Linear inverse problems on Erdős-Rényi graphs: Information-theoretic limits and efficient recovery

Emmanuel Abbe, Afonso S. Bandeira, Annina Bracher, Amit Singer
Princeton University

***Abstract*—This paper considers the linear inverse problem $Y = AX \oplus Z$, where A is the incidence matrix of an Erdős-Rényi graph, Z is an i.i.d. noise vector, and X is the vector of unknown variables, assumed to be Boolean. This model is motivated by coding, synchronization, and community detection problems. Without noise, exact recovery is possible if and only the graph is connected, with a sharp threshold at the edge probability $\log(n)/n$. The goal of this paper is to determine how the edge probability p needs to scale in order to cope with the noise. Defining the rate parameter $r = \log(n)/np$, it is shown that for an error probability of ε close to half, exact recovery is possible if and only if r is below $D(1/2||\varepsilon)$. In other words, $D(1/2||\varepsilon)$ provides the information theoretic threshold for exact recovery at low-snr. In addition, an efficient recovery algorithm based on semidefinite programming is proposed and shown to succeed up to half the threshold.**

I. Introduction

A large variety of problems in information theory, machine learning, and image processing are concerned with inverse problems on graphs, i.e., problems where a graphical structure governs the dependencies between the variables that are observed and the variables that are unknown. In simple cases, the dependency model is captured by an undirected graph with the unknown variables attached at the vertices and the observed variables attached at the edges. Let $G = (V, E)$ be a graph with vertex set V and edge set E, and let x^V be the vertex- and y^E the edge-variables. In many cases of interest (detailed below), the probabilistic model for the edge-variables *conditionally* on the vertex-variables has a simple structure: it factorizes as

$$P(y^E|x^V) = \prod_{e\in E} Q(y_e|x[e]), \tag{1}$$

where y_e denotes the variable attached to edge e, $x[e]$ denotes the two vertex-variables incident to edge e, and Q is a local probability kernel. In this paper, we consider Boolean edge- and vertex-variables, and we assume that the kernel Q is symmetric and depends only on the XOR of the vertex-variables.[1] The edge-variables can then be viewed as a random vector Y^E that satisfies

$$Y^E = B_G x^V \oplus Z^E, \tag{2}$$

where B_G is the incidence matrix of the graph, i.e., the $|E| \times |V|$ matrix with $(B_G)_{e,v} = 1$ if and only if vertex v belongs to edge e, and Z is a random vector of dimension $|E|$ representing the noise.

In the above setting, the forward problem of recovering the most likely edge-variables given the vertex-variables is trivial and amounts to maximizing Q for each edge. The inverse problem, however, is more challenging: the most likely vertex-variables (say with a uniform prior) given the edge-variables cannot be found by local maximization. This type of problem arises in various contexts:

Coding: For a (symmetric) kernel Q, equation (2) corresponds to the output of a simple code, namely a 2-right-degree LDGM code, over a binary (symmetric) channel. While this is not a particularly interesting code by itself (e.g., at any fixed rate, it has a constant fraction of isolated vertices), it is a relevant primitive for the construction of other codes such as LT or raptor codes [18], [19].

Constraint satisfaction problems: The model (1) is a particular case of the graphical channel studied in [2] in the context of hypergraphs. This class of models allows in particular to recover instances of planted constraint satisfaction problems (CSPs) by choosing uniform kernels Q, where the vertex-variables represent the planted assignment and the edge-variables represent the clauses. In the case of a simple graph and not a hypergraph, this provides a model for planted formulae like 2-SAT or 2-XORSAT (model (2)).

Synchronization: Equation (2) results also from the synchronization problem studied in [20], [6], [24], [3], [7], if the dimension is one, i.e., each vertex-variable is the 1-bit quantization of the reflection of a signal. The goal in synchronization over $O(d)$, the group of orthogonal matrices of size $d \times d$, is to recover exactly the original values of the node-variables $\{x_j\}_{j\in[n]}$, assumed to take values in $O(d)$, given the relative measurements $\{J_{ij}x_i^{-1}x_j\}_{i,j\in[n]}$, where J_{ij} is randomly drawn in $O(d)$ if the vertices i and j are adjacent and all-zero else.[2] When $d = 1$, we have $O(1) = \{-1, +1\}$ and the synchronization problem is equivalent to (2).

Community detection: The model in (1) can also be used as a probabilistic network model. The basic Erdős-Rényi model [11], is typically not a good model for networks: all vertices have the same expected degree and no cluster structure appears. One possibility to obtain cluster structures is to attach latent variables to the vertices and assume an edge distribution that depends on these variables.

[1] Symmetry means that $Q(y|x_1, x_2) = P(y|x_1 \oplus x_2)$ for some P that satisfies $P(1|1) = P(0|0)$.

[2] If J_{ij} is the $d \times d$ identity matrix, then the measurement is noise-free.

There are various models with latent variables, such as the exchangeable, inhomogeneous or stochastic block models [12], [25], [13], [10], [15], [9]. The model in (1) can be used for this purpose. The vertex-variables represent the community assignment, the edge-variables the connectivity, and the graph G the information is available. In the noiseless additive case and for every edge $e = (i, j)$, the edge-variable $y_e = x_i \oplus x_j$ encodes whether the vertices i and j are in the same community or not. With noise on top, vertices in the same/different communities are also allowed disconnect/connect, which is a more realistic model.

In this paper we are interested in particular in a "filtered" block model, as introduced in [2] in a different context. In such a model, a base-graph $G = (V, E(G))$ and a binary community assignment $X \in \{0, 1\}^V$ are used to generate a random graph on the vertex set V with ternary edge labels $E_{ij} \in \{*, 0, 1\}$ drawn independently with the following probability distribution:

$$\mathbb{P}\{E_{ij} = *|E(G)_{ij} = 0\} = 1 \tag{3a}$$

$$\mathbb{P}\{E_{ij} = 1|X_i = X_j, E(G)_{ij} = 1\} = q_1, \tag{3b}$$

$$\mathbb{P}\{E_{ij} = 1|X_i \neq X_j, E(G)_{ij} = 1\} = q_2. \tag{3c}$$

Put differently, (3) is a graph model where information is only available on the base-graph G, the $*$-variable encodes the absence of information, and when there is information available, then two vertices are connected with probability q_1 if they are in the same community and with probability q_2 if they are in different communities. When $G = K_n$ is the complete graph and X is uniformly distributed, this is the standard stochastic block model with two communities [8], [17]. In the case of (2), the linear structure implies that $q_1 = 1 - q_2 = \varepsilon$.

While all the above mentioned problems are concerned with related inverse problems on graphs, there are various refinements that can be considered for the recovery. This paper focuses on *exact recovery*, which requires that all vertex-variables be recovered simultaneously with high probability as the number of vertices diverges. The probability measure may depend on the graph ensemble or simply on the kernel Q if the graph is deterministic. Note, however, that exact recovery of all variables in the model (2) is not quite possible: the vertex-variables x^V and $1^V \oplus x^V$ produce the same output Y^E. Exact recovery is meant "up to a global flipping of the variables". For *partial recovery,* only a constant fraction of the vertex-variables are to be recovered correctly with high probability as the number of vertices diverges. Put differently, the true assignment need only be positively correlated with the reconstruction. The recovery requirements vary with the applications, e.g., exact recovery is typically required in coding theory to ensure reliable communication. In community detection, partial recovery is the best one can hope for sparse networks, while exact recovery can be considered for slightly denser graphs with logarithmic degree [16].

This paper focuses on exact recovery for the linear model (2) with Boolean variables, and on Erdős-Rényi models for the base-graph G. For this setup, we identify the information-theoretic phase transition for exact recovery in terms of the edge density of the graph and the noise level, and devise an efficient algorithm based on semidefinite programming (SDP), which approaches the threshold up to a factor of 2. This SDP based method was first proposed in [20], and it shares many aspects with the SDP methods in several other problems [21], [14].

II. Model and results

In this paper, we focus on the linear Boolean model

$$Y^E = B_G x^V \oplus Z^E, \tag{4}$$

where the vector components are in $\{0, 1\}$ and the addition is modulo 2. We require exact recovery for x^V arbitrary and consider for the underlying graph $G = (V, E)$ the Erdős-Rényi model $\mathrm{ER}(n, p)$, where $V = [n]$ and the edges are drawn i.i.d. with probability p. We assume that the noise vector Z^E has i.i.d. components, equal to 1 with probability ε. We assume[3] w.l.o.g. that $\varepsilon \in [0, 1/2]$, where 0 means no noise (and exact recovery amounts to having a connected graph) and $1/2$ means maximal noise (and exact recovery is impossible no matter how connected the graph is). Note that the inverse problem is much easier if the noise model causes erasures with probability ε instead of errors. Exact recovery is then possible if and only if the graph is still connected after the erasure of the edges. Since there is a sharp threshold for connectedness at $p = \frac{\log(n)}{n}$, this happens a.a.s. if $p = \frac{(1+\delta)\log(n)}{n(1-\varepsilon)}$ for some $\delta > 0$. Hence $1-\varepsilon$ is a sharp threshold in $\log(n)/np$ for the exact recovery problem with erasures and base-graph $\mathrm{ER}(n, p)$.

The goal of this paper is to find the replacement to the erasure threshold $1-\varepsilon$ for the setting where the noise causes errors. Similarly to channel coding where the Shannon capacity of the BSC(ε) differs from the BEC(ε) capacity, we obtain for the considered inverse problem the expression $D(1/2||\varepsilon)$ for the low-snr regime (ε close to 1/2).

More precisely, this paper establishes an information theoretic (IT) necessary condition that holds for every graph (Theorem IV.1), and an IT (Theorem IV.2) sufficient condition for Erdős-Rényi graphs. Moreover, we also give a recovery guarantee that holds for an efficient algorithm based on SDP (Theorem V.3).

In particular, we show that, for $\varepsilon = 1/2 - \Delta_n$, where $\Delta_n = o(1)$ and $\Delta_n = \Omega(n^{-\tau})$ for every $\tau > 0$[4], it is optimal to draw the graph from the Erdős-Rényi model: The bounds for the necessary condition for a general graph and the IT sufficient condition for the Erdős-Rényi graph match, and they match the SDP bound up to a factor $1/2$.

If the noise parameter ε is bounded away from both zero and $1/2$, then all conditions imply $m/n = \Theta(\log(n))$, where

[3]The noise model is assumed to be known, hence the regime $\varepsilon \in [1/2, 1]$ can be handled by adding an all-one vector to Y^E.

[4]The assumption $\Delta_n = o(1)$ is standard for the synchronization problem in dimension $d = 1$.

m is the expected number of edges: $m = \binom{n}{2}p$. The factors by which the bounds differ decrease with an increasing noise parameter ε. Since in the noise-free case exact recovery is possible if and only if the graph is connected, which is true for trees with $m = n-1$ and for Erdős-Rényi graphs with $m = n\log(n)/2$, the factors between the necessary condition and the sufficient conditions approaches infinity when ε decreases to zero (since $D(1/2||\varepsilon)$ diverges).

III. Directions and open problems

There are various extensions to consider for the above models, including the generalization to q-ary instead of binary variables and the extension to problems with hyper-edges instead of edges as in [2]. Non-binary variables would be particularly interesting for the synchronization problem in higher dimension, $d \geq 2$, where the orientations are quantized to a higher order. There are several extensions that are interesting for the community detection applications. First, it would be important to investigate non-symmetric noise models, i.e., noise models that are non-additive. Then, it would be interesting to study partial (as opposed to exact) recovery for sparse graphs with constant degrees, and to incorporate constraints on the size of the communities. Finally, one could consider other ensembles for the base-graph. In the full version of this paper, which will appear in [1], the spectral gap of the base-graph is used to analyze the threshold behaviour for any deterministic graph.

IV. Information Theoretic Bounds

This section presents necessary and sufficient conditions for exact recovery of the vertex-variables x^V from the edge-variables Y^E. We require that ML-decoding recover the vertex-variables x^V up to an unavoidable additive offset $\phi \in \{0^V, 1^V\}$ with some probability that converges to 1 as the number of vertices approaches infinity.

Notation: All logarithms have base e, and we denote by $D(1/2\,||\varepsilon\,) = 1/2\log(1/(2\varepsilon)) + 1/2\log(1/(2(1-\varepsilon)))$ the Kullback-Leibler divergence between $1/2$ and ε and by $h_b(q) = q\log(1/q) + (1-q)\log(1/(1-q))$ the entropy (in nats) of a binary random variable that assumes the value 1 with probability $q \in [0,1]$.

A. A Necessary Condition for Successful Recovery

For each graph $G = (V, E)$ (drawn from the Erdős-Rényi model or not), the following result holds:

Theorem IV.1. *Necessary conditions for exact recovery are: Let $0 < \tau < 2/3$. If $m/n \leq n^\tau$, then*

$$\frac{m}{n} \geq \frac{1-3\tau/2}{2D(1/2\,||\varepsilon\,)}\log(n) + o\left(\frac{\log n}{D(1/2\,||\varepsilon\,)}\right). \tag{5}$$

If $\varepsilon \to 1/2$, this implies for $(1-2\varepsilon) \geq n^{-\tau/2}$

$$\frac{m}{n} \geq \frac{1-3\tau/2}{(1-2\varepsilon)^2}\log(n) + o\left(\frac{\log n}{(1-2\varepsilon)^2}\right). \tag{6}$$

Proof: Fix a vertex v_j, and let $\mathcal{E}_j$ denote the event that the variables attached to at least half of the edges that are incident to Vertex v_j are noisy. As we argue next, if event $\mathcal{E}_j$ occurs, then ML-decoding recovers vertex-variables other than x^V or $x^V \oplus 1^V$ with probability at least $1/2$. Indeed, if ML-decoding correctly recovers the vertex-variables that are attached to the vertices adjacent to v_j up to a global additive offset $\phi \in \{0,1\}$, then—by assumption that event $\mathcal{E}_j$ occurs—the probability that ML-decoding recovers x_j with offset $\phi\oplus 1$ is at least $1/2$. In particular, this implies that ML-decoding can only be successful if the event $\bigcap_{v_j \in V} \mathcal{E}_j^c$ occurs. Let $\mathcal{Q}$ be a subset of $[n]$ such that no two vertices with indices in $\mathcal{Q}$ are adjacent. Since the noise Z^E is drawn IID, the events $\{\mathcal{E}_j\}_{j\in\mathcal{Q}}$ are independent and the probability of the event $\bigcap_{j\in\mathcal{Q}} \mathcal{E}_j^c$ is easily computable. Moreover, the event $\bigcap_{j\in[n]} \mathcal{E}_j^c$ can only occur if $\bigcap_{j\in\mathcal{Q}} \mathcal{E}_j^c$ occurs. A necessary condition for exact recovery thus is that the probability of the event $\bigcap_{j\in\mathcal{Q}} \mathcal{E}_j^c$ converges to one as the number of vertices increases. In the following, we prove the claim by identifying a set $\mathcal{Q}$ and upper-bounding the probability of the event $\bigcap_{j\in\mathcal{Q}} \mathcal{E}_j^c$.

Let $\deg(v_j)$ be the degree of Vertex v_j, and assume w.l.g. $\deg(v_1) \leq \deg(v_2) \leq \ldots \leq \deg(v_n)$. For every $0 < \delta \leq 1$

$$2m \geq \sum_{j=\lceil \delta n\rceil}^{n} \deg(v_j) \geq \lceil (1-\delta)\,n\rceil \deg\big(v_{\lceil \delta n\rceil}\big)\,. \tag{7}$$

For $j \leq \lceil \delta n\rceil$, we therefore find

$$\deg(v_j) \leq \deg\big(v_{\lceil \delta n\rceil}\big) \leq \frac{2m}{\lceil (1-\delta)\,n\rceil} \leq \frac{2m}{(1-\delta)\,n}. \tag{8}$$

This implies that for every set $\mathcal{L} \subseteq \{1,\ldots,\lceil \delta n\rceil\}$, the vertices $\{v_j \colon j \in \mathcal{L}\}$ are disconnected from at least

$$\lceil \delta n\rceil - |\mathcal{L}|\left(1 + \frac{2m}{(1-\delta)\,n}\right) \tag{9}$$

vertices in the set $\{v_j \colon j \leq \lceil \delta n\rceil\}$. In particular, there is a set $\mathcal{Q} \subseteq \big[\lceil \delta n\rceil\big]$ such that no two distinct vertices with indices in the set $\mathcal{Q}$ are adjacent and such that

$$|\mathcal{Q}| \geq \frac{\lceil \delta n\rceil}{1 + \frac{2m}{(1-\delta)n}} \geq \frac{\delta\,(1-\delta)\,n^2}{2m + (1-\delta)\,n}. \tag{10}$$

If $j \leq \lceil \delta n\rceil$, then one can show that

$$\begin{aligned}\Pr[\mathcal{E}_j] &= \sum_{k=\left\lceil \frac{\deg(v_j)}{2}\right\rceil}^{\deg(v_j)} \binom{\deg(v_j)}{k} \varepsilon^k\,(1-\varepsilon)^{\deg(v_j)-k} \\ &\geq e^{-\frac{1}{2}\log\left(\left(\frac{1-\varepsilon}{\varepsilon}\right)\frac{2m}{(1-\delta)n}\right) - \log(2) - \frac{2mD(1/2||\varepsilon\,)}{(1-\delta)n}}.\end{aligned} \tag{11}$$

Since the events $\big\{\mathcal{E}_j^c \colon j \in \mathcal{Q}\big\}$ are jointly independent,

$$\begin{aligned}\Pr\left[\bigcap_{j\in\mathcal{Q}} \mathcal{E}_j^c\right] &= \prod_{j\in\mathcal{Q}} (1 - \Pr[\mathcal{E}_j]) \\ &\overset{a)}{\leq} e^{-\sum_{j\in\mathcal{Q}} e^{-\frac{1}{2}\log\left(\left(\frac{1-\varepsilon}{\varepsilon}\right)\frac{2m}{(1-\delta)n}\right) - \log(2) - \frac{2mD(1/2||\varepsilon\,)}{(1-\delta)n}}} \\ &\overset{b)}{\leq} e^{-e^{\log\left(\frac{\delta(1-\delta)n^2}{4m+2(1-\delta)n}\sqrt{\frac{(1-\delta)n}{2m}}\sqrt{\frac{\varepsilon}{1-\varepsilon}}\right) - \frac{2mD(1/2||\varepsilon\,)}{(1-\delta)n}}},\end{aligned} \tag{12}$$

where a) holds since $1-x \leq e^{-x}$ for $x \geq 0$ and because of (11), and b) is due to (10). To conclude, take $\delta = \log(n)^{-1}$, and observe that for $m/n \leq n^{\tau}$, where $\tau < 2/3$, the RHS of (12) is bounded away from 1 unless (5) holds. ∎

It is interesting to compare Theorem IV.1 to the necessary condition $\frac{m}{n} \geq \frac{1}{1-h_b(\varepsilon)/\log 2}$, previously shown in [20, Section 5]. If $\varepsilon \in (0, 1/2)$ does not depend on n, then this condition only implies $m/n = \Omega(1)$ and is thus weaker than $m/n = \Omega(\log n)$, which follows from Theorem IV.1. If $\varepsilon = 1/2 - \Delta_n$ and $\Delta_n = o(1)$, then $h_b(\varepsilon) = 1 - 2\Delta_n^2/\log 2 + o\big(\Delta_n^2\big)$, and we can write the condition in [20] as $\Delta_n = \Omega(\sqrt{n/m})$. If there is a $\tau < 2/3$ such that $\Delta_n \geq n^{-\tau/2}$, then Theorem IV.1 is tighter: it implies $\Delta_n = \Omega(\sqrt{\log(n)\, n/m})$. However, if there is no such τ, then Theorem IV.1 cannot be applied.

B. A Sufficient Condition for Successful Recovery

We next present a sufficient condition for exact recovery. For a random base-graph $G = (V, E)$ from the Erdős-Rényi model, we require the vertex-variables x^V to be recoverable from the edge-variables Y^E except with some probability that vanishes as the number of vertices increases.

Theorem IV.2. *If the base-graph is from the Erdős-Rényi model $ER(n, p)$ with $p > 2\log(n)/n$, then the condition*

$$\frac{m}{n} \geq \frac{1}{2\left(1 - \sqrt{\frac{\log(n)}{m/n}}\right) D(1/2\,||\varepsilon\,)} \log(n) + o\left(\frac{\log(n)}{D(1/2\,||\varepsilon\,)}\right) \tag{13}$$

is sufficient for exact recovery. If $\epsilon \to 1/2$ the condition is

$$\frac{m}{n} \geq \frac{1}{(1-2\epsilon)^2} \log(n) + o\left(\frac{\log(n)}{(1-2\varepsilon)^2}\right). \tag{14}$$

Proof: One can prove the theorem using that ML-decoding succeeds if every tuple $\tilde{x}^V \notin \{x^V, x^V \oplus 1^V\}$ satisfies $d_H\big(Y^E, B_G \tilde{x}^V\big) > d_H\big(Y^E, B_G x^V\big)$. ∎

V. Computationally efficient recovery - the SDP

In this section we analyze a tractable method to recover x^V from the noisy measurements Y^E based in semidefinite programming. Ideally, one would like to find the maximum likelihood estimator $x^* = \operatorname{argmin}_{x_i \in \{0,1\}} \sum_{(i,j)\in E} 1_{\{x_i \neq y_{(i,j)} \oplus x_j\}}$. By defining the $\{\pm 1\}$-valued variables $g_i = (-1)^{x_i}$ and the coefficients $\rho_{ij} = (-1)^{y_{(i,j)}}$, the ML problem is reformulated as

$$\min_{g_i \in \{\pm 1\}} \sum_{(i,j)\in E} |g_i - \rho_{ij} g_j|^2. \tag{15}$$

This problem is known to be NP-hard in general (in fact, it is easy to see that it can encode `Max-Cut`). In what follows, we will describe and analyze a tractable algorithm, which was first proposed in [20] to approximate the solution of (15). We will state conditions under which the algorithm is able to recover the vertex-variables x^V. The idea is to consider a natural semidefinite relaxation. Other properties of this SDP have been studied in [4], [5].

Notation: Recall that G is the underlying graph on n nodes, and let H be the subgraph representing the incorrect edges (corresponding to $Z_{(i,j)} = 1$). Let A_G, A_H, D_G, D_H, L_G, and L_H be, respectively, the adjencacy, degree, and Laplacian matrices of the graphs G and H.

As in [20], we assume w.l.o.g. that $x^V \equiv 0$ so that $g \equiv 1$. Then, $A_G - 2A_H$ is the matrix whose (i,j)-th entry is equal to ρ_{ij} if $(i,j) \in E$ and 0 otherwise. Problem (15) has the same solutions as $\max_{g_i \in \{\pm 1\}} \operatorname{Tr}\left[(A_G - 2A_H) g g^T\right]$, which in turn is equivalent to

$$\begin{aligned} &\max\ \operatorname{Tr}\left[(A_G - 2A_H) X\right] \\ &\text{s.t. } X \in \mathbb{R}^{n\times n},\ X_{ii} = 1\ \forall i,\ X \succeq 0,\ \operatorname{Rank}(X) = 1. \end{aligned} \tag{16}$$

(Given the optimal rank 1 solution X of (16), $g_i = (-1)^{x_i}$ is the only non-trivial eigenvector of X.) As the rank constraint is non-convex, we consider the following convex relaxation

$$\max \operatorname{Tr}\left[(A_G - 2A_H) X\right] \quad \text{s.t. } X_{ii} = 1,\ X \succeq 0. \tag{17}$$

Note that (17) is an SDP and can be solved, up to arbitrary precision, in polynomial time [23]. Note that a solution of (17) need not be rank 1 and thus need not be a solution of (16). However, we will show that under certain conditions (17) recovers the same optimal solution as (16). In this case, $g_i = (-1)^{x_i}$ is the only non-trivial eigenvector of X and x^V can be recovered via the tractable problem (17).

Our objective is to understand when $X = gg^T = 11^T$ is the unique optimal solution to (17). The dual of the SDP is

$$\min \operatorname{Tr}(Z) \quad \text{s.t. } Z \text{ diagonal},\ Z - (A_G - 2A_H) \succeq 0. \tag{18}$$

Duality guarantees that the objective value of (17) cannot exceed that of (18). Thus, if there exists Z, feasible solution of (18), such that $\operatorname{Tr}(Z) = \operatorname{Tr}\left[(A_G - 2A_H)\, 11^T\right]$, then $X = 11^T$ is an optimal solution of (17). Moreover, Z and 11^T have to satisfy complementary slackness: $\operatorname{Tr}(11^T(Z - (A_G - 2A_H))) = 0$. Given these constraints, a natural candidate is $Z = D_G - 2D_H$. Indeed, it is easy to see that $\operatorname{Tr}(D_G - 2D_H) = \operatorname{Tr}\left[(A_G - 2A_H)\, 11^T\right]$. Hence, if

$$L_G - 2L_H = D_G - 2D_H - (A_G - 2A_H) \succeq 0, \tag{19}$$

i.e., the dual variable satisfies the PSD constraint), then 11^T must be an optimal solution of (17). Additionally, if $L_G - 2L_H$ is not only PSD but also its second smallest eigenvalue is non-zero, then it is not difficult to show that 11^T is the unique optimal solution. We have thus shown:

Lemma V.1. *If*

$$L_G - 2L_H \succeq 0 \text{ and } \lambda_2(L_G - 2L_H) > 0, \tag{20}$$

then 11^T is the unique solution to (17).

Remark V.2. *A similar exact recovery sufficient condition was independently obtained by Huang and Guibas [14] in the context of consistent shape map estimation (see Theorem 5.1. in [14]). Their analysis goes on to show, essentially, that as long as the probability of a wrong edge is strictly smaller than $\frac{1}{2}$, the probability of exact recovery converges to 1 as*

the size of the graph is arbitrarily large. On the other hand, we are able to show near tight rates at which this phase transition happens. For a given ϵ that is arbitrarily close to $\frac{1}{2}$ we give an essentially-tight bound on the size of the graph and edge density needed for exact recovery (Theorem V.3).

A. Erdős-Rényi Model

We now assume that the underlying graph is drawn from the Erdős-Rényi model $\mathrm{ER}(n,p)$ and use condition (20) to give guarantees for exact recovery.

For each pair of vertices $i<j$, let Λ_{ij} be the matrix that is 1 in the entries (i,i) and (j,j), -1 in the entries (i,j) and (j,i), and 0 elsewhere. Observe that $\Lambda_{ij}\succeq 0$, and $L_G=\sum_{i<j:(i,j)\in E}\Lambda_e$. Let α_{ij} be the random variable that takes the value 0 if Edge (i,j) is not in G, the value 1 if it is in G but not in H, and the value -1 if it is in H. Hence α_{ij} are iid and take the values $0,1,-1$ with probability, respectively, $1-p$, $p\,(1-\varepsilon)$, and $p\varepsilon$.

In the new notation, $L_G-2L_H=\sum_{i<j}\alpha_{ij}\Lambda_{ij}$. We define the centered random variables $A_{ij}=(p(1-2\varepsilon)-\alpha_{ij})\,\Lambda_{ij}$. For $A=\sum_{i<j}A_{ij}$, we can write $L_G-2L_H=p(1-2\varepsilon)(nI-\mathbf{1}\mathbf{1}^T)-A$. Since Λ_{ij} always contains the vector $\mathbf{1}$ in the null-space, (20) is equivalent to $\lambda_{\max}(A)<p(1-2\varepsilon)n$.

We next argue for which values p, ε, and n there is some $\delta>0$ such that $\mathrm{Prob}\left[\lambda_{\max}(A)\geq p(1-2\varepsilon)n\right]\leq n^{-\delta}$. To this end, we use the Matrix Bernstein inequality (Theorem 1.4 in [22]), which implies $\mathrm{Prob}\left[\lambda_{\max}(A)\geq t\right]\leq n\exp\left(-\frac{t^2/2}{\sigma^2+Rt/3}\right)$, where $\sigma^2=\left\|\sum_{i<j}\mathbb{E}A_{ij}^2\right\|$, with $\|\cdot\|$ denoting the spectral norm, and $R\geq\lambda_{\max}\left(A_k\right)$. Note that $\sigma^2=2np\left[1-p(1-2\varepsilon)^2\right]$ and $\lambda_{\max}(A_{ij})\leq 2p(1-2\varepsilon)+2$. Setting $t=p(1-2\epsilon)n$ gives us the following Theorem.

Theorem V.3. *Let m be the expected number of edges. If,*

$$\frac{m}{n}\geq(1+\delta)\left(\frac{2}{(1-2\varepsilon)^2}+\frac{2}{3(1-2\varepsilon)}\right)\log n, \qquad (21)$$

then the SDP achieves exact recovery with probability at least $1-n^{-\delta}$. When $\epsilon\to\frac{1}{2}$, condition (21) is equivalent to

$$\frac{m}{n}\geq 2\frac{(1+\delta)}{(1-2\varepsilon)^2}\log n+o\left(\frac{1}{(1-2\varepsilon)^2}\right)\log n. \qquad (22)$$

Remark V.4. *A simpler method to recover x^V would be to fix a vertex-variable, say x_i^V, to take the value 1. Then consider, for every vertex $j\neq i$, all length 2 paths of from Vertex i to Vertex j and use a voting scheme to determine x_j^V. The recovery guarantees we obtained for this method exhibit the weaker scaling $(1-2\epsilon)^4 m/n\geq\Omega\left(\log n/n\right)$.*

ACKNOWLEDGEMENTS

We thank Andrea Montanari for suggesting us the algorithm described in Remark V.4 and Joel Tropp for insightful discussions regarding [22].

REFERENCES

[1] E. Abbe, A. S. Bandeira, A. Bracher, and A. Singer, *Decoding graph labels from censored measurements: Phase transitions and efficient recovery*, In preparation.

[2] E. Abbe and A. Montanari, *Conditional random fields, planted constraint satisfaction and entropy concentration*, Proc. of RANDOM (Berkeley), 2013, pp. 332–346.

[3] B. Alexeev, A. S. Bandeira, M. Fickus, and D. G. Mixon, *Phase retrieval with polarization*, SIAM J. on Imaging Sci. **7** (2013), no. 1, 35–66.

[4] N. Alon and A. Naor, *Approximating the cut-norm via Grothendieck's inequality*, Proc. of the 36 th ACM STOC, ACM Press, 2004, pp. 72–80.

[5] A. S. Bandeira, C. Kennedy, and A. Singer, *Approximating the little grothendieck problem over the orthogonal and unitary groups*, Available online at arXiv:1308.5207 [cs.DS] (2013).

[6] A. S. Bandeira, A. Singer, and D. A. Spielman, *A Cheeger inequality for the graph connection Laplacian*, SIAM J. Matrix Anal. Appl. **34** (2013), no. 4, 1611–1630.

[7] N. Boumal, A. Singer, P.-A. Absil, and V. D. Blondel, *Cramér-rao bounds for synchronization of rotations*, Information and Inference: A Journal of the IMA.

[8] 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 (2011).

[9] P. Doreian, V. Batagelj, and A. Ferligoj, *Generalized Blockmodeling (Structural Analysis in the Social Sciences)*, Cambridge University Press, November 2004.

[10] M.E. Dyer and A.M. Frieze, *The solution of some random np-hard problems in polynomial expected time*, Journal of Algorithms **10** (1989), no. 4, 451 – 489.

[11] P. Erdös and A. Rényi, *On random graphs, I*, Publicationes Mathematicae (Debrecen) **6** (1959), 290–297.

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

[13] P. W. Holland, K. Laskey, and S. Leinhardt, *Stochastic blockmodels: First steps*, Social Networks **5** (1983), no. 2, 109–137.

[14] Q.-X. Huang and L. Guibas, *Consistent shape maps via semidefinite programming*, Computer Graphics Forum **32** (2013), no. 5, 177–186.

[15] B. Karrer and M. E. J. Newman, *Stochastic blockmodels and community structure in networks*, Phys. Rev. E **83** (2011), 016107.

[16] F. McSherry, *Spectral partitioning of random graphs*, FOCS '01: Proceedings of the 42nd IEEE symposium on Foundations of Computer Science, 2001, p. 529.

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

[18] K. Raj Kumar, P. Pakzad, A.H. Salavati, and A. Shokrollahi, *Phase transitions for mutual information*, Turbo Codes and Iterative Information Processing (ISTC), 2010 6th International Symposium on, 2010, pp. 137–141.

[19] A. Shokrollahi, *Lt-codes and phase transitions for mutual information*, Information Theoretic Security (Serge Fehr, ed.), Lecture Notes in Computer Science, vol. 6673, Springer Berlin Heidelberg, 2011, pp. 94–99.

[20] A. Singer, *Angular synchronization by eigenvectors and semidefinite programming*, Appl. Comput. Harmon. Anal. **30** (2011), no. 1, 20 – 36.

[21] A. M.-C. So, *Probabilistic analysis of the semidefinite relaxation detector in digital communications.*, SODA (Moses Charikar, ed.), SIAM, 2010, pp. 698–711.

[22] J. A. Tropp, *User-friendly tail bounds for sums of random matrices*, Found. Comput. Math. **12** (2012), no. 4, 389–434.

[23] L. Vanderberghe and S. Boyd, *Semidefinite programming*, SIAM Review **38** (1996), 49–95.

[24] L. Wang and A. Singer, *Exact and stable recovery of rotations for robust synchronization*, Information and Inference: A Journal of the IMA **2** (2013), no. 2, 145–193.

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