}{2} \ln \frac{ak}{e(a+(k-1)b)} \right) + o(n) \right).
\end{aligned}$$

In addition,

$$\begin{aligned}
&\mathbb{P}\left\{ \mathrm{Bin}\left(\frac{n^2(k-1)}{2k^2}, \frac{a}{n} \right) + \mathrm{Bin}\left(\frac{n^2(k-1)^2}{2k^2}, \frac{b}{n} \right) \leq \frac{bn(k-1)}{2k} \right\}, \\
&\quad \leq O(n^2) \\
&\qquad \cdot \max_{cn \in [0, \frac{n^2(k-1)}{2k^2}]} \mathbb{P}\left\{ \mathrm{Bin}\left(\frac{n^2(k-1)}{2k^2}, \frac{a}{n} \right) = cn \right\} \mathbb{P}\left\{ \mathrm{Bin}\left(\frac{n^2(k-1)^2}{2k^2}, \frac{b}{n} \right) = \frac{bn(k-1)}{2k} - cn \right\} \\
&\quad \leq \exp\left(-\frac{n}{k}\left(\frac{(k-1)(a+(k-1)b)}{2k} + \frac{b(k-1)}{2} \ln \frac{bk}{e(a+(k-1)b)} \right) + o(n) \right).
\end{aligned}$$

The result follows from algebraic manipulations. □

Size of the Typical Set

The goal of this section is to lower-bound the size of the typical set $T_\delta(G)$. Recall that the expected node degree in $\mathrm{SBM}(n,k,a,b)$ is

$$d := \frac{a+(k-1)b}{k}.$$

THEOREM 6.28. *Let $T_\delta(G)$ denote the typical set of clusterings for G drawn under* $\mathrm{SBM}(n,k,a,b)$*. Then, for any $\varepsilon > 0$,*

$$\mathbb{P}\{|T_\delta(G)| < \max(k^{(\psi-\varepsilon)n}, 2^{(e^{-a/k}(1-(1-e^{-b/k})^{k-1})-\epsilon)n})\} = o(1),$$

where

$$\begin{aligned}\psi := {} & \frac{\tau}{d}\left(1-\frac{\tau}{2}\right) \\ & + \frac{1}{\ln(k)}\left(\frac{a}{a+(k-1)b}\ln\left(\frac{a+(k-1)b}{a}\right) + \frac{(k-1)b}{a+(k-1)b}\ln\left(\frac{a+(k-1)b}{b}\right)\right) \\ & \qquad \cdot \left(\frac{\tau^2}{2d} + (d-\tau)e^{-(d-\tau)}\right),\end{aligned}$$

and $\tau = \tau_d$ is the unique solution in $(0,1)$ of

$$\tau e^{-\tau} = de^{-d} \tag{6.8}$$

or equivalently $\tau = \sum_{j=1}^{+\infty} \frac{j^{j-1}}{j!}(de^{-d})^j$.

As a first step to proving this theorem, we prove one-half of the inequality.

LEMMA 6.29. *Let $T_\delta(G)$ denote the typical set of clusterings for G drawn under* $\mathrm{SBM}(n,k,a,b)$*. Then, for any $\varepsilon > 0$,*

$$\mathbb{P}\{|T_\delta(G)| < 2^{(e^{-a/k}(1-(1-e^{-b/k})^{k-1})-\epsilon)n}\} = o(1).$$

PROOF. Construct a clustering of G, σ' as follows. Start with $\sigma' = \sigma$. Then, go through the vertices of G in a random order, and for each v with no neighbor v' such that $\sigma'_v = \sigma'_{v'}$, change σ'_v to a random element of $[k]\backslash\{\sigma'_{v'} : (v,v') \in E(G)\}$.

Initially, $\sigma' = \sigma$, so the probability distributions of (G,σ) and (G,σ') are identical. For a given ordering of the vertices, if these probability distributions are identical immediately before σ'_v is changed, then for given values of G and $\sigma'\backslash\sigma'_v$, it is equally likely that σ'_v had each value that it may be assigned in this step. So, the probability distribution is unchanged. Therefore, the probability distribution of (G,σ') is identical to the probability distribution of (G,σ) at any point in this algorithm. In particular, $\sigma' \in T_\delta(G)$ with probability $1-o(1)$.

Now, for each $v \in G$, let z_v be 1 if there is more than one option for σ'_v when it is time for the algorithm to assign it and 0 otherwise. A vertex has no neighbors in its own community with probability $e^{-a/k} + o(1)$, and no neighbors in any given other community with probability $e^{-b/k} + o(1)$. So, a vertex has no neighbors in its own community and at least one other community with probability $e^{-a/k}(1 - (1-e^{-b/k})^{k-1}) + o(1)$. Thus, for each v,

$$\mathbb{E}(z_v) = e^{-a/k}(1-(1-e^{-b/k})^{k-1}) + o(1).$$

Also, given two different vertices, they both have no neighbors in their own community and at least one other with probability $e^{-2a/k}(1-(1-e^{-b/k})^{k-1})^2 + o(1)$.

Given any v and v', assume without loss of generality that v is before v' in the random ordering. With probability $1-o(1)$, v' will not be in the last $\sqrt{n}$ vertices, and at the time v' comes up in the ordering, $e^{-a/k}(1-(1-e^{-b/k})^{k-1})+o(1)$ of the remaining vertices will have no neighbors claimed to be in their communities and at least one other. So, symmetry between the vertices implies that the correlation between z_v and $z_{v'}$ is $o(1)$. That means that with probability $1-o(1)$, we have

$$\sum_{v\in G} z_v = e^{-a/k}(1-(1-e^{-b/k})^{k-1})n + o(n).$$

For fixed values of (G,σ) and any given σ'_0, z_0, we have that

$$\mathbb{P}\Big\{\sigma'=\sigma'_0, \sum_{v\in G} z_v \ge z_0\Big\} \le 2^{-z_0}$$

because every time the algorithm has more than one option for σ'_v, there is at most a $\frac{1}{2}$ chance that it sets σ'_v to $(\sigma'_0)_v$, and if $\sum z_v \ge z_0$, then there are at least z_0 times when it had such a choice. So, for a fixed graph G,

$$\text{(6.9)}\quad \mathbb{P}\Big\{\sigma' \in T_\delta(G), \sum_{v\in G} z_v > e^{-a/k}(1-(1-e^{-b/k})^{k-1})n - \epsilon n/2\Big\} \le |T_\delta(G)| \cdot 2^{e^{-a/k}(1-(1-e^{-b/k})^{k-1})n-\epsilon n/2}.$$

We already know that with probability $1-o(1)$ the probability in question is $1-o(1)$, so the conclusion holds. □

To prove the rest of this theorem, we need some topological properties of the SBM graph, analogous to the Erdős-Rényi case [38].

LEMMA 6.30. *Let T_j denote the number of isolated j-trees (i.e., trees on j vertices) in* SBM(n,k,a,b) *and M_j the number of edges contained in those trees. Then, for any $\varepsilon>0$,*

$$\mathbb{P}\{T_j/n \notin [\tau_j/d-\varepsilon, \tau_j/d+\varepsilon]\} = o(1),$$
$$\mathbb{P}\{M_j/n \notin [(j-1)\tau_j/d-\varepsilon, (j-1)\tau_j/d+\varepsilon]\} = o(1),$$

where

$$\tau_j = \frac{j^{j-2}(de^{-d})^j}{j!}.$$

LEMMA 6.31. *Let T denote the number of isolated trees in* SBM(n,k,a,b) *and M the number of edges contained in those trees. Then, for any $\varepsilon>0$,*

$$\mathbb{P}\Big\{T/n \notin \Big[\frac{\tau}{d}\Big(1-\frac{\tau}{2}\Big)-\varepsilon, \frac{\tau}{d}\Big(1-\frac{\tau}{2}\Big)+\varepsilon\Big]\Big\} = o(1),$$
$$\mathbb{P}\Big\{M/n \notin \Big[\frac{\tau^2}{2d}-\varepsilon, \frac{\tau^2}{2d}+\varepsilon\Big]\Big\} = o(1),$$

where τ is defined in (6.8).

LEMMA 6.32 ([18]). *Let Q denote the number of nodes that are in the giant component in* $\mathrm{SBM}(n,k,a,b)$*, i.e., nodes that are in a linear-size component of* $\mathrm{SBM}(n,k,a,b)$*. Then, for any $\varepsilon > 0$,*

$$\mathbb{P}\{Q/n \notin [\beta-\varepsilon, \beta+\varepsilon]\} = o(1),$$

where $\beta = 1 - \tau/d$ and τ is defined in (6.8).

LEMMA 6.33. *Let F denote the number of edges that are in planted trees of the giant in* $\mathrm{SBM}(n,k,a,b)$*, i.e., trees formed by nodes that have a single edge connecting them to a linear-size component in* $\mathrm{SBM}(n,k,a,b)$*. Then, for any $\varepsilon > 0$,*

$$\mathbb{P}\{F/n \notin [\phi-\varepsilon, \phi+\varepsilon]\} = o(1),$$

where $\phi := (d-\tau)e^{-(d-\tau)}$ and τ is defined in (6.8).

PROOF OF THEOREM 6.28. Let $G \sim \mathrm{SBM}(n,k,a,b)$, and let T be the number of isolated trees in G, M the number of edges in those trees, and F the number of edges in the planted trees of the largest connected component of G (i.e., the giant). We have from Lemma 6.30, Lemma 6.31, and Lemma 6.33 that, for $\varepsilon > 0$, with high probability on G,

$$\begin{aligned} T &\in \Big[\frac{\tau}{d}\Big(1-\frac{\tau}{2}\Big) - \varepsilon, \frac{\tau}{d}\Big(1-\frac{\tau}{2}\Big) + \varepsilon\Big], \\ R &\in \left[\frac{\tau^2}{2d} + (d-\tau)e^{-(d-\tau)} - \varepsilon, \frac{\tau^2}{2d} + (d-\tau)e^{-(d-\tau)} + \varepsilon\right]. \end{aligned}$$

Assume that T and R take typical values as above. We now build a typical vertex labeling on these trees:

- Pick an arbitrary node in each isolated tree, denote these by $\{v_1,\dots,v_T\}$, and denote the set of edges contained in these trees by $\{E_1,\dots,E_M\}$.
- Pick the root node for each planted tree that is in the giant, denote these by $\{w_1,\dots,w_k\}$, where k is the number of planted trees (which is not relevant in the following computations), and denote by $\{E_{M+1},\dots,E_{M+F}\}$ the number of edges contained in those planted trees.
- Assign the labels $U_1^T := (U_{v_1},\dots,U_{v_T})$ uniformly at random in $[k]$, and set $\widehat{X}_{w_1} = \sigma_{w_1},\dots,\widehat{X}_{w_k} = \sigma_{w_k}$, i.e., assign the latter labels to their true community assignments. Then broadcast each of these labels in their corresponding trees by forwarding the labels on each edge with an independent k-ary symmetric channel of flip probability $\frac{b}{a+(k-1)b}$. This means that the variables $Z_1,\dots,Z_{M+F}$ are drawn i.i.d. from the distribution

$$\nu := \left(\frac{a}{a+(k-1)b}, \frac{b}{a+(k-1)b}, \dots, \frac{b}{a+(k-1)b}\right) \tag{6.10}$$

on $\mathbb{F}_k := \{0,1,\dots,k-1\}$, and that for each edge e in the trees, the input bit is forwarded by adding to it the Z_e variable modulo k.

- Assign any other label (that is not contained in the trees) to their true community assignments. Define $R := M + F$, $Z_1^R := (Z_1, \dots, Z_R)$, and denote by $\widehat{X}(U_1^T, Z_1^R)$ the previously defined assignment.

Note that the above gives the induced label distribution on trees in SBM(n, k, a, b). Thus, with high probability on G, as T and M grow with n, this assignment is typical with high probability on U_1^T, Z_1^R, i.e.,

$$\mathbb{P}_{U_1^T, Z_1^R}\left\{\widehat{X}\left(U_1^T, Z_1^R\right) \in T_\delta(G)\right\} = 1 - o(1). \tag{6.11}$$

Denote by $\mathrm{Emp}(Z_1^R)$ the empirical distribution on $\mathbb{F}_k$ of the R-vector Z_1^R, and by $\mathcal{B}_\varepsilon(\nu)$ the l_1-ball around μ of radius ε. Denote by η the uniform distribution on $[k]$. Then by Sanov's theorem, for any $\varepsilon > 0$,

$$\mathbb{P}_{Z_1^R}\left\{\mathrm{Emp}\left(U_1^T\right) \in \mathcal{B}_\varepsilon(\eta), \mathrm{Emp}\left(Z_1^R\right) \in \mathcal{B}_\varepsilon(\nu)\right\} \to 1, \tag{6.12}$$

as T, R diverge with n. Define now the set of realizations of Z_1^R that have a typical likelihood by

$$A_\varepsilon(\nu) := \left\{z_1^R \in \mathbb{F}_k^R : k^{-R(H(\nu)+\varepsilon)} \le \mathbb{P}\left\{Z_1^R = z_1^r\right\} \le k^{-R(H(\nu)-\varepsilon)}\right\}$$

where H is the entropy with the logarithm in base k. For convenience of notation, define also $A_\varepsilon(\eta) = \mathcal{B}_\varepsilon(\eta)$. Again, for all $\varepsilon > 0$,

$$\mathbb{P}\left\{U_1^T \in A_\varepsilon(\eta), Z_1^R \in A_\varepsilon(\nu)\right\} \to 1.$$

Therefore,

$$\begin{aligned}
&\mathbb{P}_{U_1^T, Z_1^R}\left\{\widehat{X}\left(U_1^T, Z_1^R\right) \in T_\delta(G)\right\} \\
&\quad = \mathbb{P}_{U_1^T, Z_1^R}\left\{\widehat{X}\left(U_1^T, Z_1^R\right) \in T_\delta(G), U_1^T \in A_\varepsilon(\eta), Z_1^R \in A_\varepsilon(\nu)\right\} + o(1) \\
&\quad \le \sum_{\substack{u_1^T \in A_\varepsilon(\eta), \\ z_1^R \in A_\varepsilon(\nu)}} \mathbb{1}\left(\widehat{X}\left(u_1^T, z_1^R\right) \in T_\delta(G)\right) k^{-T} k^{-R(H(\nu)-\varepsilon)} + o(1).
\end{aligned} \tag{6.13}$$

We have

$$\sum_{\substack{u_1^T \in A_\varepsilon(\eta), \\ z_1^R \in A_\varepsilon(\nu)}} \mathbb{1}\left(\widehat{X}\left(u_1^T, z_1^R\right) \in T_\delta(G)\right) \le |T_\delta(G)|,$$

since the left-hand side counts a subset of the typical clusterings. We thus obtain from (6.11) and (6.13) that with high probability on G,

$$|T_\delta(G)| \ge (1 - o(1)) k^{T + R(H(\nu)-\varepsilon)}.$$

Thus, with high probability on G,

$$|T_\delta(G)| \ge (1 - o(1)) k^{n(\psi-\varepsilon)+o(n)},$$

where

$$\psi = \frac{\tau}{d}\left(1 - \frac{\tau}{2}\right) + H(\nu)\left(\frac{\tau^2}{2d} + (d - \tau)e^{-(d-\tau)}\right),$$

which proves the half of the claim not covered by Lemma 6.29. □

PROOF OF LEMMA 6.30. Let T_j denote the number of isolated j-trees (i.e., trees on j vertices) in SBM(n, k, a, b), i.e.,

$$T_j = \sum_{S \in \mathcal{T}_j(n)} \mathbb{1}_{\text{iso}}(S),$$

where $\mathcal{T}_j(n)$ denotes the set of all j-trees on the vertex set $[n]$, and where $\mathbb{1}_{\text{iso}}(S)$ is equal to 1 if S is an isolated tree and 0 otherwise. Since the number of different trees on j vertices is given by j^{j-2} [25], and since there is an edge on a designated pair of vertices with probability d/n, we have

$$\mathbb{E}[T_j] = \binom{n}{j} j^{j-2} \left(\frac{d}{n}\right)^{j-1} \left(1 - \frac{d}{n}\right)^{j(n-j)} = \frac{n}{d}(de^{-d})^j \frac{j^{j-2}}{j!} + O(1)$$

and

$$\operatorname{Var} T_j = \frac{1}{2}\frac{n}{d}(de^{-d})^{2j} \left(\frac{j^{j-2}}{j!}\right)^2 + O(1) = O(n).$$

Thus, by Chebyshev's inequality, for any $\varepsilon > 0$,

$$\mathbb{P}\{T_j/n \notin [\tau_j/d - \varepsilon, \tau_j/d + \varepsilon]\} = O(1/n),$$

where

$$\tau_j = \frac{j^{j-2}(de^{-d})^j}{j!},$$

and since each tree contains $j - 1$ edges,

$$\mathbb{P}\{M_j/n \notin [(j-1)\tau_j/d - \varepsilon, (j-1)\tau_j/d + \varepsilon]\} = O(1/n). \qquad \square$$

PROOF OF LEMMA 6.31. Let T denote the number of isolated trees in SBM(n, k, a, b), i.e., $T = \sum_{j=1}^{n} T_j$, where T_j is the number of isolated j-trees. Hence,

$$\mathbb{E}[T] = \sum_{j=1}^{n} \mathbb{E}[T_j] = \frac{n}{d} \sum_{j=1}^{n} \frac{j^{j-2}(de^{-d})^j}{j!} + O(1).$$

Since $\sum_{j=n}^{\infty} \frac{j^{j-2}(de^{-d})^j}{j!} = O(n^{-3/2})$, we have

$$\lim_{n\to\infty} \frac{1}{n}\mathbb{E}[T] = \frac{1}{d} \sum_{j=1}^{\infty} \frac{j^{j-2}(de^{-d})^j}{j!}.$$

Finally, since $\operatorname{Var}(T) = O(n)$, the result for T follows from Chebyshev's inequality and the fact that (see [67])

$$\sum_{j=1}^{\infty} \frac{j^{j-2}(de^{-d})^j}{j!} = \tau - \frac{\tau^2}{2} \tag{6.14}$$

when $\tau = \sum_{j=1}^{\infty} \frac{j^{j-1}(de^{-d})^j}{j!}$. For M, the result follows from similar arguments and the fact that

$$\begin{aligned}\mathbb{E}[M] = \sum_{j=1}^{n}(j-1)\mathbb{E}[T_j] &= \frac{n}{d}\sum_{j=1}^{n}(j-1)\frac{j^{j-2}(de^{-d})^j}{j!} + O(1)\\ &= \frac{n}{d}\frac{\tau^2}{2} + O(1),\end{aligned}$$

which uses again (6.14). □

PROOF OF LEMMA 6.33. From Lemma 6.32, the giant component has with high probability a relative size in $[\beta-\varepsilon, \beta+\varepsilon]$. The probability that a node is connected to a giant component by a single edge is thus given by

$$\beta n(d/n)(1-d/n)^{\beta n} + o(1) = \beta d e^{-\beta d} + o(1), \tag{6.15}$$

and the expected number of such nodes is

$$n\beta d e^{-\beta d} + o(n). \tag{6.16}$$

Now, let v and v' be random vertices. On the condition that v is connected to the giant component by a single edge, the giant component still has relative size in $[\beta-\varepsilon, \beta+\varepsilon]$ for any ε with probability $1-o(1)$, so the probability that v' is connected to the giant component by a single edge is also $\beta d e^{-\beta d} + o(1)$. That means that the expected value of the square of the number of nodes that are connected to the giant component by a single edge is

$$(n\beta d e^{-\beta d})^2 + o(n^2);$$

hence the variance in the number of such nodes is $o(n^2)$ and the lemma follows. □

Sampling Probability Estimates

PROOF OF THEOREM 2.10. Let

$$t = \max(k^{(\psi-\varepsilon)n}, 2^{(e^{-a/k}(1-(1-e^{-b/k})^{k-1})-\epsilon)n}).$$

We have

$$\begin{aligned}&\mathbb{P}\{\hat{\sigma}(G) \in B_\varepsilon(\sigma)\}\\ &\quad = \mathbb{E}\frac{|T_\delta(G) \cap B_\varepsilon(\sigma)|}{|T_\delta(G)|}\\ &\quad \le \mathbb{E}\frac{|T_\delta(G) \cap B_\varepsilon(\sigma)|}{|T_\delta(G)|}\mathbb{1}(|T_\delta(G)| \ge t)\mathbb{1}(\sigma \in \mathrm{Bal}(n,k,\delta)) + o(1) \qquad (6.17)\\ &\quad \le (1/t)\cdot\mathbb{E}|T_\delta(G) \cap B_\varepsilon(\sigma)|\mathbb{1}(\sigma \in \mathrm{Bal}(n,k,\delta)) + o(1)\\ &\quad \le (1/t)k^n e^{-A(\varepsilon,\delta)n/k} + o(1).\end{aligned}$$

Using the bound $t \geq 2^{(e^{-a/k}(1-(1-e^{-b/k})^{k-1})-\epsilon)n}$ and the bound on $A(\varepsilon, \delta)$ from Lemma 6.26, the exponent in (6.17) can be made to vanish if

$$\begin{aligned}&\frac{1}{k}\left(\frac{a+b(k-1)}{2}\ln\frac{k}{(a+(k-1)b)}+\frac{a}{2}\ln a+\frac{b(k-1)}{2}\ln b\right)\\&\quad+e^{-a/k}(1-(1-e^{-b/k})^{k-1})\ln(2)\\&\quad\quad>\ln(k).\end{aligned}$$

Alternatively, plugging in the values of ψ from Theorem 6.28 and $A(\varepsilon, \delta)$ from Lemma 6.26, the exponent in (6.17) is vanishing if

$$\begin{aligned}&\frac{1}{k\ln(k)}\left(\frac{a+b(k-1)}{2}\ln\frac{k}{(a+(k-1)b)}+\frac{a}{2}\ln a+\frac{b(k-1)}{2}\ln b\right)+\frac{\tau}{d}\left(1-\frac{\tau}{2}\right)\\&\quad+\frac{1}{\ln(k)}\left(\frac{a}{a+(k-1)b}\ln\left(\frac{a+(k-1)b}{a}\right)+\frac{(k-1)b}{a+(k-1)b}\ln\left(\frac{a+(k-1)b}{b}\right)\right)\\&\quad\quad\cdot\left(\frac{\tau^2}{2d}+(d-\tau)e^{-(d-\tau)}\right)>1.\end{aligned}\tag{6.18}$$

Since τ is the solution in $(0,1)$ of $\tau e^{-\tau} = de^{-d}$, we have

$$\frac{\tau^2}{2d}+(d-\tau)e^{-(d-\tau)}=\tau\left(1-\frac{\tau}{2d}\right),$$

and this simplifies to

$$\begin{aligned}&\frac{1}{2\ln(k)}\left(-d\ln d+\frac{a\ln a+b(k-1)\ln b}{k}\right)\\&\quad+\frac{1}{2\ln(k)}\left(d\ln d+d\ln k-\frac{a\ln a+b(k-1)\ln b}{k}\right)\cdot\frac{2\tau}{d}\left(1-\frac{\tau}{2d}\right)\\&\quad>1-\frac{\tau}{d}\left(1-\frac{\tau}{2}\right).\end{aligned}\tag{6.19}$$

Algebraic manipulations lead to the bound in the theorem. □

PROOF OF COROLLARY 2.13. Note that dropping the term in (6.18) above and ignoring the contribution of (6.18) leads to the weaker bound

$$\frac{1}{2\ln k}\left(\frac{a\ln a+(k-1)b\ln b}{k}-d\ln d\right)>1-\frac{\tau}{d}\left(1-\frac{\tau}{2}\right),$$

which implies equation (2.7) in Corollary 2.13 since $\frac{\tau}{d}(1-\frac{\tau}{2})$ tends to $\frac{1}{2}$ as τ tends to 1 when b tends to 0 and d tends to a/k. Equation (2.6) in the corollary follows from algebraic manipulations. □

6.3 Learning the Model

The proof is analogous to the case $k = 2$ from [61].

LEMMA 6.34. *Let G be drawn from* $\mathrm{SBM}(n,p,W)$*, $m>0$, and $v_0,\dots,v_m$ be vertices such that $v_i \neq v_j$ whenever $|i-j|<\max(m,2)$. For any fixed values of σ_{v_0} and σ_{v_m}, the probability that there is an edge between v_i and v_{i+1} for all $1 \le i < m$ is $e_{\sigma_{v_0}} \cdot P^{-1}(PW)^m e_{\sigma_{v_m}}$.*

PROOF. We proceed by induction on m. If $m=1$, then the probability that there is an edge between v_0 and v_1 is $W_{\sigma_{v_0},\sigma_{v_1}} = e_{\sigma_{v_0}} \cdot P^{-1}PWe_{\sigma_{v_m}}$, as desired. Now, assume that the lemma holds for $m=m_0$, and consider the case where $m=m_0+1$. The probability that there is an edge between v_i and v_{i+1} for all $1 \le i < m$ is

$$\begin{aligned}&\sum_i (e_{\sigma_{v_0}} \cdot P^{-1}(PW)^{m-1}e_i)p_i W_{i,\sigma_{v_m}}\\ &\quad= \sum_i (e_{\sigma_{v_0}} \cdot P^{-1}(PW)^{m-1}e_i)(p_i e_i \cdot We_{\sigma_{v_m}})\\ &\quad= e_{\sigma_{v_0}} \cdot P^{-1}(PW)^m e_{\sigma_{v_m}}\end{aligned}$$

as desired. □

COROLLARY 6.35. *The expected number of cycles of length m in* $\mathrm{SBM}(n,p,Q/n)$ *is asymptotic to $\frac{1}{2m}\sum_{i=1}^k \lambda_i^m$, where $\{\lambda_i\}$ are the eigenvalues of PQ, with multiplicity.*

PROOF. There are $n(n-1)\cdots(n-m+1)=\Theta(n^m)$ possible series of m distinct vertices in the graph, and each cycle has $2m$ possible choices of a starting vertex and a direction. The starting vertex is in community i with probability p_i. So, the expected number of length-m cycles is asymptotic to

$$\frac{n^m}{2m}\sum_i p_i e_i \cdot P^{-1}(PQ/n)^m e_i = \frac{1}{2m}\sum_i e_i \cdot (PQ)^m e_i = \frac{1}{2m}\sum_{i=1}^k \lambda_i^m. \qquad \square$$

PROOF OF LEMMA 2.17. The variance in the number of cycles is asymptotic to the mean, so the expected difference between the actual number of cycles of a given size and the expected number is proportional to the square root of the expected number. In the symmetric SBM where two vertices in the same community are connected with probability a/n and two vertices in different communities are connected with probability b/n, this means that with high probability, the number of cycles of length m is

$$\frac{1}{2m}\left(\frac{a+(k-1)b}{k}\right)^m + \frac{k-1}{2m}\left(\frac{a-b}{k}\right)^m + O\left(\left(\frac{a+(k-1)b}{k}\right)^{m/2} \Big/ \sqrt{m}\ln(n)\right).$$

Now, assume that $(\frac{a-b}{k})^2 > \frac{a+(k-1)b}{k}$. Clearly $\frac{a+(k-1)b}{k}$ can be computed up to an error of $O(1/\sqrt{n})$ by counting the edges in the graph, so given the number of cycles of length m for all $m \le M = \omega(1)$, one can determine $(a-b)/k$ and k with error asymptotic to 0, which provides enough information to determine a and b with error asymptotic to 0.

In order to obtain an efficient estimator, one can count nonbacktracking walks instead of cycles. For $m = o(\log_d^{1/4}(n))$, the neighborhood at depth m of a vertex in $\mathrm{SBM}(n,k,a,b)$ contains at most one cycle with high probability. Thus the number of cycles of length m in $\mathrm{SBM}(n,k,a,b)$ is with high probability equal to $\sum_{v\in[n]} C_v$ where C_v is the number of nonbacktracking walks of length m starting and ending at v. This is efficiently computable as shown in [61]. □

Acknowledgment. This research was partly supported by National Science Foundation CAREER Award CCF-1552131, ARO Grant W911NF-16-1-0051, and the Google Faculty Research Award.

Bibliography

[1] Abbe, E.; Bandeira, A. S.; Bracher, A.; Singer, A. Decoding binary node labels from censored edge measurements: phase transition and efficient recovery. *IEEE Trans. Network Sci. Eng.* **1** (2014), no. 1, 10–22. doi:10.1109/TNSE.2014.2368716

[2] Abbe, E.; Bandeira, A. S.; Hall, G. Exact recovery in the stochastic block model. *IEEE Trans. Inform. Theory* **62** (2016), no. 1, 471–487. doi:10.1109/TIT.2015.2490670

[3] Abbe, E.; Montanari, A. Conditional random fields, planted constraint satisfaction, and entropy concentration. *Theory Comput.* **11** (2015), 413–443. doi:10.4086/toc.2015.v011a017

[4] Abbe, E.; Sandon, C. Community detection in general stochastic block models: fundamental limits and efficient algorithms for recovery. *2015 IEEE 56th Annual Symposium on Foundations of Computer Science—FOCS 2015*, 670–688. IEEE Computer Society, Los Alamitos, Calif., 2015. doi:10.1109/FOCS.2015.47

[5] Abbe, E.; Sandon, C. Detection in the stochastic block model with multiple clusters: proof of the achievability conjectures, acyclic BP, and the information-computation gap. Preprint, 2015. arXiv:1512.09080 [math.PR]

[6] Abbe, E.; Sandon, C. Recovering communities in the general stochastic block model without knowing the parameters. *Advances in Neural Information Processing Systems (NIPS) 28*, 676–684. Edited by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, 2015.

[7] Abbe, E.; Sandon, C. Achieving the KS threshold in the general stochastic block model with linearized acyclic belief propagation. *Advances in Neural Information Processing Systems 29*, 1334–1342. Edited by D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett. Curran Associates, 2016.

[8] Abbe, E.; Sandon, C. Crossing the KS threshold in the stochastic block model with information theory. *2016 IEEE International Symposium on Information Theory (ISIT)*, 840–844. IEEE, 2016. doi:10.1109/ISIT.2016.7541417

[9] Achlioptas, D.; Naor, A. The two possible values of the chromatic number of a random graph. *Ann. of Math. (2)* **162** (2005), no. 3, 1335–1351. doi:10.4007/annals.2005.162.1335

[10] Amini, A.; Levina, E. On semidefinite relaxations for the block model. Preprint, 2014. arXiv:1406.5647 [cs.LG]

[11] Ball, B.; Karrer, B.; Newman, M. E. J. An efficient and principled method for detecting communities in networks. *Phys. Rev. E* **84** (2011), no. 3, 036103. doi:10.1103/PhysRevE.84.036103

[12] Bandeira, A. S. Random Laplacian matrices and convex relaxations. Preprint, 2015. arXiv:1504.03987.

[13] Banks, J.; Moore, C. Information-theoretic thresholds for community detection in sparse networks. Preprint, 2016. arXiv:1601.02658 [math.PR]

[14] Banks, J.; Moore, C.; Neeman, J.; Netrapalli, P. Information-theoretic thresholds for community detection in sparse networks. *29th Annual Conference on Learning Theory*, 383–416. *Journal of Machine Learning Research Workshop and Conference Proceedings* **49** (2016).

[15] Berthet, Q.; Rigollet, P. Optimal detection of sparse principal components in high dimension. *Ann. Statist.* **41** (2013), no. 4, 1780–1815. doi:10.1214/13-AOS1127

[16] Bhattacharyya, S.; Bickel, P. J. Community detection in networks using graph distance. Preprint, 2014. arXiv:1401.3915 [stat.ML]

[17] Bickel, P. J.; Chen, A. A nonparametric view of network models and Newman-Girvan and other modularities. *Proc. Natl. Acad. Sci.* **106** (2009), no. 50, 21068–21073. doi:10.1073/pnas.0907096106

[18] Bollobás, B.; Janson, S.; Riordan, O. The phase transition in inhomogeneous random graphs. *Random Structures Algorithms* **31** (2007), no. 1, 3–122. doi:10.1002/rsa.20168

[19] Boppana, R. Eigenvalues and graph bisection: An average-case analysis. *28th Annual Symposium on Foundations of Computer Science*, 280–285.IEEE, 1987. doi:10.1109/SFCS.1987.22

[20] Bordenave, C.; Lelarge, M.; Massoulié, L. Non-backtracking spectrum of random graphs: community detection and non-regular Ramanujan graphs. *2015 IEEE 56th Annual Symposium on Foundations of Computer Science—FOCS 2015*, 1347–1357. IEEE Computer Society, Los Alamitos, Calif., 2015. doi:10.1109/FOCS.2015.86

[21] Borgs, C.; Chayes, J.; Smith, A. Private graphon estimation for sparse graphs. *Advances in Neural Information Processing Systems 28*, 1369–1377. Edited by C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett. Curran Associates, 2015.

[22] Bui, T. N.; Chaudhuri, S.; Leighton, F. T.; Sipser, M. Graph bisection algorithms with good average case behavior. *Combinatorica* **7** (1987), no. 2, 171–191. doi:10.1007/BF02579448

[23] Cabreros, I.; Abbe, E.; Tsirigos, A. Detecting community structures in Hi-C genomic data. *2016 Annual Conference on Information Science and Systems (CISS)*, 584–589. IEEE, 2016.

[24] Carson, T.; Impagliazzo, R. Hill-climbing finds random planted bisections. *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms (Washington, DC, 2001)*, 903–909. SIAM, Philadelphia, 2001.

[25] Cayley, A. *Collected mathematical papers*. Cambridge University Press, Cambridge, 1889.

[26] Chen, Y.; Sanghavi, S.; Xu, H. Clustering sparse graphs. *IEEE Transactions on Information Theory* **60** (2014), no. 10, 6440-6455. doi:10.1109/TIT.2014.2346205

[27] Chen, Y.; Xu, J. Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. *J. Mach. Learn. Res.* **17** (2016), Paper No. 27, 57 pp.

[28] Chen, J.; Yuan, B. Detecting functional modules in the yeast protein-protein interaction network. *Bioinformatics* **22** (2006), no. 18, 2283–2290. doi:10.1093/bioinformatics/btl370

[29] Chin, P.; Rao, A.; Vu, V. Stochastic block model and community detection in the sparse graphs: a spectral algorithm with optimal rate of recovery. Preprint, 2015. arXiv:1501.05021 [cs.DS]

[30] Choi, D. S.; Wolfe, P. J.; Airoldi, E. M. Stochastic blockmodels with a growing number of classes. *Biometrika* **99** (2012), no. 2, 273–284. doi:10.1093/biomet/asr053

[31] Cline, M.; Smoot, M.; Cerami, E.; Kuchinsky, A.; Landys, N.; Workman, C.; Christmas, R.; Avila-Campilo, I.; Creech, M.; Gross, B.; Hanspers, K.; Isserlin, R.; Kelley, R.; Killcoyne, S.; Lotia, S.; Maere, S.; Morris, J.; Ono, K.; Pavlovic, V.; Pico, A.; Vailaya, A.; Wang, P.; Adler, A.; Conklin, B.; Hood, L.; Kuiper, M.; Sander, C.; Schmulevich, I.; Schwikowski, B.; Warner, G. J.; Ideker, T.; Bader, G. Integration of biological networks and gene expression data using Cytoscape. *Nature Protocols* **2** (2007), no. 10, 2366–2382. doi:10.1038/nprot.2007.324

[32] Coja-Oghlan, A. Graph partitioning via adaptive spectral techniques. *Combin. Probab. Comput.* **19** (2010), no. 2, 227–284. doi:10.1017/S0963548309990514

[33] Condon, A.; Karp, R. M. Algorithms for graph partitioning on the planted partition model. *Randomization, approximation, and combinatorial optimization (Berkeley, CA, 1999)*, 221–232. Lecture Notes in Computer Science, 1671. Springer, Berlin, 1999. doi:10.1007/978-3-540-48413-4_23

[34] Decelle, A.; Krzakala, F.; Moore, C.; Zdeborová, L. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Phys. Rev. E* **84** (2011), no. 6, 066106. doi:10.1103/PhysRevE.84.066106

[35] Deshpande, Y.; Montanari, A. Finding hidden cliques of size $\sqrt{N/e}$ in nearly linear time. Preprint, 2013. arXiv:1304.7047 [math.PR]

[36] Donoho, D. L.; Maleki, A.; Montanari, A. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences* **106** (2009), no. 45, 18914–18919. doi:10.1073/pnas.0909892106

[37] Dyer, M. E.; Frieze, A. M. The solution of some random NP-hard problems in polynomial expected time. *J. Algorithms* **10** (1989), no. 4, 451–489. doi:10.1016/0196-6774(89)90001-1

[38] Erdős, P.; Rényi, A. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl.* **5** (1960), 17–61.

[39] Feldman, V.; Perkins, W.; Vempala, S. On the complexity of random satisfiability problems with planted solutions. *STOC'15—Proceedings of the 2015 ACM Symposium on Theory of Computing*, 77–86. ACM, New York, 2015.

[40] Fienberg, S. E.; Meyer, M. M.; Wasserman, S. S. Statistical analysis of multiple sociometric relations. *J. Amer. Statist. Assoc.* **80** (1985), no. 389, 51–67. doi:10.1080/01621459.1985.10477129

[41] Fortunato, S. Community detection in graphs. *Phys. Rep.* **486** (2010), no. 3-5, 75–174. doi:10.1016/j.physrep.2009.11.002

[42] Gao, C.; Ma, Z.; Zhang, A. Y.; Zhou, H. H. Achieving optimal misclassification proportion in stochastic block model. Preprint, 2015. arXiv:1505.03772 [math.ST]

[43] Goldenberg, A.; Zheng, A. X.; Fienberg, S. E.; Airoldi, E. M. A survey of statistical network models. *Foundations and Trends in Machine Learning* **2** (2010), no. 2, 129–233. doi:10.1561/2200000005

[44] Guédon, O.; Vershynin, R. Community detection in sparse networks via Grothendieck's inequality. *Probab. Theory Related Fields* **165** (2016), no. 3-4, 1025–1049. doi:10.1007/s00440-015-0659-z

[45] Hajek, B.; Wu, Y.; Xu, J. Achieving exact cluster recovery threshold via semidefinite programming. Preprint, 2014. arXiv:1412.6156 [stat.ML]

[46] Hajek, B.; Wu, Y.; Xu, J. Recovering a hidden community beyond the spectral limit in $O(|E|\log^*|V|)$ time. Preprint, 2015. arXiv:1510.02786 [stat.ML]

[47] Hashimoto, K.-I. Zeta functions of finite graphs and representations of p-adic groups. *Automorphic forms and geometry of arithmetic varieties*, 211–280. Advanced Studies in Pure Mathematics, 15. Academic Press, Boston, 1989.

[48] Heimlicher, S.; Lelarge, M.; Massoulié, L. Community detection in the labelled stochastic block model. Preprint, 2012. arXiv:1209.2910 [cs.SI]

[49] Holland, P. W.; Laskey, K. B.; Leinhardt, S. Stochastic blockmodels: first steps. *Social Networks* **5** (1983), no. 2, 109–137. doi:10.1016/0378-8733(83)90021-7

[50] Horton, M. D.; Stark, H. M.; Terras, A. A. What are zeta functions of graphs and what are they good for? *Quantum graphs and their applications*, 173–189. Contemporary Mathematics, 415. American Mathematical Society, Providence, R.I., 2006. doi:10.1090/conm/415/07868

[51] Jerrum, M.; Sorkin, G. B. The Metropolis algorithm for graph bisection. *Discrete Appl. Math.* **82** (1998), no. 1-3, 155–175. doi:10.1016/S0166-218X(97)00133-9

[52] Karrer, B.; Newman, M. E. J. Stochastic blockmodels and community structure in networks. *Phys. Rev. E (3)* **83** (2011), no. 1, 016107, 10 pp. doi:10.1103/PhysRevE.83.016107

[53] Krzakala, F.; Moore, C.; Mossel, E.; Neeman, J.; Sly, A.; Zdeborová, L.; Zhang, P. Spectral redemption in clustering sparse networks. *Proc. Natl. Acad. Sci. USA* **110** (2013), no. 52, 20935–20940. doi:10.1073/pnas.1312486110

[54] Linden, G.; Smith, B.; York, J. Amazon.com recommendations: item-to-item collaborative filtering. *IEEE Internet Computing* **7** (2003), no. 1, 76–80. doi:10.1109/MIC.2003.1167344

[55] Massoulié, L. Community detection thresholds and the weak Ramanujan property. *STOC'14—Proceedings of the 2014 ACM Symposium on Theory of Computing*, 694–703. ACM, New York, 2014.

[56] McSherry, F. Spectral partitioning of random graphs. *42nd IEEE Symposium on Foundations of Computer Science (Las Vegas, NV, 2001)*, 529–537. IEEE Computer Society, Los Alamitos, Calif., 2001.

[57] Moitra, A.; Perry, W.; Wein, A. S. How robust are reconstruction thresholds for community detection? *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, 828–841. ACM, New York, 2016. doi:10.1145/2897518.2897573

[58] Montanari, A. Finding one community in a sparse graph. *J. Stat. Phys.* **161** (2015), no. 2, 273–299. doi:10.1007/s10955-015-1338-2

[59] Montanari, A.; Sen, S. Semidefinite programs on sparse random graphs and their application to community detection. *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, 814–827. ACM, New York, 2016. doi:10.1145/2897518.2897548

[60] Mossel, E.; Neeman, J.; Sly, A. A proof of the block model threshold conjecture. Preprint, 2013. arXiv:1311.4115 [math.PR]

[61] Mossel, E.; Neeman, J.; Sly, A. Reconstruction and estimation in the planted partition model. *Probab. Theory Related Fields* **162** (2015), no. 3-4, 431–461. doi:10.1007/s00440-014-0576-6

[62] Mossel, E.; Neeman, J.; Sly, A. Consistency thresholds for binary symmetric block models. *Electronic Journal of Probability* **21** (2016), no. 21, 1–24. doi:10.1214/16-EJP4185

[63] Mossel, E.; Peres, Y. Information flow on trees. *Ann. Appl. Probab.* **13** (2003), no. 3, 817–844. doi:10.1214/aoap/1060202828

[64] Murphy, K. P.; Weiss, Y.; Jordan, M. I. Loopy belief propagation for approximate inference: an empirical study. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, 467–475. Morgan Kaufmann, San Francisco, 1999.

[65] Neeman, J.; Netrapalli, P. Non-reconstructability in the stochastic block model. Preprint, 2014. arXiv:1404.6304 [math.PR]

[66] Newman, M. E. J.; Watts, D. J.; Strogatz, S. H. Random graph models of social networks. *Proc. Natl. Acad. Sci. USA* **99**, 2566–2572. doi:10.1073/pnas.012582999

[67] Rényi, A. Some remarks on the theory of trees. *Magyar Tud. Akad. Mat. Kutató Int. Közl.* **4** (1959), 73–85.

[68] Rohe, K.; Chatterjee, S.; Yu, B. Spectral clustering and the high-dimensional stochastic blockmodel. *Ann. Statist.* **39** (2011), no. 4, 1878–1915. doi:10.1214/11-AOS887

[69] Saade, A.; Krzakala, F.; Lelarge, M.; Zdeborová, L. Spectral detection in the censored block model. *2015 IEEE International Symposium on Information Theory (ISIT)*, 1184–1188. doi:10.1109/ISIT.2015.7282642. IEEE, 2015.

[70] Saade, A.; Krzakala, F.; Zdeborová, L. Spectral clustering of graphs with the Bethe Hessian. *Advances in Neural Information Processing Systems 27*, 406–414. Curran Associates, 2014.

[71] Shi, J.; Malik, J. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** (1997), no. 8, 888–905. IEEE, 2002. doi:10.1109/34.868688

[72] Sly, A. Reconstruction for the Potts model. *STOC'09—Proceedings of the 2009 ACM International Symposium on Theory of Computing*, 581–590. ACM, New York, 2009.

[73] Snijders, T. A. B.; Nowicki, K. Estimation and prediction for stochastic blockmodels for graphs with latent block structure. *J. Classification* **14** (1997), no. 1, 75–100. doi:10.1007/s003579900004

[74] Sorlie, T.; Perou, C.; Tibshirani, R.; Aas, T.; Geisler, S.; Johnsen, H.; Hastie, T.; Eisen, M.; van de Rijn, M.; Jeffrey, S.; Thorsen, T.; Quist, H.; Matese, J.; Brown, P.; Botstein, D.; Lonning, P.; Borresen-Dale, A. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proc. Natl. Acad. Sci* **98** (2001), no. 19, 10869–10874. doi:10.1073/pnas.191367098

[75] Vu, V. A simple SVD algorithm for finding hidden partitions. Preprint, 2014. arXiv:1404.3918 [math.CO]

[76] Wang, Y. J.; Wong, G. Y. Stochastic blockmodels for directed graphs. *J. Amer. Statist. Assoc.* **82** (1987), no. 397, 8–19.

[77] White, H. C.; Boorman, S. A.; Breiger, R. L. Social structure from multiple networks. *Am. J. Sociol.* **81** (1976), 730–780.

[78] Yun, S.-Y.; Proutiere, A. Optimal cluster recovery in the labeled stochastic block model. Preprint, 2015. arXiv:1510.05956

EMMANUEL ABBE
Princeton University
Fine Hall 212, Washington Street
Princeton, NJ 08544
USA
E-mail: eabbe@princeton.edu

COLIN SANDON
Princeton University
Department of Mathematics
35 University Place, Room 32
Princeton, NJ 08540
USA
E-mail: sandon@princeton.edu

Received September 2016.