2015 IEEE 56th Annual Symposium on Foundations of Computer Science

Community detection in general stochastic block models: fundamental limits and efficient algorithms for recovery

Emmanuel Abbe
PACM and EE Department
Princeton University
Princeton, NJ, USA
Email: eabbe@princeton.edu

Colin Sandon
Mathematics Department
Princeton University
Princeton, NJ, USA
Email: sandon@princeton.edu

Abstract

New phase transition phenomena have recently been discovered for the stochastic block model, for the special case of two non-overlapping symmetric communities. This gives raise in particular to new algorithmic challenges driven by the thresholds. This paper investigates whether a general phenomenon takes place for multiple communities, without imposing symmetry.

In the general stochastic block model SBM(n, p, W), n vertices are split into k communities of relative size $\{p_i\}_{i\in[k]}$, and vertices in community i and j connect independently with probability $\{W_{i,j}\}_{i,j\in[k]}$. This paper investigates the partial and exact recovery of communities in the general SBM (in the constant and logarithmic degree regimes), and uses the generality of the results to tackle overlapping communities.

The contributions of the paper are: (i) an explicit characterization of the recovery threshold in the general SBM in terms of a new f-divergence function D_+, which generalizes the Hellinger and Chernoff divergences, and which provides an operational meaning to a divergence function analog to the KL-divergence in the channel coding theorem, (ii) the development of an algorithm that recovers the communities all the way down to the optimal threshold and runs in quasi-linear time, showing that exact recovery has no information-theoretic to computational gap for multiple communities, (iii) the development of an efficient algorithm that detects communities in the constant degree regime with an explicit accuracy bound that can be made arbitrarily close to 1 when a prescribed signal-to-noise ratio (defined in term of the spectrum of $\mathrm{diag}(p)W$) tends to infinity.

Keywords

Community detection, stochastic block models, phase transitions, clustering algorithms, information measures, graph-based codes.

I. Introduction

Detecting communities (or clusters) in graphs is a fundamental problem in computer science and machine learning. This applies to a large variety of complex networks in social sciences and biology, as well as to data sets engineered as networks via similarity graphs, where one often attempts to get a first impression on the data by trying to identify groups with similar behavior. In particular, finding communities allows one to find like-minded people in social networks [1], [2], to improve recommendation systems [3], [4], to segment or classify images [5], [6], to detect protein complexes [7], [8], to find genetically related sub-populations [9], [10], or to discover new tumor subclasses [11].

While a large variety of community detection algorithms have been deployed in the past decades, understanding the fundamental limits of community detection and establishing rigorous benchmarks for algorithms remains a major challenge. Significant progress has recently been made for the stochastic block model, but mainly for the special case of two non-overlapping communities. The goal of this paper is to establish the fundamental limits of recovering communities in general stochastic block models, with multiple (possibly overlapping) communities. We first provide some motivations behind these questions.

Probabilistic network models can be used to model real networks [12], to study the average-case complexity of NP-hard problems on graphs (such as min-bisection or max-cut [13], [14], [15], [16]), or to set benchmarks for clustering algorithms with well defined ground truth. In particular, the latter holds irrespective of how exactly the model fits the data sets, and is a crucial aspect in community detection as a vast majority of algorithms are based on heuristics and no ground truth is typically available in applications. This is in particular a well known challenge for Big Data problems where one cannot manually determine the quality of the clusters [17].

Evaluating the performance of algorithms on models is, however, non-trivial. In some regimes, most reasonable algorithms may succeed, while in others, algorithms may be doomed to fail due to computational barriers. Thus, an important question is to characterize the regimes where the clustering tasks can be solved efficiently or information-theoretically. In particular, models may benefit from asymptotic phase transition phenomena, which, in addition to being mathematically interesting, allow location of the bottleneck regimes to benchmark algorithms. Such phenomena are commonly used in coding theory

0272-5428/15 $31.00 © 2015 IEEE
DOI 10.1109/FOCS.2015.47

(with the channel capacity [18]), or in constraint satisfaction problems (with the SAT thresholds, see [19] and references therein).

Recently, similar phenomena have been identified for the stochastic block model (SBM), one of the most popular network models exhibiting community structures [20], [21], [22], [23], [24], [25]. The model[1] was first proposed in the 80s [20] and received significant attention in the mathematics and computer science literature [14], [13], [26], [27], [15], [28], [29], as well as in the statistics and machine learning literature [30], [24], [31], [32]. The SBM puts a distribution on n-vertices graphs with a hidden (or planted) partition of the nodes into k communities. Denoting by p_i, $i \in [k]$, the relative size of each community, and assuming that a pair of nodes in communities i and j connects independently with probability $W_{i,j}$, the SBM can be defined by the triplet (n, p, W), where p is a probability vector of dimension k and W a $k \times k$ symmetric matrix with entries in $[0, 1]$.

The SBM recently came back at the center of the attention at both the practical level, due to extensions allowing overlapping communities [33] that have proved to fit well real data sets in massive networks [34], and at the theoretical level due to new phase transition phenomena discovered for the two-community case [35], [36], [37], [38], [39], [40]. To discuss these phenomena, we need to first introduce the recovery requirements (formal definitions are in Section II):

- **Weak recovery** (also called detection). This only requires the algorithm to output a partition of the nodes which is positively correlated with the true partition (whp[2]). Note that weak recovery is relevant in the fully symmetric case where all nodes have identical average degree,[3] since otherwise weak recovery can be trivially solved. If the model is perfectly symmetric, like the SBM with two equally-sized clusters having the same connectivity parameters, then weak recovery is non-trivial. Full symmetry may not be representative of reality, but it sets analytical and algorithmic challenges. The weak-recovery threshold for two symmetric communities was achieved efficiently in [37], [38], settling a conjecture established in [36]. The case with more than two communities remains open.
- **Partial recovery.** One may ask for the finer question of *how much* can be recovered about the communities. For a given set of parameters of the block model, finding the proportion of nodes (as a function of p and W) that can be correctly recovered (whp) is an open problem. Obtaining a closed form formula for this question is unlikely, even in the symmetric case with two communities. Partial results were obtained in [41] for two symmetric communities, but the general problem remains open even for determining scaling laws. One may also consider the special case of partial recovery where only an $o(n)$ fraction of nodes is allowed to be mis-classified (whp), called almost exact recovery or weak consistency, but no sharp phase transition is to be expected for this requirement (when parameters differ at the first order).
- **Exact recovery** (also called recovery or strong consistency.) Finally, one may ask for the regimes for which an algorithm can recover the entire clusters (whp). This is non-trivial for both symmetric and asymmetric parameters. One can also study "partial-exact-recovery," namely, which communities can be exactly recovered. While exact recovery has been the main focus in the literature for the past decades (see table in Section V), the phase transition for exact recovery was only obtained last year for the case of two symmetric communities [39], [40]. The case with more than two communities remains open.

This paper addresses items 2 and 3 for the general stochastic block model. Note that the above questions naturally require studying different regimes for the parameters. Weak recovery requires the edge probabilities to be $\Omega(1/n)$, in order to have many vertices in all but one community to be non-isolated (i.e., a giant component in the symmetric case), and recovery requires the edge probabilities to be $\Omega(\ln(n)/n)$, in order to have all vertices in all but one community to be non-isolated (i.e., a connected graph in the symmetric case). The difficulty is to understand how much more is needed in order to weakly or exactly recover the communities. In particular, giants and connectivity have phase transitions, and similar phenomena may be expected for weak and exact recovery.

Note that these regimes are not only rich mathematically, but are also relevant for applications, as a vast collection of real networks ranging from social (LinkedIn, MSN), collaborative (movies, arXiv), or biological (yeast) networks and more were shown to be sparse [42], [43]. Note however that the average degree is typically not small in real networks, and it seems hard to distinguish between a large constant or a slowly growing function. Both regimes are of interest to us.

Finally, there is an important distinction to be made between the information-theoretic thresholds, which do not put constraints on the algorithm's complexity, and the computational thresholds, which require polynomial-time algorithms. In the case of two symmetric communities, the information-theoretic and computational thresholds were proved to be the same

[1] See Section V for further references.

[2] whp means with high probablity, i.e., with probability $1 - o_n(1)$ when the number of nodes in the graph diverges.

[3] At least for the case for communities having linear size. One may otherwise define stronger notions of weak recovery that apply to non-symmetric cases.

for weak recovery [37], [38] and exact recovery [39], [40]. A gap is conjectured to take place for weak recovery for more than 4 communities [36]. No conjectures were made for exact recovery for multiple communities.

This paper focuses on partial and exact recovery (items 2 and 3) for the general stochastic block model with linear size communities, and uses the generality of the results to address overlapping communities (see Section IV). Recall that for the case of two communities, if

$$q_{in} = a\ln(n)/n,$$
$$q_{out} = b\ln(n)/n,$$

are respectively the intra- and extra-cluster probabilities, with $a > b > 0$, then exact recovery is possible if and only if

$$\sqrt{a} - \sqrt{b} \geq \sqrt{2}, \tag{1}$$

and this is efficiently achievable. However, there is currently no general insight regarding equation (1), as it emerges from estimating a tail event for Binomial random variable specific to the case of two symmetric communities. Moreover, no results are known to prove partial recovery bounds for more than two communities (recent progress where made in [44]). This represents a limitation of the current techniques, and an impediment to progress towards more realistic network models that may have overlapping communities, and for which analytical results are currently unknown.[4] We next present our effort towards such a general treatment.

II. Results

The main advances of this paper are:

(i) an algorithm (`Sphere-comparison`) that detects communities in the general SBM with $W = Q/n$ with an explicit accuracy guarantee: introducing a signal-to-noise ratio defined by the ratio $|\lambda_{\min}|^2/\lambda_{\max}$ where $\lambda_{\min}$ and $\lambda_{\max}$ are respectively the smallest[5] and largest eigenvalue of $\mathrm{diag}(p)Q$, it is shown that when the SNR diverges, the algorithm accuracy tends to 1 (exponentially fast) and its complexity becomes quasi-linear, i.e., $o(n^{1+\varepsilon})$, for all $\varepsilon > 0$,

(ii) an explicit characterization of the recovery threshold in the general SBM in terms of a divergence function D_+, which provides a new operational meaning to a divergence analog to the KL-divergence in the channel coding theorem (see Section II-C), and which allows determining which communities can be recovered by solving a packing problem in the appropriate embedding,

(iii) a quasi-linear time algorithm (`Degree-profiling`) that solves exact recovery in the regime $W = Q\ln(n)/n$ whenever it is information-theoretically solvable[6], showing in particular that there is no information-theoretic to computational gap for exact recovery with multiple communities (as opposed to the conjectures on weak recovery). Note that the algorithm replicates statistically the performance of maximum-likelihood (which is NP-hard in the worst-case) with an optimal (i.e., quasi-linear) complexity. In particular, it improves significantly on the SDPs developed for symmetric communities (see Section V) both in terms of generality and complexity.

A. Definitions and terminologies

The general stochastic block model, SBM(n, p, W), is a random graph ensemble defined as follows:

- n is the number of vertices in the graph, $V = [n]$ denotes the vertex set.
- Each vertex $v \in V$ is assigned independently a hidden (or planted) label σ_v in $[k]$ under a probability distribution $p = (p_1, \ldots, p_k)$ on $[k]$. That is, $\mathbb{P}\{\sigma_v = i\} = p_i$, $i \in [k]$. We also define $P = \mathrm{diag}(p)$.
- Each (unordered) pair of nodes $(u, v) \in V \times V$ is connected independently with probability W_{σ_u,σ_v}, where W_{σ_u,σ_v} is specified by a symmetric $k \times k$ matrix W with entries in $[0, 1]$.

The above gives a distribution on n-vertex graphs. Note that $G \sim SBM(n, p, W)$ denotes a random graph drawn under this model, without the hidden (or planted) clusters (i.e., the labels σ_v) revealed. The goal is to recover these labels by observing only the graph.

This paper focuses on p independent of n (the communities have linear size), W dependent on n such that the average node degrees are either constant or logarithmically growing and k fixed. These assumptions on p and k could be relaxed, for example to slowly growing k, but we leave this for future work. As discussed in the introduction, the above regimes for W are both motivated by applications and by the fact that interesting mathematical phenomena take place in these regimes. For convenience, we attribute specific notations for the model in these regimes:

[4]Different models than the SBM allowing for overlapping communities have been studied for example in [45].

[5]The smallest eigenvalue of $\mathrm{diag}(p)Q$ is the one with least magnitude.

[6]Assuming that the entries of Q_{ij} are non-zero — see Remark 1 for zero entries.

Definition 1. *For a symmetric matrix $Q \in \mathbb{R}_+^{k\times k}$,*

- $\mathbb{G}_1(n,p,Q)$ *denotes SBM*$(n,p,Q/n)$*,*
- $\mathbb{G}_2(n,p,Q)$ *denotes SBM*$(n,p,\ln(n)Q/n)$*.*

We now discuss the recovery requirements.

Definition 2. *(Partial recovery.) An algorithm recovers or detects communities in SBM(n,p,W) with an accuracy of $\alpha \in [0,1]$, if it outputs a labelling of the nodes $\{\sigma'(v), v \in V\}$, which agrees with the true labelling σ on a fraction α of the nodes with probability $1-o_n(1)$. The agreement is maximized over relabellings of the communities.*

Definition 3. *(Exact recovery.) Exact recovery is solvable in SBM(n,p,W) for a community partition $[k] = \sqcup_{s=1}^t A_s$, where A_s is a subset of $[k]$, if there exists an algorithm that takes $G \sim SBM(n,p,W)$ and, with probability $1-o_n(1)$, assigns to every node in G an element of $\{A_1,\dots,A_t\}$ that contains its true community[7]. Exact recovery is solvable in SBM(n,p,W) if it is solvable for the partition of $[k]$ into k singletons, i.e., all communities can be recovered. The problem is solvable information-theoretically if there exists an algorithm that solves it, and efficiently if the algorithm runs in polynomial-time in n.*

Note that exact recovery for the partition $[k] = \{i\} \sqcup ([k] \setminus \{i\})$ is equivalent to extracting community i. In general, recovering a partition $[k] = \sqcup_{s=1}^t A_s$ is equivalent to merging the communities that are in a common subset A_s and recovering the merged communities. Note also that exact recovery in SBM(n,p,W) requires the graph not to have vertices of degree 0 in multiple communities (with high probability). In the symmetric case, this amounts to asking for connectivity. Therefore, for exact recovery, we will focus below on W scaling like $\frac{\ln(n)}{n}Q$ where Q is a fixed matrix, i.e., on the $\mathbb{G}_2(n,p,Q)$ model.

B. Main results

We next present our main results and algorithms for partial and exact recovery in the general SBM. We present slightly simplified versions in this section, and provide full statements in Sections 6 and 7 of [46].

The CH-embedding and exact recovery. We explain first how to identify the communities that can be extracted from a graph drawn under $\mathbb{G}_2(n,p,Q)$. Define first the community profile of community $i \in [k]$ by the vector

$$\theta_i := (PQ)_i \in \mathbb{R}_+^k, \tag{2}$$

i.e., the i-th column of the matrix $\mathrm{diag}(p)Q$. Note that $\|\theta_i\|_1 \log(n)$ gives the average degree of a node in community i. Two communities having the same community profile cannot be distinguished, in that the random graph distribution is invariant under any permutation of the nodes in these communities. Intuitively, one would expect that the further "apart" the community profiles are, the easier it should be to distinguish the communities. The challenge is to quantify what "apart" means, and whether there exists a proper distance notion to rely on. We found that the following function gives the appropriate notion,

$$\begin{aligned} D_+ : \mathbb{R}_+^k \times \mathbb{R}_+^k &\to \mathbb{R}_+ \\ (\theta_i,\theta_j) &\mapsto D_+(\theta_i,\theta_j) = \max_{t\in[0,1]} \sum_{x\in[k]} \left(t\theta_i(x) + (1-t)\theta_j(x) - \theta_i(x)^t\theta_j(x)^{1-t}\right). \end{aligned} \tag{3}$$

For a fixed t, the above is a so-called f-divergence (obtained for $f(x) = 1-t+tx-x^t$), a family of divergences generalizing the KL-divergence (relative entropy) defined in [47], [48], [49] and used in information theory and statistics. As explained in Section II-C, D_+ can be viewed as a generalization of the Hellinger divergence (obtained for $t = 1/2$) and the Chernoff divergence. We therefore call D_+ the Chernoff-Hellinger (CH) divergence. Note that for the case of two symmetric communities, $D_+(\theta_1,\theta_2) = \frac{1}{2}(\sqrt{a}-\sqrt{b})^2$, recovering the result in [39], [40].

To determine which communities can be recovered, partition the community profiles into the largest collection of disjoint subsets such that the CH-divergence among these subsets is at least 1 (where the H-divergence between two subsets of profiles is the minimum of the H-divergence between any two profiles in each subset). We refer to this as the *finest partition* of the communities. Note that it is the set of connected components in the graph where each community is a vertex, and two communities are adjacent if and only if they have CH-divergence less than 1. Figure 1 illustrates this partition. The theorem below shows that this is indeed the most granular partition that can be recovered about the communities, in particular, it characterizes the information-theoretic and computational threshold for exact recovery.

Theorem 1. *(See Theorem 6 in [46]) Let Q be a $k\times k$ matrix with nonzero entries, $p \in (0,1)^k$ with $\sum p = 1$.*

[7]This is again up to relabellings of the communities.

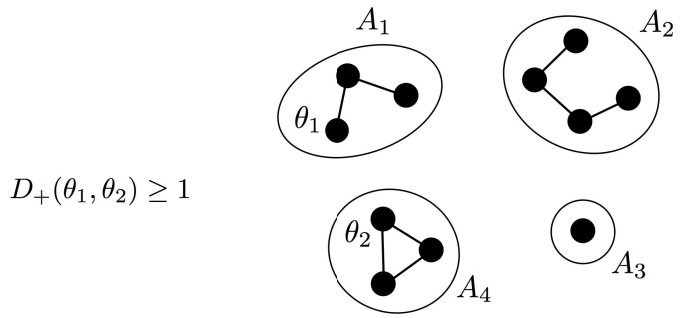


Figure 1: Finest partition: To determine which communities can be recovered in the SBM $\mathbb{G}_2(n, p, Q)$, embed each community with its community profile $\theta_i = (PQ)_i$ in $\mathbb{R}_+^k$ and find the partition of $\theta_1, \ldots, \theta_k$ into the largest number of subsets that are at CH-divergence at least 1 from each other.

- *Exact recovery is information-theoretically solvable in the stochastic block model $\mathbb{G}_2(n, p, Q)$ for a partition $[k] = \sqcup_{s=1}^t A_s$ if and only if for all i and j in different subsets of the partition,*

$$D_+((PQ)_i, (PQ)_j) \geq 1, \tag{4}$$

  In particular, exact recovery is information-theoretically solvable in $\mathbb{G}_2(n, p, Q)$ if and only if $\min_{i,j\in[k],i\neq j} D_+((PQ)_i||(PQ)_j) \geq 1$.
- *The* `Degree-profiling` *algorithm (see Section III-B) recovers the finest partition with probability $1-o_n(1)$ and runs in $o(n^{1+\epsilon})$ time for all $\epsilon > 0$. In particular, exact recovery is efficiently solvable whenever it is information-theoretically solvable.*

Let us stress that the regime considered in the above theorem with $W = Q\log(n)/n$ is the "bottleneck regime", i.e., unless the entries of W are of the exact same order, the theorem characterizes when exact recovery is possible or not. For example, in the dense regime where all the entires of W are different constants, the condition of the theorem becomes extremal and trivially verified. In addition, this extends McSherry [29] unless $p_{in} \sim p_{out}$.

To achieve this result we rely on a two step procedure, via a "graph-splitting" technique. First an algorithm is developed on a random sub-graph to recover all but a vanishing fraction of nodes — this is the main focus of our partial recovery result next discussed — and then a procedure is used to "clean up" the leftover graphs using the node degrees of the preliminary classification. This turns out to be much more efficient than aiming for an algorithm that directly achieves exact recovery. This strategy was already used in [39] for the two-community case, and similar ideas appeared in earlier works such as [13], [50]. The problem is much more involved here as no algorithm is known to ensure partial recovery in the general SBM (with partial results in [35]), and as classifying the nodes based on their degrees requires solving a general hypothesis testing problem for the degree-profiles in the SBM (rather than evaluating tail events of Binomial distributions). The latter part reveals the CH-divergence as the threshold for exact recovery. We next present our result for partial recovery.

Remark 1. *If $Q_{ij} = 0$ for some i and j then the theorem above still hold, except that if for all i and j in different subsets of the partition,*

$$D_+((PQ)_i, (PQ)_j) \geq 1, \tag{5}$$

but there exist i and j in different subsets of the partition such that $D_+((PQ)_i, (PQ)_j) = 1$ and $((PQ)_{i,k} \cdot (PQ)_{j,k} \cdot ((PQ)_{i,k} - (PQ)_{j,k}) = 0$ for all k, then the optimal algorithm will have an assymptotically constant failure rate. The recovery algorithm also needs to be modified to accomodate 0's in Q.

Partial recovery. We obtain an algorithm that recovers the communities with an accuracy bound that tends to 1 when the average degree of the nodes gets large, and which runs in quasi-linear time.

Theorem 2. *[See Theorem 4 in [46]] Given any $k \in \mathbb{Z}$, $p \in (0,1)^k$ with $|p| = 1$, and symmetric matrix Q with no two rows equal, let λ be the largest eigenvalue of PQ, and λ' be the eigenvalue of PQ with the smallest nonzero magnitude.*

If the following signal-to-noise ratio (SNR) ρ satisfies

$$\rho := \frac{|\lambda'|^2}{\lambda} > 4, \tag{6}$$

$$\lambda^7 < (\lambda')^8, \tag{7}$$

$$4\lambda^3 < (\lambda')^4, \tag{8}$$

then for some $\varepsilon = \varepsilon(\lambda, \lambda')$ and $C = C(p, Q) > 0$, the algorithm `Sphere-comparison` *(see Section III-A) detects with high probability communities in graphs drawn from $\mathbb{G}_1(n, p, Q)$ with accuracy*

$$1 - \frac{4ke^{-\frac{C\rho}{16k}}}{1 - e^{-\frac{C\rho}{16k}\left(\frac{(\lambda')^4}{\lambda^3} - 1\right)}}, \tag{9}$$

provided that the above is larger than $1 - \frac{\min_i p_i}{2\ln(4k)}$, and runs in $O(n^{1+\epsilon})$ time. Moreover, $\varepsilon = O(\ln(\lambda\sqrt{2}/|\lambda'|)/\ln(\lambda))$, and $C(p, \alpha Q)$ is independent of α.

We next detail what the previous theorem gives in the case of k symmetric clusters.

Corollary 1. *Consider the k-block symmetric case. In other words, $p_i = \frac{1}{k}$ for all i, and $Q_{i,j}$ is α if $i = j$ and β otherwise. The vector whose entries are all 1s is an eigenvector of PQ with eigenvalue $\frac{\alpha+(k-1)\beta}{k}$, and every vector whose entries add up to 0 is an eigenvector of PQ with eigenvalue $\frac{\alpha-\beta}{k}$. So, $\lambda = \frac{\alpha+(k-1)\beta}{k}$ and $\lambda' = \frac{\alpha-\beta}{k}$ and*

$$\rho > 4 \quad \Leftrightarrow \quad \frac{(\alpha-\beta)^2}{k(\alpha+(k-1)\beta)} > 4, \tag{10}$$

and as long as $k(\alpha+(k-1)\beta)^7 < (\alpha-\beta)^8$ and $4k(\alpha+(k-1)\beta)^3 < (\alpha-\beta)^4$, there exists a constant $c > 0$ (see Corollary 4 in [46] for details on c) such that `Sphere-comparison` *detects communities, and the accuracy is*

$$1 - O(e^{-c(\alpha-\beta)^2/(k(\alpha+(k-1)\beta))})$$

for sufficiently large $(\alpha-\beta)^2/(k(\alpha+(k-1)\beta))$.

Note that ρ is the SNR appearing in the conjecture on the detection threshold for multiple blocks [36], but the conjecture is that $\rho > 1$ is necessary and sufficient, so that the above gives a only sufficient condition. The following is an important consequence of the previous theorem, as it shows that `Sphere-comparison` achieves almost exact recovery when the entries of Q are amplified.

Corollary 2. *For any $k \in \mathbb{Z}$, $p \in (0,1)^k$ with $|p| = 1$, and symmetric matrix Q with no two rows equal, there exists $\epsilon(x) = O(1/\ln(x))$ and constant $c_1 > 0$ such that for all sufficiently large x there exists an algorithm (*`Sphere-comparison`*) that detects communities in graphs drawn from $\mathbb{G}_1(n, p, xQ)$ with accuracy at least $1 - O_x(e^{-c_1 x})$ in $O_n(n^{1+\epsilon(x)})$ time for all sufficiently large n.*

C. Information theoretic interpretation of the results

We give in this section an interpretation of Theorem 1 related to Shannon's channel coding theorem in information theory. At a high level, clustering the SBM is similar to reliably decoding a codeword on a channel which is non-conventional in information theory. The channel inputs are the nodes' community assignments and the channel outputs are the network edges. We next show that this analogy is more than just high-level: reliable communication on this channel is equivalent to exact recovery, and Theorem 1 shows that the "clustering capacity" is obtained from the CH-divergence of channel-kernel PQ, which is an f-divergence like the KL-divergence governing the communication capacity.

Consider the problem of transmitting a string of n k-ary information bits on a memoryless channel. Namely, let $X_1, \ldots, X_n$ be i.i.d. from a distribution p on $[k]$, the input distribution, and assume that we want to transmit those k-ary bits on a memoryless channel, whose one-time probability transition is W. This requires using a code, which embeds[8] the vector $X^n = (X_1, \ldots, X_n)$ into a larger dimension vector $U^N = (U_1, \ldots, U_N)$, the codeword ($N \geq n$), such that the corrupted version of U^N that the memoryless channel produces, say Y^N, still allows recovery of the original U^N (hence X^n) with high probability on the channel corruptions. In other words, a code design provides the map C from X^n to U^N (see Figure 2a), and a decoding map that allows recovery of X^n from Y^N with a vanishing error probability (i.e., reliable communication).

[8]This embedding is injective.

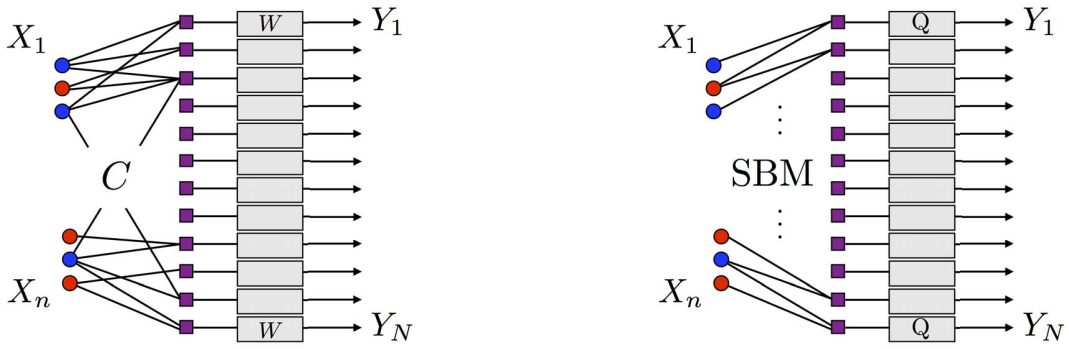
Of course, if $n = N$, the encoder C is just a one-to-one map, and there is no hope of defeating the corruptions of the channel W, unless this one is deterministic to start with. The purpose of the channel coding theorem is to quantify the best tradeoffs between n, N and the amount of randomness in W, for which one can reliably communicate. When the channel is fixed and memoryless, N can grow linearly with n, and defining the code rate by $R = n/N$, Shannon's coding theorem tells us that R is achievable (i.e., there exists an encoder and decoder that allow for reliable communication at that rate) *if and only if*

$$R < \max_p I(p, W), \tag{11}$$

where $I(p, W)$ is the mutual information of the channel W for the input distribution p, defined as

$$I(p, W) = D(p \circ W || p \times pW) = \sum_{x,y} p(x)W(y|x) \log \frac{p(x)W(y|x)}{p(x) \sum_u p(u)W(y|u)}. \tag{12}$$

Note that the channel capacity $\max_p I(p, W)$ is expressed in terms of the the Kullback-Leibler divergence (relative entropy) between the joint and product distribution of the channel.



(a) An encoder C for data transmission.

(b) The SBM encoder for network modelling.

Figure 2: Clustering over the SBM can be related to channel coding over a discrete memoryless channel, for a different type of encoder and one-time channel.

We now explain how this relates to our Theorem 1. Clustering the SBM can be cast as a decoding problem on a channel similar to the above. The n k-ary information bits X^n represent the community assignments to the n nodes in the network. As for channel coding, these are assumed to be i.i.d. under some prior distribution p on $[k]$. However, clustering has several important distinctions with coding. First of all, we do not have degree of freedom on the encoder C. The encoder is part of the model, and in the SBM C takes all possible $\binom{n}{2}$ pair of information bits. In other words, the SBM corresponds to a specific encoder which has only degree 2 on the check-nodes (the squared nodes in Figure 2b) and for which $N = \binom{n}{2}$. Next, as in channel coding, the SBM assumes that the codeword is corrupted from a memoryless channel, which takes the two selected k-ary bits and maps them to an edge variable (presence or absence of edge) with a channel W defined by the connectivity matrix:

$$W(1|x_1, x_2) = q_{x_1,x_2}, \tag{13}$$

$$W(0|x_1, x_2) = 1 - q_{x_1,x_2}, \tag{14}$$

where q scales with n here. Hence, the SBM can be viewed as a specific encoder on a memoryless channel defined by the connectivity matrix q. We removed half of the degrees of freedom from channel coding (i.e., the encoder and p are fixed), but the goal of clustering is otherwise similar to channel coding: design a decoding map that recovers the information k-ary bits X^n from the network Y^N with a vanishing error probability. In particular, exact recovery is equivalent to reliable communication.

A naive guess would be that some mutual information derived from the input distribution p and the channel induced from q could give the fundamental tradeoffs, as for channel coding. However, this is where the difference between coding and clustering is important. An encoder that achieves capacity in the traditional setting is typically "well spread," for example, like a random code which picks each edge in the bipartite graph of Figure 2a with probability one half. The SBM encoder, instead, is structured in a very special way, which may not be well suited for communication purposes[9]. This makes of course sense as the formation of a real network should have nothing to do with the design of an engineering system. Note also that the code rate in the SBM channel is fixed to $R = \frac{n}{\binom{n}{2}} \sim \frac{2}{n}$, which means that there is hope to still decode such a "poor" code, even on a very noisy channel.

Theorem 1 shows that indeed a similar phenomenon to channel coding takes place for clustering. Namely, there exists a notion of "capacity," governed not by KL-divergence but the CH-divergence introduced in Section II-B. The resulting capacity captures if reliable communication is possible or not. The relevant regime is for q that scales like $\ln(n)Q/n$, and the theorem says that it is possible to decode the inputs (i.e., to recover the communities) if and only if

$$1 \leq J(p, Q), \tag{15}$$

where

$$J(p, Q) = \min_{i \neq j} D_+((pQ)_i, (pQ)_j). \tag{16}$$

Note again the difference with the channel coding theorem: here we cannot optimize over p (since the community sizes are not a design parameter), and the rate R is fixed. One could change the latter requirement, defining a model where the information about the edges is only revealed at a given rate, in which case the analogy with Shannon's theorem can be made even stronger (see for example [51].)

The conclusiong is that we can characterize the fundamental limit for clustering, with a sharp transition governed by a measure of the channel "noisiness," that is related to the KL-divergence used in the channel coding theorem. This is due to the hypothesis testing procedures underneath both frameworks (see Section 7.2 in [46]). Defining

$$D_t(\mu, \nu) := \sum_{x \in [k]} \left(t\mu(x) + (1-t)\nu(x) - \mu(x)^t \nu(x)^{1-t} \right) \tag{17}$$

we have that

- D_t is an f-divergence, that is, it can be expressed as $\sum_x \nu(x) f(\mu(x)/\nu(x))$, where $f(x) = 1 - t + tx - x^t$, which is convex. The family of f-divergences were defined in [47], [48], [49] and shown to have various common properties when f is convex. Note that the KL-divergence is also an f-divergence for the convex function $f(x) = x \ln(x)$,
- $D_{1/2}(\mu, \nu) = \frac{1}{2} \|\sqrt{\mu} - \sqrt{\nu}\|_2^2$ is the Hellinger divergence (or distance), in particular, this is the maximizer for the case of two symmetric communities, recovering the expression $\frac{1}{2}(\sqrt{a} - \sqrt{b})^2$ obtained in [39], [40],
- $D_t(\mu, \nu) = t\bar{\mu} - (1-t)\bar{\nu} - e^{-D_t(\mu||\nu)}$, where $D_t(\cdot||\cdot)$ is the Rényi divergence, and the maximization over t of this divergence is the Chernoff divergence.

As a result, D_+ can be viewed as a generalization of the Hellinger and Chernoff divergences. We hence call it the Chernoff-Hellinger (CH) divergence. Theorem 1 gives hence an operational meaning to D_+ with the community recovery problem. It further shows that the limit can be efficiently achieved.

III. Proof Techniques and Algorithms

A. Partial recovery and the `Sphere-comparison` algorithm

The first key observation used to classify graphs' vertices is that if v is a vertex in a graph drawn from $\mathbb{G}_1(n, p, Q)$ then for all small r the expected number of vertices in community i that are r edges away from v is approximately $e_i \cdot (PQ)^r e_{\sigma_v}$. So, we define:

Definition 4. *For any vertex v, let $N_{r[G]}(v)$ be the set of all vertices with shortest paths in G to v of length r. We may use the notation $N_r(v)$ when there is no ambiguity about the background noise. When $N_r(v)$ is used as a vector, it refers to the vector whose i-th entry is the number of vertices in the set $N_r(v)$ that are in community i.*

Given a vertex v, one could determine e_{σ_v} given $(PQ)^r e_{\sigma_v}$ for some r, but using $N_r(v)$ to approximate that would require knowing how many of the vertices in $N_r(v)$ are in each community. There is no way to determine that exactly. However,

[9]It corresponds for example to a 2-right degree LDGM code in the case of the symmetric two-community SBM, a code typically not used for communication purposes.

given a vertex v' and appropriate integer r', we will get some information on how similar the community distribution of vertices in $N_r(v)$ is to the community distribution of vertices in $N_{r'}(v')$ by comparing these "spheres" in some way. This comparison gives us some information on how likely v and v' are to be in the same community, which can be used to help classify the vertices.

For example, in the case where $p_i = 1/k$ for all i, $Q_{i,i} = \alpha$ for all i, $Q_{i,j} = \beta$ for all $i \neq j$, and $\alpha - \beta$ is sufficiently large, $N_r(v)$ will typically have more vertices in v's community than in any other community when r is even. So, we could test whether two vertices v and v' are in the same community simply by checking whether $\frac{|N_r(v) \cap N_r(v')|}{|N_r(v)| \cdot |N_r(v')|}$ is greater than its average over all pairs of vertices for an appropriate r. We could then classify the vertices as follows. First, compute the average value of $\frac{|N_r(v) \cap N_r(v')|}{|N_r(v)| \cdot |N_r(v')|}$ (or an approximation of it). Next, find one vertex in each community, $v_1, v_2, ..., v_k$, by repeatedly picking a vertex v' at random, checking whether $\frac{|N_r(v_i) \cap N_r(v')|}{|N_r(v_i)| \cdot |N_r(v')|}$ is greater than the average for any v_i that has already been found, and adding v' to the list if it is not. Finally, for every remaining vertex in the graph, v'', conclude that v'' is in the same community as the v_i that maximizes the value of $\frac{|N_r(v_i) \cap N_r(v'')|}{|N_r(v_i)| \cdot |N_r(v'')|}$.

In the general case, classifying vertices will require a more detailed analysis. First, let $\lambda_1, ..., \lambda_h$ be the distinct eigenvalues of PQ, ordered so that $|\lambda_1| \geq |\lambda_2| \geq ... \geq |\lambda_h| \geq 0$. Also define η so that $\eta = h$ if $\lambda_h \neq 0$ and $\eta = h - 1$ if $\lambda_h = 0$. If W_i is the eigenspace of PQ corresponding to the eigenvalue λ_i, and P_{W_i} is the projection operator on to W_i, then for any v and v' from different communities, the fact that no two columns of Q are equal implies that $PQ \cdot e_{\sigma_v} \neq PQ \cdot e_{\sigma_{v'}}$, which implies that there exists $1 \leq i \leq \eta$ such that $P_{W_i}(e_{\sigma_v}) \neq P_{W_i}(e_{\sigma_{v'}})$. Therefore, v's community can be determined from the list of $P_{W_i}(e_{\sigma_v})$ for all $1 \leq i \leq \eta$. Furthermore,

$$P_{W_i}(N_r(v)) \approx P_{W_i}((PQ)^r e_{\sigma_v}) \approx \lambda_i^r P_{W_i}(e_{\sigma_v})$$

So, v's community can typically be determined from the eigenvector decomposition of $N_r(v)$. Our ignorance of the communities of the vertices in $N_r(v)$ usually prevents us from determining this, but given v' and r' we can hope to use some comparison between $N_r(v)$ and $N_{r'}(v')$ to approximate $P_{W_i}(e_{\sigma_v}) \cdot P^{-1} P_{W_i}(e_{\sigma_{v'}})$ for each i. Unfortunately, $|N_r(v) \cap N_{r'}(v')|$ is generally not useful for this calculation because the lack of independence between which vertices in a given community are in $N_r(v)$ and which vertices in a given community are in $N_{r'}(v')$ throws its value off. Instead, we randomly assign every edge in G to a set E with probability c, and define the following.

Definition 5. *For any vertices $v, v' \in G$, $r, r' \in \mathbb{Z}$, and subset of G's edges E, let $N_{r,r'[E]}(v \cdot v')$ be the number of pairs of vertices (v_1, v_2) such that $v_1 \in N_{r[G \backslash E]}(v)$, $v_2 \in N_{r'[G \backslash E]}(v')$, and $(v_1, v_2) \in E$.*

Note that E and $G \backslash E$ are disjoint; however, G is sparse enough that even if they were generated independently a given pair of vertices would have an edge between them in both with probability $O(\frac{1}{n^2})$. So, E is approximately independent of $G \backslash E$. Thus, for any $v_1 \in N_{r[G/E]}(v)$ and $v_2 \in N_{r'[G/E]}(v')$, $(v_1, v_2) \in E$ with a probability of approximately $cQ_{\sigma_{v_1}, \sigma_{v_2}}/n$. As a result,

$$\begin{aligned} N_{r,r'}[E](v \cdot v') &\approx N_{r[G \backslash E]}(v) \cdot \frac{cQ}{n} N_{r'[G \backslash E]}(v') \\ &\approx ((1-c)PQ)^r e_{\sigma_v} \cdot \frac{cQ}{n} ((1-c)PQ)^{r'} e_{\sigma_{v'}} \\ &= c(1-c)^{r+r'} e_{\sigma_v} \cdot Q(PQ)^{r+r'} e_{\sigma_{v'}}/n \end{aligned}$$

We probably could have used a count of the non-backtracking walks of a given length between v and v' like in [38] instead of using $N_{r,r'[E]}(v \cdot v')$. However, proving that the number of non-backtracking walks is close to its expected value is difficult. Proving that $N_{r,r'[E]}(v \cdot v')$ is within a desired range is substantially easier because for any v_1 and v_2, whether or not there is an edge between v_1 and v_2 directly effects $N_r(v)$ for at most one value of r. For appropriate E, r, and r',

we have that

$$\begin{aligned}
N_{r,r'}[E](v \cdot v') &\approx c(1-c)^{r+r'} e_{\sigma_v} \cdot Q(PQ)^{r+r'} e_{\sigma_{v'}}/n \\
&= \frac{c(1-c)^{r+r'}}{n} \left(\sum_i P_{W_i}(e_{\sigma_v}) \right) \cdot Q(PQ)^{r+r'} \left(\sum_j P_{W_j}(e_{\sigma_{v'}}) \right) \\
&= \frac{c(1-c)^{r+r'}}{n} \sum_{i,j} P_{W_i}(e_{\sigma_v}) \cdot Q(PQ)^{r+r'} P_{W_j}(e_{\sigma_{v'}}) \\
&= \frac{c(1-c)^{r+r'}}{n} \sum_{i,j} P_{W_i}(e_{\sigma_v}) \cdot P^{-1}(\lambda_j)^{r+r'+1} P_{W_j}(e_{\sigma_{v'}}) \\
&= \frac{c(1-c)^{r+r'}}{n} \sum_i \lambda_i^{r+r'+1} P_{W_i}(e_{\sigma_v}) \cdot P^{-1} P_{W_i}(e_{\sigma_{v'}})
\end{aligned}$$

where the final equality holds because for all $i \neq j$,

$$\begin{aligned}
\lambda_i P_{W_i}(e_{\sigma_v}) \cdot P^{-1} P_{W_j}(e_{\sigma_{v'}}) &= (PQP_{W_i}(e_{\sigma_v})) \cdot P^{-1} P_{W_j}(e_{\sigma_{v'}}) \\
&= P_{W_i}(e_{\sigma_v}) \cdot Q P_{W_j}(e_{\sigma_{v'}}) \\
&= P_{W_i}(e_{\sigma_v}) \cdot P^{-1} \lambda_j P_{W_j}(e_{\sigma_{v'}}),
\end{aligned}$$

and since $\lambda_i \neq \lambda_j$, this implies that $P_{W_i}(e_{\sigma_v}) \cdot P^{-1} P_{W_j}(e_{\sigma_{v'}}) = 0$.

That implies that one can approximately solve for $P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_{v'}}$ given $N_{r,r'+j}(v \cdot v')$ for all $0 \leq j < \eta$. Of course, this requires r and r' to be large enough such that

$$\frac{c(1-c)^{r+r'}}{n} \lambda_i^{r+r'+1} P_{W_i}(e_{\sigma_v}) \cdot P^{-1} P_{W_i}(e_{\sigma_{v'}})$$

is large relative to the error terms for all $i \leq \eta$. At a minimum, that requires that $|(1-c)\lambda_i|^{r+r'+1} = \omega(n)$, so

$$r + r' > \log(n)/\log((1-c)|\lambda_\eta|).$$

On the flip side, one also needs

$$r, r' < \log(n)/\log((1-c)\lambda_1)$$

because otherwise the graph will start running out of vertices before one gets r steps away from v or r' steps away from v'.

Furthermore, for any v and v',

$$\begin{aligned}
0 &\leq P_{W_i}(e_{\sigma_v} - e_{\sigma_{v'}}) \cdot P^{-1} P_{W_i}(e_{\sigma_v} - e_{\sigma_{v'}}) \\
&= P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_v} - 2 P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_{v'}} + P_{W_i} e_{\sigma_{v'}} \cdot P^{-1} P_{W_i} e_{\sigma_{v'}}
\end{aligned}$$

with equality for all i if and only if $\sigma_v = \sigma_{v'}$, so sufficiently good approximations of $P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_v}, P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_{v'}}$ and $P_{W_i} e_{\sigma'_v} \cdot P^{-1} P_{W_i} e_{\sigma_{v'}}$ can be used to determine which pairs of vertices are in the same community as follows.

The Vertex_comparison_algorithm. The inputs are $(p, Q, G, v, v', r, r', E, x, c)$, where $p \in (0,1)^k$ with $\sum p_i = 1$, Q is a $k \times k$ symmetric matrix with nonnegative coefficients, G is a graph, v, v' are two vertices, r, r' are positive integers, E is a subset of G's edges, x is a positive real number, and c is a real number between 0 and 1.

The algorithm outputs a decision on whether v and v' are in the same community or not. It proceeds as follows.

(1) Use $N_{r+j,r'[E]}(v \cdot v')$, $N_{r+j,r'[E]}(v \cdot v)$, and $N_{r+j,r'[E]}(v' \cdot v')$ for $0 \leq j < \eta$ to approximate $P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_v}, P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_{v'}}$ and $P_{W_i} e_{\sigma'_v} \cdot P^{-1} P_{W_i} e_{\sigma_{v'}}$ for each $1 \leq i \leq \eta$.

(2) If the resulting approximation of $P_{W_i}(e_{\sigma_v} - e_{\sigma_{v'}}) \cdot P^{-1} P_{W_i}(e_{\sigma_v} - e_{\sigma_{v'}})$ is greater than $5(2x(\min p_j)^{-1/2} + x^2)$ for any i then conclude that v and v' are in different communities. Otherwise, conclude that v and v' are in the same community.

One could generate a reasonable classification based solely on this method of comparing vertices (with an appropriate choice of the parameters, as later detailed). However, that would require computing $N_{r,r'[E]}(v \cdot v)$ for every vertex in the

graph with fairly large $r + r'$, which would be slow. Instead, we use the fact that for any vertices v, v', and v'' with $\sigma_v = \sigma_{v'} \neq \sigma_{v''}$,

$$P_{W_i} e_{\sigma_{v'}} \cdot P^{-1} P_{W_i} e_{\sigma_{v'}} - 2 P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_{v'}} + P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_v} = 0$$
$$\leq P_{W_i} e_{\sigma_{v''}} \cdot P^{-1} P_{W_i} e_{\sigma_{v''}} - 2 P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_{v''}} + P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_v}$$

for all i, and the inequality is strict for at least one i. So, subtracting $P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_v}$ from both sides gives us that

$$P_{W_i} e_{\sigma_{v'}} \cdot P^{-1} P_{W_i} e_{\sigma_{v'}} - 2 P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_{v'}} \leq P_{W_i} e_{\sigma_{v''}} \cdot P^{-1} P_{W_i} e_{\sigma_{v''}} - 2 P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_{v''}}$$

for all i, and the inequality is still strict for at least one i.

So, given a representative vertex in each community, we can determine which of them a given vertex, v, is in the same community as without needing to know the value of $P_{W_i} e_{\sigma_v} \cdot P^{-1} P_{W_i} e_{\sigma_v}$ as follows.

The Vertex_classification_algorithm. The inputs are $(p, Q, G, v[], v', r, r', E, x, c)$, where $p \in (0,1)^k$ with $\sum p_i = 1$, Q is a $k \times k$ symmetric matrix with nonnegative coefficients, G is a graph, $v[]$ is a list of vertices, v' is a vertex, r, r' are positive integers, E is a subset of G's edges, x is a positive real number, and c is a real number between 0 and 1. It is assumed that approximations of $P_{W_i} e_{\sigma_{v[j]}} \cdot P^{-1} P_{W_i} e_{\sigma_{v[j]}}$ have already been computed for all $1 \leq i \leq \eta$ and $0 \leq j < k$

The algorithm is supposed to output σ such that v' is in the same community as $v[\sigma]$. It works as follows.

(1) Approximate $P_{W_i} e_{\sigma_{v'}} \cdot P^{-1} P_{W_i} e_{\sigma_{v[j]}}$ for every $1 \leq i \leq \eta$ and $0 \leq j < k$ based on the values of $N_{r+i,r'[E]}(v[j] \cdot v')$ for $1 \leq i \leq \eta$ and $0 \leq j < k$.

(2) If there exists a unique j such that for all $j' \neq j$ and all i, the approximation of $P_{W_i} e_{\sigma_{v[j]}} \cdot P^{-1} P_{W_i} e_{\sigma_{v[j]}} - 2 P_{W_i} e_{\sigma_{v'}} \cdot P^{-1} P_{W_i} e_{\sigma_{v[j]}}$ does not exceed the approximation of $P_{W_i} e_{\sigma_{v[j']}} \cdot P^{-1} P_{W_i} e_{\sigma_{v[j']}} - 2 P_{W_i} e_{\sigma_{v'}} \cdot P^{-1} P_{W_i} e_{\sigma_{v[j']}}$ by more than the acceptable error of the approximations, then conclude that v' is in the same community as $v[j]$.

(3) Otherwise, Fail.

This runs fairly quickly if r is large and r' is small because the algorithm only requires focusing on $N_{r'}(v')$ vertices. This leads to the following plan for partial recovery. First, randomly select a set of vertices that is large enough to contain at least one vertex from each community with high probability. Next, compare all of the selected vertices in an attempt to determine which of them are in the same communities. Then, pick one in each community. After that, use the algorithm above to attempt to determine which community each of the remaining vertices is in. As long as there actually was at least one vertex from each community in the initial set and none of the approximations were particularly bad, this should give a reasonably accurate classification.

The Unreliable_graph_classification_algorithm. The inputs are $(p, Q, G, c, m, \epsilon, x)$, where $p \in (0,1)^k$ with $\sum p_i = 1$, Q is a $k \times k$ symmetric matrix with nonnegative coefficients, G is a graph, c is a real number between 0 and 1, m is a positive integer, ϵ is a real number between 0 and 1, and x is a positive real number.

The algorithm outputs an alleged list of communities for G. It works as follows.

(1) Randomly assign each edge in G to E independently with probability c.

(2) Randomly select m vertices in G, $v[0], ..., v[m-1]$.

(3) Set $r = (1 - \frac{\epsilon}{3}) \log n / \log((1-c)\lambda_1) - \eta$ and $r' = \frac{2\epsilon}{3} \cdot \log n / \log((1-c)\lambda_1)$

(4) Compute $N_{r''[G \setminus E]}(v[i])$ for each $r'' < r + \eta$ and $0 \leq i < m$.

(5) Run Vertex_comparison_algorithm$(p, Q, G, v[i], v[j], r, r', E, x)$ for every i and j

(6) If these give results consistent with some community memberships which indicate that there is at least one vertex in each community in $v[]$, randomly select one alleged member of each community $v'[\sigma]$. Otherwise, fail.

(7) For every v'' in the graph, compute $N_{r''[G \setminus E]}(v'')$ for each $r'' < r'$. Then, run Vertex_classification_algorithm$(p, Q, G, v'[], v'', r, r', E, x)$ in order to get a hypothesized classification of v''.

(8) Return the resulting classification.

The risk that this randomly gives a bad classification due to a bad set of initial vertices can be mitigated as follows. First, repeat the previous classification procedure several times. Next, discard any classification that differs too badly from the majority. Assuming that the procedure gives a good classification more often than not, this should eliminate any really bad classification. Finally, average the remaining classifications together. This last procedure completes the `Sphere comparison` algorithm.

The Reliable_graph_classification_algorithm (i.e., `Sphere comparison`**).** The inputs are $(p, Q, G, c, m, \epsilon, x, T(n))$, where $p \in (0,1)^k$ with $\sum p_i = 1$, Q is a $k \times k$ symmetric matrix with nonnegative coefficients, G is a graph, c is a real number between 0 and 1, m is a positive integer, ϵ is a real number between 0 and 1, x is a positive real number, and T is a function from the positive integers to itself.

The algorithm outputs an alleged list of communities for G. It works as follows.

(1) Run Unreliable_graph_classification_algorithm$(p, Q, G, c, m, \epsilon, x)$ $T(n)$ times and record the resulting classifications.

(2) Discard any classification that has greater than

$$4ke^{-\frac{(1-c)x^2\lambda_\eta^2 \min p_i}{16\lambda_1 k(1+x)}}/(1-e^{-\frac{(1-c)x^2\lambda_\eta^2 \min p_i}{16\lambda_1 k(1+x)}\cdot((\frac{(1-c)\lambda_\eta^4}{4\lambda_1^3})-1)})$$

disagreement with more than half of the other classifications. In this step, define the disagreement between two classifications as the minimum disagreement over all bijections between their communities.

(3) Let $\{\sigma[i]\}$ be the remaining classifications. For each vertex $v \in G$, randomly select some i and assert that σ_v is the j that maximizes $|\{v' : \sigma[1]_{v'} = j\} \cap \{v' : \sigma[i]_{v'} = \sigma[i]_v\}|$. In other words, assume that $\sigma[i]$ classifies v correctly and then translate that to a community of $\sigma[1]$ by assuming the communities of $\sigma[i]$ correspond to the communities of $\sigma[1]$ that they have the greatest overlap with.

(4) Return the resulting combined classification.

If the conditions of theorem 2 are satisfied, then there exists x such that for all sufficiently small c,

$$\text{Reliable\_graph\_classification\_algorithm}(G, c, \ln(4k)/\min p_i, \epsilon, x, \ln n)$$

classifies at least

$$1 - \frac{4ke^{-\frac{C\rho}{16k}}}{1-e^{-\frac{C\rho}{16k}\left(\frac{(\lambda')^2}{4\lambda^2}\rho - 1\right)}} \tag{18}$$

of G's vertices correctly with probability $1 - o(1)$ and it runs in $O(n^{1+\epsilon})$ time.

B. Exact recovery and the `Degree-profiling` *algorithm*

With our previous result achieving almost exact recovery of the nodes, we are in a position to complete the exact recovery via a procedure that performs local improvements on the rough solution. While, the exact recovery requirement is rather strong, we show that it benefits from a phase transition, as opposed to almost exact recovery, which allows us to benchmark algorithms on a sharp limit (see Introduction).

Our analysis of exact recovery relies on the fact that the probability distribution of the numbers of neighbors a given vertex has in each community is essentially a multivariable Poisson distribution. We hence investigate an hypothesis problem (see Section 7.2 in [46]), where a node in the SBM graph with known clusters (up to $o(n)$ errors due to our previous results) is taken and re-classified based on its degree profile, i.e., on the number of neighbors it has in each community. This requires solving the following hypothesis testing problem, between k multivariate Poisson distributions.

The random variable H takes values in $[k]$ with $\mathbb{P}\{H = j\} = p_j$ (this is the a priori distribution of H). Under $H = j$, an observed random variable D is drawn from a multivariate Poisson distribution with mean $\lambda(j) \in \mathbb{R}_+^k$, i.e.,

$$\mathbb{P}\{D = d | H = j\} = \mathcal{P}_{\lambda(j)}(d), \quad d \in \mathbb{Z}_+^k, \tag{19}$$

where

$$\mathcal{P}_{\lambda(j)}(d) = \prod_{i \in [k]} \mathcal{P}_{\lambda_i(j)}(d_i), \tag{20}$$

and

$$\mathcal{P}_{\lambda_i(j)}(d_i) = \frac{\lambda_i(j)^{d_i}}{d_i!} e^{-\lambda_i(j)}. \tag{21}$$

In other words, D has independent Poisson entries with different means. We use the following notation to summarize the above setting:

$$D|H = j \ \sim \mathcal{P}(\lambda(j)), \quad j \in [k]. \tag{22}$$

Our goal is to infer the value of H by observing a realization of D. To minimize the error probability given a realization of D, we must pick the most likely hypothesis conditioned on this realization, i.e.,

$$\text{argmax}_{j\in[k]}\mathbb{P}\{D=d|H=j\}p_j, \tag{23}$$

which is the Maximum A Posteriori (MAP) decoding rule.[10] To resolve this maximization, we can proceed to a tournament of $k-1$ pairwise comparisons of the hypotheses. Each comparison allows us to eliminate one candidate for the maxima, i.e.,

$$\mathbb{P}\{D=d|H=i\}p_i > \mathbb{P}\{D=d|H=j\}p_j \quad \Rightarrow \quad H\neq j. \tag{24}$$

The error probability P_e of this decoding rule is then given by,

$$P_e = \sum_{i\in[k]} \mathbb{P}\{D\in \text{Bad}(i)|H=i\}p_i, \tag{25}$$

where $\text{Bad}(i)$ is the region in $\mathbb{Z}_+^k$ where i is not maximizing (23). Moreover, for any $i\in[k]$,

$$\mathbb{P}\{D\in \text{Bad}(i)|H=i\} \leq \sum_{j\neq i} \mathbb{P}\{D\in \text{Bad}_j(i)|H=i\} \tag{26}$$

where $\text{Bad}_j(i)$ is the region in $\mathbb{Z}_+^k$ where $\mathbb{P}\{D=x|H=i\}p_i \leq \mathbb{P}\{D=x|H=j\}p_j$. Note that with this upper-bound, we are counting the overlap regions where $\mathbb{P}\{D=x|H=i\}p_i \leq \mathbb{P}\{D=x|H=j\}p_j$ for different j's multiple times, but no more than $k-1$ times. Hence,

$$\sum_{j\neq i} \mathbb{P}\{D\in \text{Bad}_j(i)|H=i\} \leq (k-1)\mathbb{P}\{D\in \text{Bad}(i)|H=i\}. \tag{27}$$

Putting (25) and (26) together, we have

$$\begin{aligned} P_e &\leq \sum_{i\neq j} \mathbb{P}\{D\in \text{Bad}_j(i)|H=i\}p_i, &(28)\\ &= \sum_{i<j}\sum_{d\in\mathbb{Z}_+^k} \min(\mathbb{P}\{D=d|H=i\}p_i, \mathbb{P}\{D=d|H=j\}p_j) &(29)\end{aligned}$$

and from (27),

$$P_e \geq \frac{1}{k-1}\sum_{i<j}\sum_{d\in\mathbb{Z}_+^k} \min(\mathbb{P}\{D=d|H=i\}p_i, \mathbb{P}\{D=d|H=j\}p_j). \tag{30}$$

Therefore the error probability P_e can be controlled by estimating the terms $\sum_{d\in\mathbb{Z}_+^k}\min(\mathbb{P}\{D=d|H=i\}p_i, \mathbb{P}\{D=d|H=j\}p_j)$. In our case, recall that

$$\mathbb{P}\{D=d|H=i\} = \mathcal{P}_{\lambda(i)}(d), \tag{31}$$

which is a multivariate Poisson distribution. In particular, we are interested in the regime where k is constant and $\lambda(i) = \ln(n)c_i$, $c_i\in\mathbb{R}_+^k$, and n diverges. Due to (29), (30), we can then control the error probability by controlling $\sum_{x\in\mathbb{Z}_+^k}\min(\mathcal{P}_{\ln(n)c_i}(x)p_i, \mathcal{P}_{\ln(n}$ which we will want to be $o(1/n)$ to classify vertices in the SBM correctly with high probability based on their degree profiles (see next section). The following lemma provides the relevant estimates.

Theorem 3. *For any $c_1, c_2 \in (\mathbb{R}_+\setminus\{0\})^k$ with $c_1\neq c_2$ and $p_1, p_2\in\mathbb{R}_+\setminus\{0\}$,*

$$\sum_{x\in\mathbb{Z}_+^k}\min(\mathcal{P}_{\ln(n)c_1}(x)p_1, \mathcal{P}_{\ln(n)c_2}(x)p_2) = O\left(n^{-D_+(c_1,c_2)-\frac{\ln\ln(n)}{2\ln(n)}}\right), \tag{32}$$

$$\sum_{x\in\mathbb{Z}_+^k}\min(\mathcal{P}_{\ln(n)c_1}(x)p_1, \mathcal{P}_{\ln(n)c_2}(x)p_2) = \Omega\left(n^{-D_+(c_1,c_2)-\frac{k\ln\ln(n)}{2\ln(n)}}\right), \tag{33}$$

where $D_+(c_1,c_2)$ is the CH-divergence as defined previously.

[10]Ties can be broken arbitrarily.

In other words, the CH-divergence provides the error exponent for deciding among multivariate Poisson distributions. We did not find this result in the literature, but found a similar result obtained by Verdú [71], who shows that the Hellinger distance (the special case with $t = 1/2$ instead of the maximization over t) appears in the error exponent for testing Poisson point-processes, although [71] does not investigate the exact error exponent.

Using this result, the error probability of the optimal test is either $o(\frac{1}{n})$ or $\omega(\frac{1}{n})$ depending on $\min_{i<j} D_+(\theta_i, \theta_j)$. If the error probability is $\omega(\frac{1}{n})$ then any method of distinguishing between vertices in those two communities must fail with probability $\omega(\frac{1}{n})$, so any possible algorithm attempting to distinguish between them must misclassify at least one vertex with probability $1 - o(1)$. On the other hand, if the degree of overlap between all communities we are trying to distinguish between is $o(1/n)$ then with probability $1 - o(1)$ one could correctly classify any vertex in the graph if one knew what community each of its neighbors was in. There exists δ such that attempting to classify a vertex based on classifications of its neighbors that are wrong with probability x results in a probability of misclassifying the vertex that is only $n^{\delta x}$ times as high as it would be if they were all classified correctly. Based on this, the obvious approach to exact recovery would be to use a partial recovery algorithm to create a preliminary classification and then attempt to determine which family of communities each vertex is in based on its neighbors' alleged communities. However, the standard partial recovery algorithm has a constant error rate, so this proceedure's output would have an error rate n^c times as large as if each vertex were being classified based on its neighbors' true communities for some $c > 0$. If the degrees of overlap are only barely below $1/n$ then this would increase the error rate enough that this procedure would misclassify at least one vertex with high probability.

Instead, we go through three successively more accurate classifications as follows. Given a partial reconstruction of the communities with an error rate that is a sufficiently low constant, one can classify vertices based on their neighbors' alleged communities with an accuracy of $1 - O(n^{-c})$ for some constant $c > 0$. Then one can use this classification of a vertex's neighbors to determine which family of communities it is in with an accuracy of $1-o(\frac{1}{n} \cdot n^{\delta c' n^{-c}}) = 1-o(1/n)$. Therefore, the resulting classification is correct with probability $1 - o(1)$.

We formulate the algorithm in an adaptive way, where we first identify which communities can be exactly recovered with the notion of "finest partition," and then proceed to extract this partition. In other words, even in the case where not all communities can be exactly recovered, the algorithm may be able to fully extract a subset of the communities. Overall, the algorithm for exact recovery works as follows.

The `Degree-profiling` **algorithm.** The inputs are (G, γ), where G is a graph, and $\gamma \in [0, 1]$ (see Theorem 6 in [46] for how to set γ). The algorithm outputs an assignment of each vertex to one of the groups of communities $\{A_1, \ldots, A_t\}$, where $A_1, \ldots, A_t$ is the partition of $[k]$ in to the largest number of subsets such that $D_+((pQ)_i, (pQ)_j) \geq 1$ for all i, j in $[k]$ that are in different subsets (i.e., the "finest partition," see Firgure 1). It does the following:

(1) Define the graph g' on the vertex set $[n]$ by selecting each edge in g independently with probability γ, and define the graph g'' that contains the edges in g that are not in g'.

(2) Run `Sphere-comparison` on g' to obtain the preliminary classification $\sigma' \in [k]^n$ (see Section III-A.)

(3) Determine the edge density between each pair of alleged communities, and use this information and the alleged communities' sizes to attempt to identify the communities up to symmetry.

(4) For each node $v \in [n]$, determine in which community node v is most likely to belong to based on its degree profile in g'' computed from the preliminary classification σ' (see Section III-B), and call it σ''_v

(5) For each node $v \in [n]$, determine in which group $A_1, \ldots, A_t$ node v is most likely to belong to based on its degree profile in g'' computed from the preliminary classification σ''. See Section 7.2 in [46] for more details.

IV. Overlapping communities

We now define a model that accounts for overlapping communities, we refer to it as the overlapping stochastic block model (OSBM).

Definition 6. *Let $n, t \in \mathbb{Z}_+$, $f : \{0,1\}^t \times \{0,1\}^t \to [0,1]$ symmetric, and p a probability distribution on $\{0,1\}^t$. A random graph with distribution OSBM(n, p, f) is generated on the vertex set $[n]$ by drawing independently for each $v \in [n]$ the vector-labels (or user profiles) $X(v)$ under p, and by drawing independently for each $u, v \in [n]$, $u < v$, an edge between u and v with probability $f(X(u), X(v))$.*

Example 1. *One may consider $f(x, y) = \theta_g(x, y)$, where x_i encodes whether a node is in community i or not, and*

$$\theta_g(x, y) = g(\langle x, y \rangle), \tag{34}$$

where $\langle x, y \rangle = \sum_{i=1}^{t} x_i y_i$ counts the number of common communities between the labels x and y, and $g : \{0, 1, \ldots, t\} \to [0, 1]$ is a function that maps the overlap score into probabilities (g is typically increasing).

Example 2. *As a special case of the previous example, one may consider that a connection takes place between each pair of nodes as follows: each community (i.e., each component in the user profile) generates a connection independently with probability q_+ if the two nodes are in that community (i.e., if that component is 1 for both profiles), and multiple connections are equivalent to one connection. We also assume that any pair of nodes without a common community connects with probability q_-, so that*

$$g(s) = \begin{cases} 1-(1-q_+)^s, & \text{if } s \neq 0, \\ p_-, & \text{if } s = 0. \end{cases} \tag{35}$$

If we consider q_- and q_+ to be vanishing, like $O(\log(n)/n)$, we may consider the equivalent model where

$$g(s) = \begin{cases} sq_+, & \text{if } s \neq 0, \\ p_-, & \text{if } s = 0. \end{cases} \tag{36}$$

If $t = 1$, this model collapses to the usual symmetric stochastic block model with non-overlapping communities.

Note that in general we can represent the OSBM as a SBM with $k = 2^t$ communities, where each community represents a possible profile in $\{0,1\}^t$. For example, two overlapping communities can be modelled by assigning nodes with a single attribute $(1,0)$ and $(0,1)$ to each of the disjoint communities and nodes with both attributes $(1,1)$ to the overlap community, while nodes having none of the attributes, i.e., $(0,0)$, may be assigned to the null community.

Assume now that we identify community $i \in [k]$ with the profile corresponding to the binary expansion of $i-1$. The prior and connectivity matrix of the corresponding SBM are then given by

$$p_i = p(b(i)) \tag{37}$$

$$q_{i,j} = f(b(i), b(j)), \tag{38}$$

where $b(i)$ is the binary expansion of $i-1$, and

$$\text{OSBM}(n,p,f) \overset{(d)}{=} \text{SBM}(n,p,q). \tag{39}$$

We can then use the results of previous sections to obtain exact recovery in the OSBM.

Corollary 3. *Exact recovery is solvable for the OSBM if the conditions of Theorem 1 apply to the SBM(n,p,q) with p and q as defined in* (37)*,* (38)*.*

V. Further literature

The stochastic block model was first introduced in [20], and in [14], [13] as the planted bisection model. For the first three decades, a major portion of the literature has focused on exact recovery, in particular on the case with two symmetric communities. The table below summarizes a partial list of works for **exact recovery:**

Bui, Chaudhuri, Leighton, Sipser '84	min-cut method	$p = \Omega(1/n), q = o(n^{-1-4/((p+q)n)})$
Dyer, Frieze '89	min-cut via degrees	$p - q = \Omega(1)$
Boppana '87	spectral method	$(p-q)/\sqrt{p+q} = \Omega(\sqrt{\log(n)/n})$
Snijders, Nowicki '97	EM algorithm	$p - q = \Omega(1)$
Jerrum, Sorkin '98	Metropolis algorithm	$p - q = \Omega(n^{-1/6+\varepsilon})$
Condon, Karp '99	augmentation algorithm	$p - q = \Omega(n^{-1/2+\varepsilon})$
Carson, Impagliazzo '01	hill-climbing algorithm	$p - q = \Omega(n^{-1/2}\log^4(n))$
Mcsherry '01	spectral method	$(p-q)/\sqrt{p} \geq \Omega(\sqrt{\log(n)/n})$
Bickel, Chen '09	N-G modularity	$(p-q)/\sqrt{p+q} = \Omega(\log(n)/\sqrt{n})$
Rohe, Chatterjee, Yu '11	spectral method	$p - q = \Omega(1)$

These works display an impressive diversity of algorithms, but are mainly driven by the methodology and do not reveal the sharp behavioral transition that takes place in this model, as later shown in [39], [40] (see below). Before discussing these results, one should mention that various other works have considered recovery algorithms for multiple communities without identifying phase transitions. We refer to [52], [53] for a summary of these results. In particular, [53] has recently studied information-theoretic vs. computational tradeoffs in coarse regimes of the parameters for symmetric block models with a growing number of communities.

Phase transition phenomena for the SBM appeared first for weak recovery. In 2010, Coja-Oghalan [35] introduced the weak-recovery problem, and obtained bounds for the constant average degree regime using a spectral algorithm. Soon after,

[36] proposed a precise picture for weak-recovery using statistical physics arguments, with a sharp threshold conjectured at $(a-b)^2 = 2(a+b)$, when $a = pn$ and $b = pn$. This has opened the door to a new series of work on the SBM driven by phase transitions. The impossibility part of the conjecture was first proved in [54], using a reduction to broadcasting on trees [55], and the conjecture was fully established in 2014 with [37], [38].

Recently it was realized that exact recovery also admits a phase transition pheonemon. This was set in [39], and shortly after in [40], with the threshold located[11] at $\sqrt{a} - \sqrt{b} = \sqrt{2}$ when $a = pn/\ln(n)$ and $b = qn/\ln(n)$. Efficient algorithms were also obtained in these papers. Hence, weak and exact recovery are solved in the symmetric two-community SBM.

One should also mention a line of work on another community detection model called the Censored Block Model (CBM), studied in [56], [51]. This model and its variants were also studied in [57], [58], [59], [60], [61], [62]. A SDP relaxation as in [39] for the SBM was first proposed in [51] for the CBM, with a performance gap having roughly a factor 2. This gap was recently closed in [63]. SDP relaxations for block models were also studied in [53], [64], [65]. Note that SDP algorithms are polynomial time but far from quasi-linear time. For the CBM, recent works [44], [66] obtained tight bounds for weak recovery using spectral methods.

Two recent works [65], [44] have also obtained bounds for partial recovery in the SBM with multiple communities, for the case of symmetric blocks or with bounds on the connectivity probabilities in terms of symmetric blocks. The general SBM was treated in [67] for exact recovery with a spectral method, but the phase transition is not identified. No phase transitions for exact or weak recovery have yet been proved for the SBM with more than two communities.

VI. Open problems

Several extensions would be interesting for the SBM with specified parameters, such as considering parameters that vary with n, in particular for the number of communities, or communities of sub-linear sizes. Part of the results obtained in this paper should extend without much difficulty to some of these cases. It would also be interesting to investigate how the complexity of algorithms scales with the number of communities.[12] It would also be important to obtain results and algorithms that do not rely on the knowledge of the model parameters. Here also, some of the techniques in this paper may extend. For partial recovery, it would interesting to obtain tight upper-bounds on the accuracy of the reconstruction in the general SBM, in particular for the regime of large constant degrees, to check if the bound obtained in this paper is tight. For the symmetric case, the information-theoretic and computational thresholds for weak-recovery remain open for more than 2 communities.

Finally, there are many interesting other models to investigate, such as the Censored Block Model [56], [51], [61], [57], [58], [59], [60], [62], the Labelled Block Model [68], [69] and many more. It would be natural to expect that for these models as well, an information-measure à la CH-divergence obtained in this paper determines the recovery threshold.

Acknowledgements

We would like to thank Bell Labs for supporting part of this research, as well as Sergio Verdú and Imre Csiszár for useful discussions on f-divergences.

References

[1] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826, 2002.

[2] M. E. J. Newman, D. J. Watts, and S. H. Strogatz. Random graph models of social networks. *Proc. Natl. Acad. Sci. USA*, 99:2566–2572.

[3] G. Linden, B. Smith, and J. York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, January 2003.

[4] J. Xu, R. Wu, K. Zhu, B. Hajek, R. Srikant, and L. Ying. Jointly clustering rows and columns of binary matrices: Algorithms and trade-offs. *SIGMETRICS Perform. Eval. Rev.*, 42(1):29–41, June 2014.

[5] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22:888–905, 1997.

[6] M. Sonka, V. Hlavac, and R. Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007.

[11] [40] allows for a slightly more general model where a and b are $\Theta(1)$ and gives the behaviour at the threshold. Note that at the threshold, one has to distinguish the case of $b = 0$ and $b > 0$ (assuming $a > b$), since for $b = 0$ the clusters are not connected whp and it is not possible to recover the clusters with a vanishing error probability.

[12] In [53] this question is studied for coarser regimes of the parameters.

[7] J. Chen and B. Yuan. Detecting functional modules in the yeast proteinprotein interaction network. *Bioinformatics*, 22(18):2283–2290, 2006.

[8] E.M. Marcotte, M. Pellegrini, H.-L. Ng, D.W. Rice, T.O. Yeates, and D. Eisenberg. Detecting protein function and protein-protein interactions from genome sequences. *Science*, 285(5428):751–753, 1999.

[9] J. K. Pritchard, M. Stephens, and P. Donnelly. Inference of Population Structure Using Multilocus Genotype Data. *Genetics*, 155(2):945–959, June 2000.

[10] D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: a survey. *Knowledge and Data Engineering, IEEE Transactions on*, 16(11):1370–1386, Nov 2004.

[11] T. Sorlie, C.M. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, Mi.B. Eisen, M. van de Rijn, S.S. Jeffrey, T. Thorsen, H. Quist, J.C. Matese, P.O. Brown, D. Botstein, P.E. Lonning, and A. Borresen-Dale. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. 98(19):10869–10874, 2001.

[12] M. Newman. *Networks: an introduction*. Oxford University Press, Oxford, 2010.

[13] M.E. Dyer and A.M. Frieze. The solution of some random NP-hard problems in polynomial expected time. *Journal of Algorithms*, 10(4):451 – 489, 1989.

[14] T.N. Bui, S. Chaudhuri, F.T. Leighton, and M. Sipser. Graph bisection algorithms with good average case behavior. *Combinatorica*, 7(2):171–191, 1987.

[15] A. Condon and R. M. Karp. Algorithms for graph partitioning on the planted partition model. *Lecture Notes in Computer Science*, 1671:221–232, 1999.

[16] B. Bollobás and A. D. Scott. Max cut for random graphs with a planted partition. *Comb. Probab. Comput.*, 13(4-5):451–474, July 2004.

[17] C. Rudin, D. Dunson, H. Ji R. Irizarry, E. Laber, J. Leek, T. McCormick, S. Rose, C. Schafer, M. van der Laan, L. Wasserman, and L. Xue. Discovery with data: Leveraging statistics with computer science to transform science and society. *American Statistical Association. Report*, July 2014.

[18] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, july, october 1948.

[19] D. Achlioptas, A. Naor, and Y. Peres. Rigorous Location of Phase Transitions in Hard Optimization Problems. *Nature*, 435:759–764, 2005.

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

[21] H. C. White, S. A. Boorman, and R. L. Breiger. Social structure from multiple networks. *American Journal of Sociology*, 81:730–780, 1976.

[22] S. E. Fienberg, M. M. Meyer, and S. S. Wasserman. Statistical analysis of multiple sociometric relations. *Journal of The American Statistical Association*, pages 51–67, 1985.

[23] Y. J. Wang and G. Y. Wong. Stochastic blockmodels for directed graphs. *Journal of the American Statistical Association*, pages 8–19, 1987.

[24] P. J. Bickel and A. Chen. A nonparametric view of network models and newmangirvan and other modularities. *Proceedings of the National Academy of Sciences*, 2009.

[25] B. Karrer and M. E. J. Newman. Stochastic blockmodels and community structure in networks. *Phys. Rev. E*, 83:016107, Jan 2011.

[26] R.B. Boppana. Eigenvalues and graph bisection: An average-case analysis. *In 28th Annual Symposium on Foundations of Computer Science*, pages 280–285, 1987.

[27] Mark Jerrum and Gregory B. Sorkin. The metropolis algorithm for graph bisection. *Discrete Applied Mathematics*, 82(13):155 – 175, 1998.

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

[29] F. McSherry. Spectral partitioning of random graphs. In *Foundations of Computer Science, 2001. Proceedings. 42nd IEEE Symposium on*, pages 529–537, 2001.

[30] T. A. B. Snijders and K. Nowicki. Estimation and Prediction for Stochastic Blockmodels for Graphs with Latent Block Structure. *Journal of Classification*, 14(1):75–100, January 1997.

[31] K. Rohe, S. Chatterjee, and B. Yu. Spectral clustering and the high-dimensional stochastic blockmodel. *The Annals of Statistics*, 39(4):1878–1915, 08 2011.

[32] D. S. Choi, P. J. Wolfe, and E. M. Airoldi. Stochastic blockmodels with a growing number of classes. *Biometrika*, pages 1–12, 2012.

[33] E. M. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. *J. Mach. Learn. Res.*, 9:1981–2014, June 2008.

[34] P. K. Gopalan and D. M. Blei. Efficient discovery of overlapping communities in massive networks. *Proceedings of the National Academy of Sciences*, 2013.

[35] A. Coja-oghlan. Graph partitioning via adaptive spectral techniques. *Comb. Probab. Comput.*, 19(2):227–284, March 2010.

[36] A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová. Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications. *Phys. Rev. E*, 84:066106, December 2011.

[37] L. Massoulié. Community detection thresholds and the weak Ramanujan property. In *STOC 2014: 46th Annual Symposium on the Theory of Computing*, pages 1–10, New York, United States, June 2014.

[38] E. Mossel, J. Neeman, and A. Sly. A proof of the block model threshold conjecture. *Available online at arXiv:1311.4115 [math.PR]*, January 2014.

[39] E. Abbe, A. S. Bandeira, and G. Hall. Exact recovery in the stochastic block model. *To appear in IEEE Transactions on Information Theory. Available at ArXiv:1405.3267*, May 2014.

[40] E. Mossel, J. Neeman, and A. Sly. Consistency thresholds for binary symmetric block models. *Arxiv:arXiv:1407.1591. To appear in STOC15.*, July 2014.

[41] E. Mossel, J. Neeman, and A. Sly. Belief propagation, robust reconstruction, and optimal recovery of block models. *Arxiv:arXiv:1309.1380*, 2013.

[42] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney. Statistical properties of community structure in large social and information networks. In *Proceedings of the 17th international conference on World Wide Web*, WWW '08, pages 695–704, New York, NY, USA, 2008. ACM.

[43] S. H. Strogatz. Exploring complex networks. *Nature*, 410(6825):268–276, March 2001.

[44] P. Chin, A. Rao, and V. Vu. Stochastic block model and community detection in the sparse graphs: A spectral algorithm with optimal rate of recovery. *arXiv:1501.05021*, January 2015.

[45] S. Sachdeva G. Schoenebeck S. Arora, R. Ge. Finding Overlapping Communities in Social Networks: Toward a Rigorous Approach. *arXiv:1112.1831*, December 2011.

[46] E. Abbe and C. Sandon. Community detection in general stochastic block models: fundamental limits and efficient recovery algorithms. *arXiv:1503.00609*, March 2015.

[47] I. Csiszár. Eine informationstheoretische ungleichung und ihre anwendung auf den beweis der ergodizitat von markoffschen ketten. *Magyar. Tud. Akad. Mat. Kutató Int. Közl*, 8:85–108, 1963.

[48] T. Morimoto. Markov processes and the h-theorem. *Journal of the Physical Society of Japan*, 18(3):328–331, 1963.

[49] S. M. Ali and S. D. Silvey. A General Class of Coefficients of Divergence of One Distribution from Another. *Journal of the Royal Statistical Society. Series B (Methodological)*, 28(1):131–142, 1966.

[50] N. Alon and N. Kahale. A spectral technique for coloring random 3-colorable graphs. In *SIAM Journal on Computing*, pages 346–355, 1994.

[51] E. Abbe, A.S. Bandeira, A. Bracher, and A. Singer. Decoding binary node labels from censored edge measurements: Phase transition and efficient recovery. *Network Science and Engineering, IEEE Transactions on*, 1(1):10–22, Jan 2014.

[52] Y. Chen, S. Sanghavi, and H. Xu. Clustering Sparse Graphs. *arXiv:1210.3335*, 2012.

[53] J. Xu Y. Chen. Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. *arXiv:1402.1267*, February 2014.

[54] E. Mossel, J. Neeman, and A. Sly. Stochastic Block Models and Reconstruction. 2012. arXiv:1202.1499 [math.PR].

[55] W. Evans, C. Kenyon, Y. Peres, and L. J. Schulman. Broadcasting on trees and the Ising model. *Ann. Appl. Probab.*, 10:410–433, 2000.

[56] E. Abbe and A. Montanari. Conditional random fields, planted constraint satisfaction and entropy concentration. In *Proc. of RANDOM*, pages 332–346, Berkeley, August 2013.

[57] E. Abbe and A. Montanari. Conditional random fields, planted constraint satisfaction and entropy concentration. *Theory of Computing*, 11, 2015.

[58] Q.-X. Huang and L. Guibas. Consistent shape maps via semidefinite programming. *Computer Graphics Forum*, 32(5):177–186, 2013.

[59] Y. Chen, Q.-X. Huang, and L. Guibas. Near-optimal joint object matching via convex relaxation. *Available Online: arXiv:1402.1473 [cs.LG]*, 2014.

[60] Y. Chen and A. J. Goldsmith. Information recovery from pairwise measurements. *In Proc. ISIT, Honolulu.*, 2014.

[61] E. Abbe, A.S. Bandeira, A. Bracher, and A. Singer. Linear inverse problems on Erdös-Rényi graphs: Information-theoretic limits and efficient recovery. In *Information Theory (ISIT), 2014 IEEE International Symposium on*, pages 1251–1255, June 2014.

[62] A. Globerson, T. Roughgarden, D. Sontag, and C. Yildirim. Tight error bounds for structured prediction. *CoRR*, abs/1409.5834, 2014.

[63] J. Xu B. Hajek, Y. Wu. Achieving exact cluster recovery threshold via semidefinite programming. *arXiv:1412.6156*, November 2014.

[64] A. Amini and E. Levina. On semidefinite relaxations for the block model. *arXiv:1406.5647*, June 2014.

[65] O. Guédon and R. Vershynin. Community detection in sparse networks via Grothendieck's inequality. *ArXiv:1411.4686*, November 2014.

[66] A. Saade, F. Krzakala, M. Lelarge, and L. Zdeborová. Spectral detection in the censored block model. *arXiv:1502.00163*, January 2015.

[67] V. Vu. A simple svd algorithm for finding hidden partitions. *Available online at arXiv:1404.3918*, April 2014.

[68] S. Heimlicher, M. Lelarge, and L. Massoulié. Community detection in the labelled stochastic block model. *arXiv:1209.2910*, 2012.

[69] J. Xu, M. Lelarge, and L. Massoulie. Edge label inference in generalized stochastic block models: from spectral theory to impossibility results. *Proceedings of COLT 2014*, 2014.

[70] M. Raginsky and I. Sason. Concentration of measure inequalities in information theory, communications, and coding. *Foundations and Trends in Communications and Information Theory*, 10(1-2):1–246, 2013.

[71] S. Verdú. Asymptotic error probability of binary hypothesis testing for poisson point-process observations (corresp.). *Information Theory, IEEE Transactions on*, 32(1):113–115, Jan 1986.