Improved Graph Clustering

Yudong Chen, Sujay Sanghavi, *Member, IEEE*, and Huan Xu

***Abstract*— Graph clustering involves the task of dividing nodes into clusters, so that the edge density is higher within clusters as opposed to across clusters. A natural, classic, and popular statistical setting for evaluating solutions to this problem is the stochastic block model, also referred to as the planted partition model. In this paper, we present a new algorithm—a convexified version of maximum likelihood—for graph clustering. We show that, in the classic stochastic block model setting, it outperforms existing methods by polynomial factors when the cluster size is allowed to have general scalings. In fact, it is within logarithmic factors of known lower bounds for spectral methods, and there is evidence suggesting that no polynomial time algorithm would do significantly better. We then show that this guarantee carries over to a more general extension of the stochastic block model. Our method can handle the settings of semirandom graphs, heterogeneous degree distributions, unequal cluster sizes, unaffiliated nodes, partially observed graphs, planted clique/coloring, and so on. In particular, our results provide the best exact recovery guarantees to date for the planted partition, planted k-disjoint-cliques and planted noisy coloring models with general cluster sizes; in other settings, we match the best existing results up to logarithmic factors.**

***Index Terms*— Graph clustering, maximum likehood estimator, convex optimization, stochastic block model.**

I. Introduction

THIS paper proposes a new algorithm for the following task: given an undirected unweighted graph, assign the nodes into disjoint clusters so that the density of edges within clusters is higher than the edge density across clusters. Clustering arises in applications such as a community detection, user profiling, link prediction, collaborative filtering etc. In these applications, one is often given as input a set of similarity relationships (either "1" or "0") and the goal is to identify groups of similar objects. For example, given the friendship relations on Facebook, one would like to detect tightly connected communities, which is useful for subsequent tasks like customized recommendation and advertisement.

Graphs in modern applications have several characteristics that complicate graph clustering:

- **Small Density Gap:** the edge density across clusters is only a small additive or multiplicative factor different from within clusters;
- **Sparsity:** the graph is overall very sparse even within clusters;
- **High Dimensionality:** the number of clusters may grow unbounded as a function of the number of nodes n, which means the sizes of the clusters can be vanishingly small compared to n;
- **Unaffiliated Nodes:** there may exist a large number of nodes that do not belong to any clusters and are loosely connected to the rest of the graph;
- **Heterogeneity:** the cluster sizes, node degrees and edge densities may be non-uniform across the graph; edge connections may not be well-modeled by a probabilistic distribution, and there may exist hierarchical cluster structures.

Various large modern datasets and graphs have such characteristics [1], [2]; examples include the web graph and social graphs of various social networks etc. As has been well-recognized, these characteristics make clustering more difficult. When the in-cluster and across-cluster edge densities are close or there are many unstructured unaffiliated nodes, the clustering structure is less significant and thus harder to detect. Sparsity further reduces the amount of information and makes the problem noisier. In the high dimensional regime, there are many small clusters, which are easy to lose in the noise. Heterogeneous and non-random structures in the graphs foil many algorithms that otherwise perform well; for example, conventional spectral clustering methods are often known to be not robust to heterogeneity in the graphs [3], [4]. Finally, the existence of hierarchical structures and unaffiliated nodes renders many existing algorithms and theoretical results inapplicable, as they fix the number of clusters a priori and force each node to be assigned to a cluster. It is desirable to design an algorithm that can handle all these issues in a principled manner.

A. Our Contributions

Our algorithmic contribution is a new method for unweighted graph clustering. It is motivated by the maximum-likelihood estimator for the classical Stochastic Block-model [5] (also known as the Planted Partition Model [6])

Manuscript received September 20, 2013; revised March 11, 2014; accepted July 14, 2014. Date of publication August 7, 2014; date of current version September 11, 2014. Y. Chen was supported in part by the National Science Foundation (NSF) under Grant EECS-1056028 and in part by the Defense Threat Reduction Agency (DTRA) under Grant HDTRA 1-08-0029. S. Sanghavi was supported in part by the DTRA Young Investigator Award and in part by the NSF under Grant 1302435, Grant 1017525, and Grant 0954059. H. Xu was supported by the Ministry of Education of Singapore through Academic Research Fund Tier 2 under Grant R-265-000-443-112. This paper was presented at the 2012 NIPS conference.

Y. Chen is with the Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Berkeley, CA 94720 USA (e-mail: yudong.chen@eecs.berkeley.edu).

S. Sanghavi is with the Department of Electrical and Computer Engineering, University of Texas at Austin, Austin, TX 78712 USA (e-mail: sanghavi@mail.utexas.edu).

H. Xu is with the Department of Mechanical Engineering, National University of Singapore, Singapore 117575 (e-mail: mpexuh@nus.edu.sg).

Communicated by N. Cesa-Bianchi, Associate Editor for Pattern Recognition, Statistical Learning and Inference.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIT.2014.2346205

0018-9448 © 2014 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

for random clustered graphs. In particular, we show that this maximum-likelihood estimator can be written as a linear objective over combinatorial constraints; our algorithm is a convex relaxation of these constraints, yielding a convex program overall. While this is the motivation, it performs well—both in theory and empirically—in settings that are not just the standard stochastic blockmodel.

Our main analytical result in this paper is theoretical guarantees on our algorithm's performance; we study it in a *semi-random generalized stochastic blockmodel*. This model generalizes not only the standard stochastic blockmodel and planted partition model, but many other classical planted models including planted k-disjoint-cliques [7], [8], planted dense subgraph [9], planted coloring [4], [10] and their semi-random variants [11]–[13]. Our main result gives the conditions (as a function of the in/cross-cluster edge densities p and q, the density gap $|p-q|$, the minimum cluster size K and the total number of nodes n) under which our algorithm is guaranteed to recover the ground-truth clustering. When $p > q$, the key condition reads

$$p - q = \Omega\left(\frac{\sqrt{p(1-q)n}}{K}\right); \tag{1}$$

here all the parameters are allowed to scale with n. Note that the condition does not depend explicitly on the number of unaffiliated nodes or the number of clusters. An analogous result holds for $p < q$.

While the planted and stochastic block models have a rich literature, this single result shows that the performance of our algorithm matches all existing methods (up to at most logarithmic factors) in exact recovery; moreover, in the cases of the standard planted partition/k-disjoint-cliques/noisy-coloring models with general scaling of p, q and K, we achieve order-wise improvement over existing methods, in the sense that our algorithm succeeds for a much larger range of the parameters. In fact, there is evidence indicating that we are close to the boundary at which any polynomial-time algorithm can be expected to work. The proof for our main theorem is relatively simple, relying only on standard concentration results. Our simulation study supports our theoretic finding, that the proposed method is effective in clustering noisy graphs and outperforms existing methods.

The rest of the paper is organized as follows: Section I-B provides an overview of related work; Section II presents our algorithm; Section III describes the Semi-Random Generalized Stochastic Blockmodel, which is a generalization of the standard stochastic blockmodel, one that allows the modeling of the issues mentioned above; Section IV presents the main results—a performance analysis of our algorithm for the semi-random generalized stochastic blockmodel, and provides a detailed comparison to the existing literature and a discussion of the implications for different special cases; Section V provides simulation results; the proofs of our theoretic results are given in Sections VI to IX; the paper concludes with a discussion in Section X.

B. Related Work

The general field of clustering, or even graph clustering, is too vast for a detailed survey here; we focus on the most related threads, and therein too primarily on work which provides analytical guarantees on the resulting algorithms.

1) Stochastic Block Models: Also called "planted models" [5], [6], these are arguably the most natural random clustered graph models. In the simplest or standard setting, n nodes are partitioned into disjoint subsets of equal size K (called the true clusters), and then edges are generated independently and at random, with the probability p of an edge between two nodes in the same cluster higher than the probability q for two nodes in different clusters. The task is to recover the true clusters given the graph. The parameters p, q, K and n typically govern whether an algorithm succeeds in recovery or not.

There is now a long line of analytical work on stochastic block models; we focus on methods that allow for *exact recovery* (i.e., every node is correctly classified), and summarize the conditions required by known methods in Table I. As can be seen, we improve over existing methods by polynomial factors for general values of K—in particular, when the cluster size satisfies $K = n^{1-\alpha}$ for any constant $\alpha > 0$ (which means the number of clusters is growing at the rate $n/K = n^{\alpha}$.)[1] In addition, as opposed to several of these methods, our method can handle unaffiliated nodes, heterogeneity, hierarchy in clustering etc, and apply to other models including planted clique and planted noisy coloring.

We would like to mention two recent results that appeared after the conference version [23] of this paper. The work in [24] shows that a computationally efficient tensor decomposition approach succeeds for the standard stochastic blockmodel when $p - q = \Omega\left(\sqrt{pn}\ \text{polylog}\ n/K\right)$; our guarantee (1) is better by a factor of $\Theta(\text{polylog}\ n/\sqrt{1-q})$. Moreover, for the standard planted clique model ($p = 1, q = 1/2$), we only require the clique size to be $K = \Omega(\sqrt{n})$, better than their requirement $K = \tilde{\Omega}(n^{2/3})$. Another subsequent work [25] considers the setting with heterogeneous cluster sizes and no unaffiliated nodes, and shows that our algorithm can be combined with an iterative reduction procedure to sequentially recover clusters smaller than is allowed in this paper.

A complimentary line of work has investigated *lower bounds* for the stochastic blockmodel; i.e., for what values/scalings of p, q and K it is *not* possible (either for any algorithm, or for any polynomial-time algorithm) to recover the underlying true clusters [3], [26], [27]. We discuss and compare with these two lines of work in more details in the main results section.

2) Convex Methods for Matrix Decomposition: Our method is related to recent literature on the recovery of low-rank matrices using convex optimization, and in particular the recovery of such matrices from "sparse" perturbations (i.e., where a fraction of the elements of the low-rank matrix are possibly arbitrarily modified, while others are untouched). Sparse and low-rank matrix decomposition using convex optimization was initiated by [28] and [29]; follow-up works [30], [31] have the current state-of-the-art guarantees on this problem, and [32] applies it directly to graph clustering.

[1]Our comparison focuses on polynomial factors and the setting with general values of K. We note that in the special case of $K = \Theta(n)$, some existing results (see [19]) are better then ours by logarithmic factors.

TABLE I
COMPARISON WITH LITERATURE FOR THE STANDARD STOCHASTIC BLOCKMODEL

Paper	Cluster size K	Density gap $p-q$	Sparsity p
Boppana (1987) [14]	$n/2$	$\tilde{\Omega}\left(\sqrt{\frac{p}{n}}\right)$	$\tilde{\Omega}\left(\frac{1}{n}\right)$
Jerrum et al. (1998) [15]	$n/2$	$\tilde{\Omega}\left(\frac{1}{n^{1/6-\epsilon}}\right)$	$\tilde{\Omega}\left(n^{1/6-\epsilon}\right)$
Condon et al. (2001) [5]	$\Theta(n)$	$\tilde{\Omega}\left(\frac{1}{n^{1/2-\epsilon}}\right)$	$\tilde{\Omega}\left(n^{1/2-\epsilon}\right)$
Carson et al. (2001) [16]	$n/2$	$\tilde{\Omega}\left(\sqrt{\frac{p}{n}}\right)$	$\tilde{\Omega}\left(\frac{1}{n}\right)$
Feige et al. (2001) [12]	$n/2$	$\tilde{\Omega}\left(\sqrt{\frac{p}{n}}\right)$	$\tilde{\Omega}\left(\frac{1}{n}\right)$
McSherry (2001) [9]	$\Omega\left(n^{2/3}\right)$	$\tilde{\Omega}\left(\sqrt{\frac{pn^2}{K^3}}\right)$	$\tilde{\Omega}\left(\frac{n^2}{K^3}\right)$
Bollobas (2004) [11]	$\Theta(n)$	$\tilde{\Omega}\left(\sqrt{\frac{q}{n}} \vee \frac{1}{n}\right)$	$\tilde{\Omega}\left(\frac{1}{n}\right)$
Giesen et al. (2005) [17]	$\Omega\left(\sqrt{n}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$
Shamir (2007) [18]	$\Omega\left(\sqrt{n\log n}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$
Coja-Oghlan (2010) [19]	$\Omega(n^{4/5})$	$\tilde{\Omega}\left(\sqrt{\frac{pn^4}{K^5}}\right)$	$\tilde{\Omega}\left(\frac{n^4}{K^5}\right)$
Rohe et al. (2011) [20]	$\Omega\left((n\log n)^{2/3}\right)$	$\tilde{\Omega}\left(\frac{n^{1/2}}{K^{3/4}}\right)$	$\tilde{\Omega}\left(\frac{1}{\sqrt{\log n}}\right)$
Oymak et al. (2011) [21]	$\Omega\left(\sqrt{n}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$
Chaudhuri et al. (2012) [3]	$\Omega\left(\sqrt{n\log n}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$
Ames (2012) [22]	$\Omega\left(\sqrt{n}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{n}}{K}\right)$
Our result	$\Omega\left(\sqrt{n}\right)$	$\tilde{\Omega}\left(\frac{\sqrt{pn}}{K}\right)$	$\tilde{\Omega}\left(\frac{n}{K^2}\right)$

The method in this paper is Maximum Likelihood, but it can also be viewed as a weighted version of sparse and low-rank matrix decomposition, with *different elements of the sparse part penalized differently, based on the given input graph.* There is currently little work or analysis on weighted matrix decomposition; in that sense, while our weights have a natural motivation in our setting, our recovery results are likely to have broader implications, for example robust versions of PCA when not all errors are created equal but have a corresponding prior.

II. ALGORITHM

We now describe our algorithm. As mentioned, it is a convex relaxation of the Maximum Likelihood (ML) estimator as applied to the standard stochastic blockmodel. So, in what follows, we first develop notation and the exact ML estimator, and then its relaxation.

ML for the Standard Stochastic Blockmodel: Recall that in the standard stochastic blockmodel nodes are divided into disjoint clusters, and edges in the graph are chosen independently; the probability of an edge between a pair of nodes in the same cluster is p, and for a pair of nodes in different clusters it is q. Given the graph, the task is to find the underlying clusters that generated it. To write down the ML estimator for this, let us represent any candidate clustering by a corresponding *cluster matrix* $Y \in \mathbb{R}^{n\times n}$ where $y_{ij} = 1$ if and only if nodes i and j are assigned to the same cluster,[2] and $y_{ij} = 0$ otherwise; in particular, $y_{ii} = 1$ for any node i that belongs to a cluster. Any Y thus needs to have a block-diagonal structure, with each block being all 1's.

A vanilla ML estimator then involves optimizing a likelihood subject to the combinatorial constraint that the search space is the cluster matrices. Let $A \in \mathbb{R}^{n\times n}$ be the observed adjacency matrix of the graph (we assume $a_{ii} = 1$ for all i); then, the log likelihood function of A given Y is

$$\log \mathbb{P}(A|Y) = \log\left(\prod_{(i,j):y_{ij}=1} p^{a_{ij}}(1-p)^{1-a_{ij}} \prod_{(i,j):y_{ij}=0} q^{a_{ij}}(1-q)^{1-a_{ij}}\right).$$

We notice that this can be written, via a re-arrangement of terms, as

$$\log \mathbb{P}(A|Y) = \log\left(\frac{p}{q}\right)\sum_{a_{ij}=1} y_{ij} - \log\left(\frac{1-q}{1-p}\right)\sum_{a_{ij}=0} y_{ij} + C, \quad (2)$$

where C collects the terms that are independent of Y. The ML estimator would be maximizing the above expression subject to Y being a cluster matrix. While the objective is a linear function of Y, this optimization problem is combinatorial due to the requirement that Y be a cluster matrix (i.e., block-diagonal with each block being all-ones), and is intractable in general.

Our Algorithm: We obtain a convex and tractable algorithm by replacing the constraint "Y is a cluster matrix" with (i) the constraints $0 \le y_{ij} \le 1$ for all pairs (i, j), and (ii) a nuclear norm[3] regularizer $\|Y\|_*$ in the objective. The latter encourages Y to be *low-rank*, and is based on the well-established insight that a cluster matrix has low rank—in particular, its rank equals the number of clusters. (We discuss other related relaxations later in this section.)

Also notice that the likelihood expression (2) is linear in Y and only the *ratio* of the two coefficients $\log(p/q)$ and $\log((1-q)/(1-p))$ is important. We therefore introduce a parameter t which allows us to choose any ratio. This has

[2]Throughout this paper, for any matrix M, m_{ij} denotes its (i, j)-th entry.

[3]The nuclear norm of a matrix is the sum of its singular values.

Algorithm 1 Convex Clustering

Input: $A \in \mathbb{R}^{n\times n}$, $t \in (0,1)$
Solve program (3)–(4) with weights (5). Let $\widehat{Y}$ be an optimal solution.
if $\widehat{Y}$ is a cluster matrix **then**
  Output cluster memberships obtained from $\widehat{Y}$.
else
  Output "Failure".
end if

the advantage that instead of knowing both p and q, we only need to choose one number t (which should be between p and q; we remark on how to choose t later). This leads to the following convex formulation:

$$\max_{Y\in\mathbb{R}^{n\times n}} \quad c_{\mathcal{A}} \sum_{a_{ij}=1} y_{ij} \; - c_{\mathcal{A}^c} \sum_{a_{ij}=0} y_{ij} \; - 48\sqrt{n}\,\|Y\|_* \tag{3}$$

$$\text{s.t.} \;\; 0 \le y_{ij} \le 1, \quad \forall i, j. \tag{4}$$

where the weights $c_{\mathcal{A}}$ and $c_{\mathcal{A}^c}$ are given by

$$c_{\mathcal{A}} = \sqrt{\frac{1-t}{t}} \quad \text{and} \quad c_{\mathcal{A}^c} = \sqrt{\frac{t}{1-t}}. \tag{5}$$

Here the factor $48\sqrt{n}$ balances the contributions of the nuclear norm and the likelihood; the specific values of this factor as well as of $c_{\mathcal{A}}$ and $c_{\mathcal{A}^c}$ are derived from our analysis (cf. Section VII-E). The optimization problem (3)–(4) is convex and can be cast as a Semidefinite Program (SDP) [28], [33]. More importantly, it can be solved using efficient first-order methods for large graphs (see Section V-A).

Our algorithm is given as Algorithm 1. Depending on the given A and the choice of t, the optimal solution $\widehat{Y}$ may or may not be a cluster matrix. Checking if a given $\widehat{Y}$ is a cluster matrix can be done easily, e.g., via an SVD, which will also reveal the cluster memberships if it is a cluster matrix. If it is not, any one of several rounding/aggregation ideas (e.g., the one in [34]) can be used empirically; we do not delve into this approach in this paper, and simply output failure. In Section IV we provide sufficient conditions under which $\widehat{Y}$ is guaranteed to be a cluster matrix, with *no* rounding required.

A. Remarks About the Algorithm

Note that while we derive our algorithm from the standard stochastic blockmodel, our analytical results hold in a much more general setting. In practice, one could execute the algorithm (with appropriate choice of t, and hence $c_{\mathcal{A}}$ and $c_{\mathcal{A}^c}$) on any given graph.

1) Tighter Relaxations: The formulation (3)–(4) is not the only way to relax the non-convex ML estimator. Instead of the nuclear norm regularizer, a hard constraint $\|Y\|_* \le n$ may be used. One may further replace this constraint with the positive semidefinite constraint $Y \succeq 0$ and the linear constraints $y_{ii} = 1$, both satisfied by any cluster matrix.[4] It is not hard to check that these modifications lead to convex relaxations with smaller feasible sets, so any performance guarantee for our formulation (3)–(4) also applies to these alternative formulations. We choose to focus on our original formulation based on the following theoretical and practical considerations: a) Its performance guarantees apply to the other tighter relaxations as well. b) We do not obtain order-wise better theoretical guarantees with these alternative formulations. The work [34] considers these tighter relaxations but does not obtain better exact recovery guarantees than ours. In fact, as we argue in the next section, our guarantees are likely to be order-wise optimal and thus any alternative convex formulations are unlikely to provide significant improvements in a scaling sense. c) Our simpler formulation facilitates efficient solution for large graphs via first-order methods; we describe one such method in Section V-A.

2) Choice of t: Our algorithm requires an extraneous input t. For the standard planted k-disjoint-cliques problem [7], [9] (with k disjoint cliques planted in a random graph $G_{n,1/2}$), one can use $t = 3/4$ (see Section IV-C.2). For the standard stochastic blockmodel (with nodes partitioned into equal-size clusters and edge probabilities being uniformly p and q inside and across clusters), the value of t can be determined from the data (see Section IV-D). In these cases, our algorithm has no tuning parameters whatsoever and does not require knowledge of the number or sizes of the clusters. For the general setting, t should be chosen to lie between p and q, which now represent the lower/upper bounds for the in/cross-cluster edge densities. As such, t can be interpreted as the *resolution* of the clustering algorithm. To see this, suppose the clusters have a hierarchical structure, where each big cluster is partitioned into smaller sub-clusters with higher edge densities inside. In this case, either level of the clusters, the top-level big ones or the bottom-level small ones, can be considered as the ground truth, and it is a priori not clear which of them should be recovered. This ambiguity is resolved by specifying t: our algorithm recovers those clusters with in-cluster edge density higher than t and cross-cluster density lower than t. With a larger t, the algorithm operates at a higher resolution and detects small clusters with high density. By varying t, our algorithm can be turned into a method for *multi-resolution clustering* [1] which explores all levels of the cluster hierarchy. We leave this to future work. Importantly, the above example shows that it is generally impossible to uniquely determine the value of t from data.

III. The Generalized Stochastic Blockmodel

While our algorithm above is derived as a relaxation of ML estimator for the standard stochastic blockmodel, we establish performance guarantees in a much more general setting. The model is described below, which is defined by six parameters n, n_1, r, K, p and q.

Definition 1 (Generalized Stochastic Blockmodel (GSBM)): The $n = n_1 + n_2$ nodes are divided into two sets V_1 and V_2. The n_1 nodes in V_1 are further partitioned into r disjoint sets, which we will refer to as the "true" clusters. Let K be the minimum size *of a true cluster. If $p > q$, consider a random graph generated as follows: For every pair of nodes i, j that belong to the same true cluster, edge (i, j) is present in the*

[4]The constraints $y_{ii} = 1, \forall i$ are satisfied when there is no unaffiliated node.

graph with probability that is at least p*, while for every pair where the nodes are in different clusters the edge is present with probability* at most q*. The other* n_2 *nodes in* V_2 *are not in any cluster (we will call them unaffiliated nodes); for each* $i \in V_2$ *and* $j \in V_1 \cup V_2$*, there is an edge between the pair* i, j *with probability* at most q*. If* $p < q$*, then the graph is generated similarly as above, except that the probability of an in-cluster edge is* at most p*, while the probability of other edges is* at least q*. Note that it is implicit that* $r \geq 1$*,* $K \geq 1$ *and* $n \geq n_1 \geq rK$*.*

Definition 2 (Semi-Random GSBM): On a graph generated from GSBM with $p > q$ ($p < q$*, resp.), an adversary is allowed to arbitrarily (a) add (remove, resp.) edges between pairs of nodes in the same true cluster, and (b) remove (add, resp.) edges between pairs of nodes if they are in different clusters, or if at least one of them is an unaffiliated node in* V_2*.*

The **objective** is to find the underlying true clusters, given the graph generated from the semi-random GSBM.

The standard stochastic blockmodel/planted partition model is a special case of GSBM with $n_2 = 0, r \geq 2$, all cluster sizes equal to K, and all in-cluster and cross-cluster probabilities equal to p and q, respectively. GSBM generalizes the standard models as it allows for heterogeneity in the graph:

- p and q are lower and upper bounds instead of exact edge probabilities, so nodes can have different degrees; there may also exist nested clusters (cf. Section II-A).
- K is also a lower bound, so clusters can have different sizes.
- Unaffiliated nodes (nodes not in any cluster) are allowed.

GSBM removes many restrictions in the standard planted models and better models practical graphs.

The semi-random GSBM allows for further modeling power. It blends the worst case models, which are often overly pessimistic,[5] and the purely random graphs, which are extremely unstructured and have very special properties usually not possessed by real-world graphs [35]. This semi-random framework has been used and studied extensively in the computer science literature as a better model for real-world networks [11]–[13], as it allows for *non-randomness* in a graph. Note that the term "adversary" means *arbitrary* deviation from the random model (as long as it is allowed by the semi-random model), and it covers, but is not limited to, adversarial deviation. At first glance, the adversary seems to make the problem easier as it adds in-cluster edges and removes cross-cluster edges (when $p > q$). This is not necessarily the case. The adversary can significantly change some statistical properties of the random graph (e.g., alter spectral structure and node degrees, and create local optima by adding dense spots [12]), and foil algorithms that over-exploit such properties. For example, some spectral algorithms that work well on random models are proved to fail in the semi-random setting [4]. An algorithm that works well in the semi-random setting is likely to be more robust to model mis-specification in real-world applications [12]. As shown later, our algorithm processes this desired property.

[5]For example, the minimum graph bisection problem is NP-hard.

A. Special Cases

GSBM recovers as special cases many classical and widely studied models for clustered graphs, by considering different values for the parameters n_1, n_2, r, K, p and q. We classify these models into two categories based on the relation between p and q.

1) $p > q$**:** GSBM with $p > q$ models *homophily*, the tendency that individuals belonging to the same community tend to connect *more* than those in different communities. Special cases include:
   - **Planted Clique** [8]: $p = 1$, $r = 1$ (so $n_1 = K$) and $n_2 > 0$;
   - **Planted** r**-Disjoint-Cliques** [7], [9]: $p = 1$ and $r \geq 1$;
   - **Planted Dense Subgraph** [9]: $p < 1$, $r = 1$ and $n_2 > 0$;
   - **Stochastic Blockmodel, Planted Partition** [5], [6]: $n_2 = 0$, $r \geq 2$ with all cluster sizes equal to K. The special case with $r = 2$ can be call the Planted Bisection Model [5], [11].
2) $p < q$**:** This is complementary to the homophily case above. Special cases include:
   - **Planted Coloring** [12]: $q > p = 0$, $r \geq 2$, and $n_2 = 0$;
   - **Planted** r**-Cut, Planted Noisy Coloring** [11], [26]: $q > p \geq 0$, $r \geq 2$, and $n_2 = 0$.

Recall that the max-clique, max-cut, graph partition and graph coloring problems are all NP-hard in the worst case [5], [8], [10], [36]. The above "planted" variants of these problems are standard models for studying their average-case behavior.

In the next section, we provide performance guarantees for our algorithm under the semi-random GSBM. This implies guarantees for all the special cases above. We provide a detailed comparison with literature after presenting our results.

IV. Main Results: Performance Guarantees

In this section we study the performance of our algorithm under the semi-random GSBM and provide theoretical guarantees. We give a unified theorem, and then discuss its consequences for various special cases, and compare with literature. We also discuss how to estimate the parameter t in the special case of the standard stochastic blockmodel. We shall first consider the case with $p > q$. The $p < q$ case is similar and is discussed in Section IV-C.3. All proofs are postponed to Sections VI to IX.

A. A Monotone Lemma

Our optimization-based algorithm has a nice monotone property: adding/removing edges "aligned with" the optimal $\widehat{Y}$ (as is done by the adversary under the semi-random setting) cannot result in a different optimal solution. This is summarized in the following lemma.

Lemma 1: Suppose $p > q$ *and* $\widehat{Y}$ *is the unique optimal solution of (3)–(4) for a given* A *and* t*. If now we arbitrarily change some edges of* A *to obtain* $\widetilde{A}$*, by (a) choosing some edges such that* $\widehat{y}_{ij} = 1$ *but* $a_{ij} = 0$*, and making* $\widetilde{a}_{ij} = 1$*,*

and (b) choosing some edges where $\widehat{y}_{ij} = 0$ but $a_{ij} = 1$, and making $\widetilde{a}_{ij} = 0$. Then, $\widehat{Y}$ is also the unique optimal solution of (3)–(4) with $\widetilde{A}$ as the input and the same t.

The lemma shows that our algorithm is inherently robust under the semi-random model. In particular, the algorithm succeeds in recovering the true clusters on the semi-random GSBM as long as it succeeds on the GSBM with the same parameters. In the sequel, we therefore focus solely on the GSBM, with the understanding that any guarantee for it immediately implies a guarantee for the semi-random variant.

B. Main Theorem

Let Y^* be the matrix corresponding to the true clusters in the GSBM, i.e., $y^*_{ij} = 1$ if and only if $i, j \in V_1$ and they are in the same true cluster, and 0 otherwise. The theorem below establishes conditions under which our algorithm, specifically the convex program (3)–(4), yields this Y^* as the unique optimal solution with high probability (without any further need for rounding etc.).

Theorem 1: Suppose the graph A is generated according to the GSBM with $p > q$. If t in (5) is chosen to satisfy

$$\frac{1}{4}p + \frac{3}{4}q \le t \le \frac{3}{4}p + \frac{1}{4}q, \tag{6}$$

then Y^* is the unique optimal solution to the convex program (3)–(4) with probability at least $1 - 4n^{-8}$ provided

$$\frac{p-q}{\sqrt{p(1-q)}} \ge c_1 \max\left\{\frac{\sqrt{n}}{K}, \frac{\log^2 n}{\sqrt{K}}\right\}, \tag{7}$$

where c_1 is an absolute constant independent of p, q, K, r and n.

Our theorem quantifies the tradeoff between the four parameters governing the hardness of GSBM—the minimum in-cluster edge density p, the maximum across-cluster density q, the minimum cluster size K and the number of unaffiliated nodes $n_2 = n - n_1$—required for our algorithm to succeed, i.e., to recover the underlying true clustering without any error. Note that we can handle any values of p, q, n_2 and K as long as they satisfy the condition in the theorem; in particular, they are allowed to scale with n. Interestingly, the theorem does not have an explicit dependence on the number of clusters r (except for the requirement $rK \le n$). We note that by using a slightly stronger version of the spectral bound in Lemma 4 in the appendix (see [37]), it is possible to improve the $\log^2 n$ factor in (7) to $\sqrt{\log n}$. We omit such logarithmic improvement for reasons of space.

We now discuss the *tightness* of Theorem 1 in terms of these model parameters. When the minimum cluster size $K = \Theta(n)$, we have a near-matching converse result.

Theorem 2: Suppose all clusters have equal size K, and the in-cluster (cross-cluster, resp.) edge probabilities are uniformly p (q, resp.), with $K = \Theta(n)$ and $n_2 = \Theta(n_1)$. Under GSBM with $p > q$ and n sufficiently large, for any algorithm to correctly recover the clusters with probability at least $\frac{3}{4}$, we must have

$$\frac{p-q}{\sqrt{p(1-q)}} \ge c_2 \frac{1}{\sqrt{n}},$$

where c_2 is an absolute constant.

This theorem gives a necessary condition for *any* algorithm to succeed regardless of its computational complexity. It shows that Theorem 1 is optimal up to logarithmic factors for all values of p and q when $K = \Theta(n)$.

For smaller values of the minimum cluster size K, Theorem 1 requires K to be $\Omega(\sqrt{n})$ since the left hand side of (7) is less than 1. This lower-bound is achieved when p and q are both constants independent of n and K. There are reasons to believe that this requirement is unlikely to be improvable using polynomial-time algorithms. Indeed, the special case with $p = 1$ and $q = \frac{1}{2}$ corresponds to the classical planted clique problem [8]; finding a clique of size $K = o(\sqrt{n})$ is widely believed to be computationally hard even on average and has been used as a hard problem for cryptographic applications [38], [39].

For other values of p and q, no general and rigorous converse result exists. However, there is evidence suggesting that no other polynomial-time algorithm is likely to have better guarantees than our result in (7). The authors of [26] show, using non-rigorous but deep arguments from statistical physics, that recovering the clustering is impossible in polynomial time if $\frac{p-q}{\sqrt{p}} = o\left(\frac{\sqrt{n}}{K}\right)$. Moreover, the work in [27] shows that a large class of spectral algorithms fail under similar conditions. In view of these results, it is possible that our algorithm is order-wise optimal with respect to all polynomial-time algorithms for all values of p, q and K.

We give several further remarks regarding Theorem 1.

- A nice feature of our result is that we only need $p - q$ to be large *as compared to* $\sqrt{p}$; several other existing results (see Table I) require a lower bound (as a function of n and K) on $p - q$ itself. When K is $\Theta(n)$, we allow p and $p - q$ to be as small as $\Theta\left(\log^4(n)/n\right)$.
- The number of clusters r is allowed to grow rapidly with n—sometimes called the high-dimensional setting [20]. In particular, our algorithm can recover up to $r = \Theta(\sqrt{n})$ equal-sized clusters when $p - q = \Theta(1)$. Any algorithm with a better scaling would recover cliques of size $o(\sqrt{n})$, an unlikely task in polynomial time in light of the hardness of the planted clique problem discussed above.
- The number of unaffiliated nodes can be large, as many as $n_2 = \Theta(n) = \Theta(n_1^2)$, which is attained when $p - q, r$ are $\Theta(1)$ and the clusters have equal size. In other words, almost all the nodes can be unaffiliated, and this is true even when there are multiple clusters that are not cliques (i.e., $p < 1$).
- Not all existing methods can handle non-uniform edge probabilities and node degrees, which often require special treatment (see [3]). This issue is addressed seamlessly by our method by definition of GSBM.

C. Consequences and Comparison With Literature

In this subsection we discuss the consequences of Theorem 1 for specific planted problems and compare with existing work. Our results match the best existing results in all cases (up to logarithmic factors), and in many important settings lead to order-wise stronger guarantees.

1) Standard Stochastic Blockmodel (a.k.a. Planted Partition Model): This model assumes that all clusters have the same size K with no unaffiliated nodes ($n_2 = 0$) and $p > q$. We compared our result to past approaches and theoretical results in Table I: For general values of p, q and K, our result has the scaling $p - q = \Omega\left(\frac{\sqrt{pn}}{K}\right)$ and $p = \Omega\left(\frac{n}{K^2}\right)$, which improves on all existing results by polynomial factors. This means that we can handle much noisier and sparser graphs, especially when the number of clusters $r = n/K$ is growing.

2) Planted r-Disjoint-Cliques Problem: Here the task is to find a set of r disjoint cliques, each of size at least K, that have been planted in an Erdos-Renyi random graphs $G(n, q)$. Setting $p = 1$ in Theorem 1, we obtain the following guarantee for this problem.

Corollary 1: For the planted r-disjoint-cliques problem, the formulation (3)-(5) *with t chosen according to Theorem 1 finds the hidden cliques with probability at least $1 - 4n^{-8}$ provided*

$$1 - q \geq c_3 \max\left\{\frac{n}{K^2}, \frac{\log^4 n}{K}\right\},$$

where c_3 is an absolute constant.

In the regime where r is allowed to scale with n and q is bounded away from zero, the best previous results are given in [9] with $1 - q = \Omega(\frac{rn}{K^2})$ and in [22] with $1 - q = \Omega(\frac{\sqrt{n}}{K})$. Corollary 1 is stronger than both of them for large r. In the special case with $r = 1$ and $q = 1/2$, which is the standard planted clique problem, the corollary guarantees recovery for the clique size $K = \Omega(\sqrt{n})$, matching the best known bound [8].

3) The $p < q$ Case: Given a graph A generated from the semi-random GSBM with in/cross-cluster densities $p < q$, we can run our algorithm on the graph $A' = \mathbf{1}\mathbf{1}^\top - A$, where $\mathbf{1}\mathbf{1}^\top$ is the all-one matrix. Note that A' can be considered as generated from GSBM with in/cross-cluster densities $p' = 1 - p$ and $q' = 1 - q$, where $p' > q'$. With this reduction, Theorem 1 immediately yields the following guarantee.

Corollary 2: Under the semi-random GSBM with $p < q$, the formulation (3)-(5) *applied to $\mathbf{1}\mathbf{1}^\top - A$ with t satisfying*

$$\frac{3}{4}p + \frac{1}{4}q \leq 1 - t \leq \frac{1}{4}p + \frac{3}{4}q$$

finds the true clustering with probability $1 - 4n^{-8}$ provided

$$q - p \geq c_3\sqrt{(1-p)q}\max\left\{\frac{\sqrt{n}}{K}, \frac{\log^2 n}{\sqrt{K}}\right\},$$

where c_3 is an absolute constant.

This corollary implies guarantees for the planted coloring problem [10] and the planted r-cut [11] (a.k.a. planted noisy coloring [26]) problem. We are not aware of any exiting work that explicitly considers the GSBM with $p < q$ in its general form (i.e., $n_2 > 0$, $1 > q > p > 0$, and $K = o(n)$ with potential non-random edges). However, since any guarantee for GSBM with $p > q$ implies a guarantee for GSBM with $p < q$, Table I provides a comparison with existing work when $n_2 = 0$ and the edge probabilities and cluster sizes are uniform. Again our guarantee outperforms all existing ones.

4) Planted Coloring Problems: This is a special case of the above setting, where $p = 0$, $n_2 = 0$ and the goal is to find the r planted groups of colored nodes with no edge between nodes with the same color. The best existing result $q = \Omega\left(\frac{n}{K^2} + \frac{\log n}{K}\right)$ is achieved by various algorithms; see [4], [10]. By Corollary 2, our algorithm succeeds when $q = \Omega\left(\frac{n}{K^2} + \frac{\log^4 n}{K}\right)$. We match the best existing algorithms for $K = O(n/\log^4(n))$, and are off by a few log factors for larger K.

5) Clustering Partially Observed Graphs: In many applications the pairwise relations in the graph are partially observed, meaning that the values of A_{ij} are known only for a subset of the pairs (i, j), and information of other pairs is impossible or too expensive to obtain [32], [40]. A standard and natural model for this setting is as follows: after the graph A is generated according to the GSBM with edge densities p and q, each entry of A is erased (i.e., unobserved) independently with probability $1 - s$, so $s \in [0, 1]$ is the observation probability. One possible approach is to set to zero all the entries of A that are unobserved, and apply our algorithm to the zero-imputed graph A''. Note that A'' can be considered as generated from the GSBM with in/cross-cluster densities equal to ps and qs, respectively. Theorem 1 is powerful enough to imply the following strong guarantee for this simple approach.

Corollary 3: Under the above setting with $p > q$, the formulation (3)-(5) *applied to A'' with t satisfying*

$$\frac{1}{4}ps + \frac{3}{4}qs \leq t \leq \frac{3}{4}ps + \frac{1}{4}qs$$

finds the true clustering with probability $1 - 4n^{-8}$ provided

$$(p - q)\sqrt{\frac{s}{p}} \geq c_4 \max\left\{\frac{\sqrt{n}}{K}, \frac{\log^2 n}{\sqrt{K}}\right\},$$

where c_4 is an absolute constant.

The work in [32] considers the special case with $p = 1 - q > 1/2$. Their algorithm explicitly handles unobserved pairs and requires the condition $(2p - 1)\sqrt{s} \gtrsim \frac{\sqrt{n}\log n}{K}$, which is the best known result in this setting. Corollary 3 matches this result up to at most a logarithmic factor, and in addition applies to settings with more general values of p and q. The algorithm proposed in [21] also imputes unobserved pairs with zeros and requires $(p - q)s \gtrsim \max\{\frac{\sqrt{n}}{K}, \sqrt{\frac{\log n}{K}}\}$. Corollary 3 is order-wise better whenever $K \lesssim n/\log^4 n$.

D. Estimating t in Special Cases

We argued in Section II-A that specifying t in a completely data-driven way is ill-posed for the general GSBM, e.g., when the clusters have a hierarchical structure. However, for some cases this can be done reliably with strong guarantees. Consider the standard stochastic blockmodel, where all the $r = n/K$ clusters have the same size K, the edge probabilities are uniform (i.e., equal to p within clusters and q across clusters, with $p > q$), and there are no unaffiliated nodes ($n_2 = 0$) or non-random edges. Without lost of generality, we may re-label the nodes such that the l-th cluster has nodes $(l - 1)K + 1, (l - 1)K + 2, \ldots, lK$. Observe that the matrix

Algorithm 2 Estimation of t From Data

1) Compute and sort the eigenvalues of A, denoted as $\hat{\lambda}_1 \geq \hat{\lambda}_2 \geq \ldots \geq \hat{\lambda}_n$.
2) Let $\hat{r} = \arg\max_{i=2,\ldots,n-1}(\hat{\lambda}_i - \hat{\lambda}_{i+1})$ (break ties arbitrarily). Set $\hat{K} = n/\hat{r}$.
3) Set $\hat{p} = \frac{\hat{K}\hat{\lambda}_1 + (n-\hat{K})\hat{\lambda}_2 - n}{n(\hat{K}-1)}$, $\hat{q} = \frac{\hat{\lambda}_1 - \hat{\lambda}_2}{n}$ and $t = \frac{\hat{p}+\hat{q}}{2}$.

$\bar{A} := \mathbb{E}[A] - (1-p)I$ is a matrix with blocks of p and q's,[6] and therefore can be written as

$$\bar{A} = \mathbf{1}\mathbf{1}^\top \otimes B,$$

where $\mathbf{1}$ is the all one vector in $\mathbb{R}^K$, and $B \in \mathbb{R}^{r\times r}$ equals p on the diagonal and q elsewhere. In words, $\bar{A}$ is the Kronecker product of a $K \times K$ all-one matrix $\mathbf{1}\mathbf{1}^\top$ and an $r \times r$ circulant matrix B. The matrix $\mathbf{1}\mathbf{1}^\top$ has only one non-zero eigenvalue K, and the matrix B has eigenvalues $(p-q)+rq$ and $p-q$ with multiplicities 1 and $r-1$, respectively. The eigenvalues of $\bar{A}$ are the products of the eigenvalues of $\mathbf{1}\mathbf{1}^\top$ and B. Since $n = Kr$, it follows that the eigenvalues of $\mathbb{E}[A] = \bar{A} + (1-p)I$ are:

$$\begin{cases} K(p-q) + nq + (1-p) & \text{with multiplicity } 1, \\ K(p-q) + (1-p) & \text{with multiplicity } r-1, \\ 1-p & \text{with multiplicity } n-r; \end{cases}$$

see [17] for a similar derivation. Given these eigenvalues of $\mathbb{E}[A]$, we can compute the values of r and K as there is a gap between the r-th and $(r+1)$-th eigenvalues, and then solve for p, q (and therefore t) using the first two eigenvalues. In practice, we use the observed matrix A instead of $\mathbb{E}[A]$; see Algorithm 2.

The following theorem guarantees that the estimation errors are sufficiently small.

Theorem 3: Under the standard stochastic blockmodel and the condition (7) *in Theorem 1, the parameters estimated in Algorithm 2 satisfy the following with probability at least* $1 - 4n^{-8}$*, where* c_4 *is an absolute positive constant:*

$$\hat{K} = K,$$
$$\max\left\{\left|\hat{p}-p\right|, \left|\hat{q}-q\right|\right\} \leq c_4 \frac{\sqrt{p(1-q)n}}{K},$$
$$t \in \left[\frac{1}{4}p + \frac{3}{4}q, \frac{3}{4}p + \frac{1}{4}q\right].$$

In particular, the estimated t satisfies the condition (6) in Theorem 1. The above theorem also ensures that Algorithm 2 is a consistent estimator of the parameters p and q when condition (7) is satisfied, which may be a result of independent interest. Combining Theorem 1 and Theorem 3, we obtain a complete algorithm that is guaranteed to find the clusters for the standard stochastic blockmodel under the condition (7), without any knowledge of the parameters of the underlying generative model.

[6]Recall that we use the convention $a_{ii} = 1$.

Algorithm 3 ALM Method for the Program (8) of Minimizing Nuclear Norm Plus Weighted ℓ_1 Norm

Input: $A, C \in \mathbb{R}^{n\times n}$.
Initialize: $M^{(0)} = 0$; $Y^{(0)} = 0$; $S^{(0)} = 0$; $\mu_0 > 0$; $\alpha > 1$; $k = 0$, $\lambda = \frac{1}{48\sqrt{n}}$.
while not converge **do**
$(U, \Sigma, V) = \mathrm{SVD}(A - S^{(k)} + \mu_k^{-1}M^{(k)})$.
$\bar{Y}^{(k+1)} = U\mathcal{S}_{\mu_k^{-1}}(\Sigma)V$.
For all (i, j), $y_{ij}^{(k+1)} = \max\left\{\min\left\{\bar{Y}_{ij}^{(k+1)}, 1\right\}, 0\right\}$.
$S^{(k+1)} = \mathcal{S}_{\mu_k^{-1}\lambda C}(A - Y^{(k+1)} + \mu_k^{-1}M^{(k)})$.
$M^{(k+1)} = M^{(k)} + \mu_k(A - Y^{(k+1)} - S^{(k+1)})$.
$\mu_{k+1} = \alpha\mu_k$, $k = k+1$.
end while
Return $Y^{(k+1)}$, $S^{(k+1)}$.

V. Empirical Results

In this section we discuss implementation issues of our algorithm, and provide empirical results on synthetic and real-world datasets.

A. Implementation Issues

The convex program (3)–(4) can be solved using a general purpose SDP solver, but this method does not scale well to problems with more than a few hundred nodes. To facilitate a fast and efficient solution, we propose to use a family of first-order algorithms called the Augmented Lagrange Multiplier (ALM) method. Note that the program (3)–(4) can be rewritten as

$$\begin{aligned} \min_{Y,S\in\mathbb{R}^{n\times n}} \quad & \lambda\|C \circ S\|_1 + \|Y\|_* \\ \text{s.t} \quad & Y + S = A, \\ & 0 \leq y_{ij} \leq 1, \forall i, j, \end{aligned} \tag{8}$$

where $\lambda := \frac{1}{48\sqrt{n}}$, the matrix $C \in \mathbb{R}^{n\times n}$ satisfies $c_{ij} = c_{\mathcal{A}}$ if $a_{ij} = 1$ and $c_{ij} = c_{\mathcal{A}^c}$ otherwise, and $\circ$ denotes the element-wise product. This problem can be recognized as a weighted version of the standard convex formulation of the low-rank and sparse matrix decomposition problem [28], [29], of which the numerical solution has been well studied. We adapt the ALM method in [41] to the above problem as given in Algorithm 3. Here $\mathcal{S}_X(\cdot) : \mathbb{R}^{n\times n} \mapsto \mathbb{R}^{n\times n}$ is the element-wise weighted soft-thresholding operator, defined as

$$(\mathcal{S}_X(M))_{ij} = \begin{cases} m_{ij} - x_{ij}, & \text{if } m_{ij} > x_{ij} \\ m_{ij} + x_{ij}, & \text{if } m_{ij} < -x_{ij} \\ 0, & \text{otherwise,} \end{cases}$$

for any matrices $M, X \in \mathbb{R}^{n\times n}$. In other words, it shrinks each entry of M towards zero by x_{ij}. The unweighted version $\mathcal{S}_\epsilon(\cdot) := \mathcal{S}_{\epsilon I}(\cdot)$ is also used. The parameters of the algorithm are set as $\mu_0 = 1.25/\|A\|$ and $\alpha = 1.5$ to suggested by [41]. Following [41], it can be shown that the ALM method is guaranteed to converge to a global optimal solution.

While [41] does not prove a convergence rate for the ALM method, it is observed there that it converges Q-linearly.

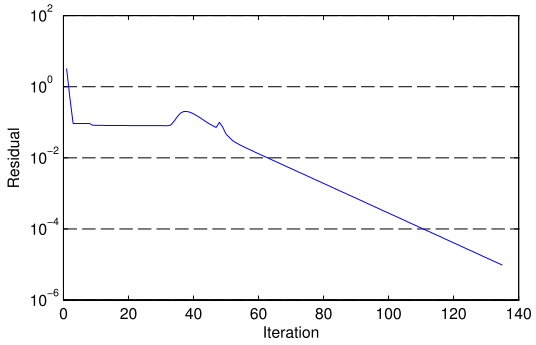


Fig. 1. Convergence of the ALM method. The figure shows the residual $\|A - Y^{(k)} - S^{(k)}\|_F / \|A\|_F$ at each iteration. The plot is generated under the setting with $n = 1000$ nodes, $r = 5$ clusters with equal size $K = 200$, and $p = 0.35$, $q = 0.1$.

We observe a similar behavior, as shown in Figure 1. In the subsequent simulations, we use $\|A - Y^{(k)} - S^{(k)}\|_F / \|A\|_F \leq 10^{-2}$ as the stopping criterion, so the number of iteration needed is usually small. The main bottleneck of the algorithm is computing the SVD in each iteration. Therefore, the time complexity of the algorithm is roughly the time for one SVD multiplied by the number of iterations. This can be compared with spectral clustering, which requires one SVD. The memory requirement of the ALM algorithm is $\Theta(n^2)$, i.e., the same order as the space needed to store the graph. It is possible to improve the space and time complexity by various approaches, such as only storing sparse and low-rank matrices and computing the first few singular values/vectors instead of a full SVD; see [41] for more discussion on implementation details.

B. Simulations

We perform experiments on synthetic data, and compare with other methods. We generate a graph using the stochastic blockmodel with $n = 1000$ nodes, $r = 5$ clusters with equal size $K = 200$, and $p, q \in [0, 1]$. We apply our method to the graph, where we pick t using Algorithm 2 and solve the optimization problem using Algorithm 3. Due to numerical accuracy, the output $\hat{Y}$ of our algorithm may not be strictly integer, so we do the following simple rounding: compute the mean $\bar{y}$ of the entries of $\hat{Y}$, and round each entry of $\hat{Y}$ to 1 if it is greater than $\bar{y}$, and 0 otherwise. We measure the error by $\|Y^* - \text{round}(\hat{Y})\|_1$, which equals the number of misclassified pairs. We say our method succeeds if it misclassifies less than 0.1% of the pairs.

For comparison, we consider three alternative methods: (1) Single-Linkage clustering (SLINK) [42], which is a hierarchical clustering method that merges the most similar clusters in each iteration. We use the difference of neighbors, namely $\|A_{i\cdot} - A_{j\cdot}\|_1$, as the distance measure of nodes i and j, and terminate when SLINK finds a clustering with $r = 5$ clusters. (2) A spectral clustering method [43], where we run SLINK on the top $r = 5$ singular vectors of A. (3) The low-rank-plus-sparse approach [21], [32], followed by the rounding scheme described in the last paragraph. Note the first two methods assume knowledge of the number of clusters r, which is not available to our method.

For each value of q, we find the smallest p for which a method succeeds, and average over 20 trials. The results are shown in Figure 2(a), where the area above each curve corresponds to the range of feasible (p, q) for each method. It can been seen that our method outperforms all others, in that we succeed for a strictly larger range of (p, q). Figure 2(b) shows more detailed results for sparse graphs ($p \leq 0.3, q \leq 0.1$), for which SLINK and the low-rank-plus-sparse approach completely fail, while our method significantly outperforms the spectral method, the only alternative method that works in this regime. The running time of each method is shown in Figure 2 (c). Our approach and the low-rank-plus-sparse approach (both based on convex optimization) require more computational time than the simpler spectral method and SLINK. This suggests a tradeoff between the statistical and computational performance of clustering algorithms.

C. Real-World Collaboration Graph

We evaluate our method on the NIPS Conference Papers Vol. 0-12 Dataset.[7] It contains the authorship relation of 2037 authors and 1740 papers. We use this dataset to generate a 2037×2037 graph of the authors by connecting co-authors; that is, we place an edge between a pair of authors if they have written at least one NIPS paper together. This is a sparse graph with an overall edge density of 0.002.

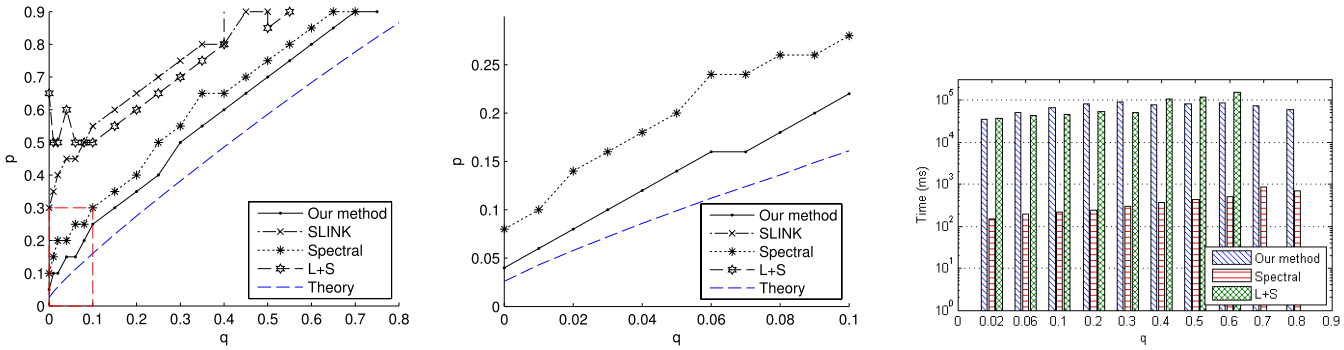
We apply the four methods to this graph and compare their performance. For fairness, we force all methods to partition the authors into $r = 8$ clusters as follows: the SLINK and spectral algorithms are the same as in the previous sub-section; for our method and the low-rank-plus-sparse approach, we apply SLINK to their output $\hat{Y}$ with $\|\hat{Y}_{i\cdot} - \hat{Y}_{j\cdot}\|_1$ as the distance measure to obtain 8 clusters; the parameter t for our method is estimated using Algorithm 2 with $\hat{r}$ fixed to 8. We measure the quality of the solutions by computing the in-cluster and cross-cluster edge densities, which are shown in Table II. The clustering produced by our method has higher in-cluster density and lower cross-cluster density.

VI. Proof of Lemma 1

In this section we establish the monotone lemma. Set $\lambda = \frac{1}{48\sqrt{n}}$. Define $\Omega_+ = \{(i, j) : a_{ij} = 0, \tilde{a}_{ij} = 1\}$ and $\Omega_- = \{(i, j) : a_{ij} = 1, \tilde{a}_{ij} = 0\}$. Let $Y \neq \widehat{Y}$ be an arbitrary alternative feasible solution obeying $0 \leq y_{ij} \leq 1, \forall i, j$. By optimality of $\widehat{Y}$ to the original program, we have

$$\left(c_{\mathcal{A}} \sum_{a_{ij}=1} \hat{y}_{ij} - c_{\mathcal{A}^c} \sum_{a_{ij}=0} \hat{y}_{ij} \right) - \frac{1}{\lambda} \left\| \widehat{Y} \right\|_* > \left(c_{\mathcal{A}} \sum_{a_{ij}=1} y_{ij} - c_{\mathcal{A}^c} \sum_{a_{ij}=0} y_{ij} \right) - \frac{1}{\lambda} \|Y\|_* . \quad (9)$$

[7] Available at http://www.cs.nyu.edu/~roweis/data.html



(a) (b) (c)

Fig. 2. Comparison of our method with Single-Linkage clustering (SLINK), spectral clustering, and low-rank-plus-sparse (L+S) approach. (a) The area above each curve is the values of (p, q) for which a method successfully recovers the underlying true clusters. The dash line corresponds to the bound $p - q \geq \sqrt{p(1-q)n}/K$ predicted by our theoretical result in Theorem 1. (b) More detailed results for the area in the box in (a), corresponding to sparse graphs. (c) Running times (in milliseconds) for each methods running on different values of q and the smallest p for which the method succeeds. A missing bar means a method fails for any p. The running time of the SLINK method is negligible compared to the other methods and is thus displayed in the plot. The experiments are conducted on synthetic data with $n = 1000$ nodes and $r = 5$ clusters with equal size $K = 200$, using a computer with a Pentium Dual-Core 3.2GHz CPU and 4.00 GB memory.

TABLE II

CLUSTERING QUALITY ON THE NIPS DATASETS

	In-Cluster edge density	Cross-cluster edge density
Our method	109×10^{-4}	1.83×10^{-4}
SLINK	26×10^{-4}	3.42×10^{-4}
Spectral	64×10^{-4}	1.88×10^{-4}
L+S	86×10^{-4}	6.07×10^{-4}

Next, by definition of $\tilde{A}$, Ω_+ and Ω_-, we have

$$\left(c_{\mathcal{A}}\sum_{\tilde{a}_{ij}=1}\hat{y}_{ij} - c_{\mathcal{A}^c}\sum_{\tilde{a}_{ij}=0}\hat{y}_{ij}\right) - \left(c_{\mathcal{A}}\sum_{a_{ij}=1}\hat{y}_{ij} - c_{\mathcal{A}^c}\sum_{a_{ij}=0}\hat{y}_{ij}\right) = \sum_{(i,j)\in\Omega_+}(c_{\mathcal{A}} + c_{\mathcal{A}^c}); \tag{10}$$

and

$$\begin{aligned}&\left(c_{\mathcal{A}}\sum_{a_{ij}=1}y_{ij} - c_{\mathcal{A}^c}\sum_{a_{ij}=0}y_{ij}\right) - \left(c_{\mathcal{A}}\sum_{\tilde{a}_{ij}=1}y_{ij} - c_{\mathcal{A}^c}\sum_{\tilde{a}_{ij}=0}y_{ij}\right)\\ &\quad= (c_{\mathcal{A}} + c_{\mathcal{A}^c})\sum_{(i,j)\in\Omega_-}y_{ij} - (c_{\mathcal{A}} + c_{\mathcal{A}^c})\sum_{(i,j)\in\Omega_+}y_{ij}\\ &\quad\geq -\sum_{(i,j)\in\Omega_+}(c_{\mathcal{A}} + c_{\mathcal{A}^c}),\end{aligned} \tag{11}$$

where we use $0 \leq y_{ij} \leq 1$ for all (i, j) in the last inequality. Summing the L.H.S. and R.H.S. of (9)–(11) establishes that

$$\left(c_{\mathcal{A}}\sum_{\tilde{a}_{ij}=1}\hat{y}_{ij} - c_{\mathcal{A}^c}\sum_{\tilde{a}_{ij}=0}\hat{y}_{ij}\right) - \frac{1}{\lambda}\left\|\widehat{Y}\right\|_* > \left(c_{\mathcal{A}}\sum_{\tilde{a}_{ij}=1}y_{ij} - c_{\mathcal{A}^c}\sum_{\tilde{a}_{ij}=0}y_{ij}\right) - \frac{1}{\lambda}\|Y\|_*.$$

Since Y is arbitrary, we conclude that $\widehat{Y}$ is the unique optimal solution to the modified program.

VII. PROOF OF THEOREM 1

We prove our main theorem in this section. In the remainder of the paper, *with high probability* (w.h.p.) means with probability at least $1 - 4n^{-12}$. The proof consists of three main steps, which we elaborate below.

A. Step 1: Reduction to Homogeneous Edge Probabilities

We show that it suffices to assume that the in-cluster edge probability is uniformly p, and the across-cluster edge probability is uniformly q. In the heterogeneous model, suppose an edge is placed between nodes i and j with probability p_{ij} if they are in the same cluster, where $p_{ij} \geq p$. This is equivalent to the following two-step model: first flip a coin with head probability p, and add the edge if it is head; if it is tail, then flip another coin and add the edge with probability $\frac{p_{ij}-p}{1-p}$. By the monotone property in Lemma 1, we know that if our convex program succeeds on the graph generated in the first step, then it also succeeds for the second step, because more in-cluster edges are added. Similarly, an across-cluster edge with probability $q_{ij} \leq q$ can be generated equivalently as follows: (1) add an edge with probability q; (2) if an edge is added in the first step, remove it with probability $\frac{q-q_{ij}}{q}$. Monotonicity can then be applied. Therefore, heterogeneous edge probabilities only make the probability of success higher, and thus we only need to prove the homogeneous case.

B. Step 2: Optimality Condition

We need some additional notation. Both m_{ij} and $(M)_{ij}$ denote the (i, j)-th entry of the matrix M, and $\langle M, N\rangle := \text{trace}(M^\top N)$ is the inner product between two matrices M and N with the same size. Four matrix norms are used: the spectral norm $\|M\|$ (the largest singular value of M), the nuclear norm $\|M\|_*$ (the sum of the singular values of M), the matrix ℓ_∞ norm $\|M\|_\infty := \max_{i,j}|m_{ij}|$, and the matrix ℓ_1 norm $\|M\|_1 := \sum_{i,j}|m_{ij}|$. We denote the singular value decomposition of Y^* (notice Y^* is symmetric

and positive definite) by $U_0\Sigma_0U_0^\top$. For any matrix M, we define $P_T(M) := U_0U_0^\top M + MU_0U_0^\top - U_0U_0^\top MU_0U_0^\top$. For a set Ω of matrix indices, let $P_\Omega(M)$ be the matrix obtained by setting the entries of M outside Ω to zero, and we use $\sum_\Omega$ as a shorthand of $\sum_{(i,j)\in\Omega}$. Define the sets $\mathcal{A} := \text{support}(A)$ and $R := \text{support}(Y^*) = \text{support}(U_0U_0^\top)$.

The true cluster matrix Y^* is an optimal solution to the program (3)–(4) if

$$\lambda c_{\mathcal{A}} \sum_{\mathcal{A}}(y^*_{ij} - y_{ij}) - \lambda c_{\mathcal{A}^c} \sum_{\mathcal{A}^c}(y^*_{ij} - y_{ij}) - (\|Y^*\|_* - \|Y\|_*) \geq 0 \tag{12}$$

for all feasible Y obeying (4). Suppose there is a matrix W that satisfies

$$\|W\| \leq 1,\ P_T(W) = 0. \tag{13}$$

The matrix $U_0U_0^\top + W$ is a subgradient of $f(X) = \|X\|_*$ at $X = Y^*$, so $\|Y\|_* - \|Y^*\|_* \geq \langle U_0U_0^\top + W, Y - Y^*\rangle$ for all Y. Then, we see that (12) is implied by

$$\begin{aligned}&\lambda c_{\mathcal{A}}\sum_{\mathcal{A}}(y^*_{ij} - y_{ij}) - \lambda c_{\mathcal{A}^c}\sum_{\mathcal{A}^c}(y^*_{ij} - y_{ij}) + \langle U_0U_0^\top + W, Y - Y^*\rangle \\ &\geq 0, \quad \forall Y \in \left\{X : 0 \leq X_{ij} \leq 1, \forall (i,j)\right\}.\end{aligned} \tag{14}$$

The above inequality holds in particular for any feasible Y of the form $Y = Y^* - e_ie_j^\top$ with $(i,j) \in R$ or $Y = Y^* + e_ie_j^\top$ with $(i,j) \in R^c$. This leads to the following element-wise inequalities:

$$\begin{aligned} -\lambda c_{\mathcal{A}^c} - (U_0U_0^\top + W)_{ij} &\geq 0, \quad \forall (i,j) \in R \cap \mathcal{A}^c, \\ -\lambda c_{\mathcal{A}} + w_{ij} &\geq 0, \quad \forall (i,j) \in R^c \cap \mathcal{A}, \\ \lambda c_{\mathcal{A}} - (U_0U_0^\top + W)_{ij} &\geq 0, \quad \forall (i,j) \in R \cap \mathcal{A}, \\ \lambda c_{\mathcal{A}^c} + w_{ij} &\geq 0, \quad \forall (i,j) \in R^c \cap \mathcal{A}^c. \end{aligned} \tag{15}$$

It is easy to see that these inequalities are actually equivalent to (14), so together with (13) they form a sufficient condition for the optimality of Y^*.

Finding a "dual certificate" W obeying the exact conditions (13) and (15) is difficult, and does not guarantee uniqueness of the optimum. Instead, we consider an alternative sufficient condition that only requires a W that *approximately* satisfies the exact conditions. This is done in Proposition 1 below (proved in Section VII-D), which significantly simplifies the construction of W. Note that condition (b) in the proposition is a relaxation of the equality in (13), whereas condition (c) tightens (15). Setting $\epsilon = 0$ and changing equalities to inequalities in the proposition recover the exact conditions.

Proposition 1: Y^* is the unique optimal solution to the program (3)–(4)*, if there exists a matrix $W \in \mathbb{R}^{n\times n}$ and a number $0 < \epsilon < 1$ that satisfy the following conditions: (a) $\|W\| \leq 1$, (b) $\|P_T(W)\|_\infty \leq \frac{\epsilon}{2}\lambda \min\{c_{\mathcal{A}^c}, c_{\mathcal{A}}\}$, and (c)*

$$\begin{aligned} -(1+\epsilon)\lambda c_{\mathcal{A}^c} - (U_0U_0^\top + W)_{ij} &= 0, \quad \forall (i,j) \in R \cap \mathcal{A}^c, \\ -(1+\epsilon)\lambda c_{\mathcal{A}} + w_{ij} &= 0, \quad \forall (i,j) \in R^c \cap \mathcal{A}, \\ (1-\epsilon)\lambda c_{\mathcal{A}} - (U_0U_0^\top + W)_{ij} &\geq 0, \quad \forall (i,j) \in R \cap \mathcal{A}, \\ (1-\epsilon)\lambda c_{\mathcal{A}^c} + w_{ij} &\geq 0, \quad \forall (i,j) \in R^c \cap \mathcal{A}^c. \end{aligned}$$

C. Step 3: Constructing W

We build a W that satisfies the conditions in Proposition 1 w.h.p. We use $\mathbf{1}$ to denote the all-one column vector in $\mathbb{R}^n$, so $\mathbf{1}\mathbf{1}^\top$ is the all-one $n \times n$ matrix. Let $\mathcal{H} := \{(i,i), i = 1,\ldots,n\}$ be the set of diagonal entries. For an ϵ to be specified later, we define $W = W_1 + W_2 + W_3 + W_4$ with W_i given by

$$\begin{aligned} W_1 &= -P_{R\cap\mathcal{A}^c}(U_0U_0^\top) + \frac{1-p}{p}P_{R\cap\mathcal{A}}(U_0U_0^\top), \\ W_2 &= (1+\epsilon)\lambda c_{\mathcal{A}^c}\left[-P_{R\cap\mathcal{A}^c}(\mathbf{1}\mathbf{1}^\top) + \frac{1-p}{p}P_{R\cap\mathcal{A}}(\mathbf{1}\mathbf{1}^\top)\right], \\ W_3 &= (1+\epsilon)\lambda c_{\mathcal{A}}\left[P_{(R^c\cap\mathcal{H}^c)\cap\mathcal{A}}(\mathbf{1}\mathbf{1}^\top) - \frac{q}{1-q}P_{(R^c\cap\mathcal{H}^c)\cap\mathcal{A}^c}(\mathbf{1}\mathbf{1}^\top)\right], \\ W_4 &= (1+\epsilon)\lambda c_{\mathcal{A}}P_{R^c}(I), \end{aligned}$$

where I is the identity matrix. We briefly explain the ideas behind the construction. Each of the matrices W_1, W_2 and W_3 is the sum of two terms. The first term is derived from the equalities in condition (c) in Proposition 1. The second term is constructed in such a way that each W_i is a zero-mean random matrix (due to the randomness in the set $\mathcal{A}$), so it is likely to have small norms and satisfy conditions (a) and (b). The matrix W_4 accounts for the unaffiliated nodes. In particular, it is a diagonal matrix with $(W_4)_{ii}$ being non-zero if and only if $i \in V_2$.

The following proposition (proved in Section VII-E) shows that W indeed satisfies all the desired conditions w.h.p., hence establishing Theorem 1.

Proposition 2: Under the conditions in Theorem 1, W constructed above satisfies the conditions (a)–(c) in Proposition 1 w.h.p. with

$$\epsilon := \frac{48}{\sqrt{t(1-t)}}\max\left\{\frac{\sqrt{n}}{K}, \sqrt{\frac{\log^4 n}{K}}\right\}.$$

D. Proof of Proposition 1 (Optimality Condition)

Let $P_{T^\perp}(W) := W - P_T(W)$. Consider any feasible solution Y and let $D := Y - Y^*$. The difference between the objective values of Y and Y^* satisfies

$$\begin{aligned} (*) &:= c_{\mathcal{A}}\sum_{\mathcal{A}} d_{ij} - c_{\mathcal{A}^c}\sum_{\mathcal{A}^c} d_{ij} - \frac{1}{\lambda}\left\|Y^* + D\right\|_* + \frac{1}{\lambda}\left\|Y^*\right\|_* \\ &\leq c_{\mathcal{A}}\sum_{\mathcal{A}} d_{ij} - c_{\mathcal{A}^c}\sum_{\mathcal{A}^c} d_{ij} - \frac{1}{\lambda}\left\langle U_0U_0^\top + P_{T^\perp}(W), D\right\rangle \\ &= c_{\mathcal{A}}\sum_{\mathcal{A}} d_{ij} - c_{\mathcal{A}^c}\sum_{\mathcal{A}^c} d_{ij} - \frac{1}{\lambda}\left\langle U_0U_0^\top + W, D\right\rangle + \frac{1}{\lambda}\langle P_T W, D\rangle, \end{aligned} \tag{16}$$

where in the inequality we use the fact that $U_0U_0^\top + P_{T^\perp}(W)$ is a subgradient of $\|Y\|_*$ at Y^*, a consequence of condition (a) in the proposition and $\|P_{T^\perp}(W)\| \leq \|W\|$. We substitute the condition (c) into the third term in (16) to

obtain

$$\begin{aligned}(*) &\le \epsilon c_{\mathcal{A}} \sum_{R\cap\mathcal{A}} d_{ij} - \epsilon c_{\mathcal{A}^c} \sum_{R^c\cap\mathcal{A}^c} d_{ij} + \epsilon c_{\mathcal{A}^c} \sum_{R\cap\mathcal{A}^c} d_{ij} \\ &\quad -\epsilon c_{\mathcal{A}} \sum_{R^c\cap\mathcal{A}} d_{ij} + \frac{1}{\lambda}\langle P_T W, D\rangle \\ &\le -\epsilon \min\{c_{\mathcal{A}}, c_{\mathcal{A}^c}\}\|D\|_1 + \frac{1}{\lambda}\langle P_T W, D\rangle,\end{aligned}$$

where we used the fact that $d_{ij} \le 0$ for $(i, j) \in R$ and $d_{ij} \ge 0$ for $(i, j) \in R^c$ since $Y = Y^* + D$ satisfies (4). Applying condition (b) yields

$$\begin{aligned}(*) &\le -\epsilon \min\{c_{\mathcal{A}^c}, c_{\mathcal{A}}\}\|D\|_1 + \frac{1}{\lambda}\|P_T W\|_\infty \|D\|_1 \\ &\le -\frac{\epsilon}{2}\min\{c_{\mathcal{A}^c}, c_{\mathcal{A}}\}\|D\|_1.\end{aligned}$$

The last R.H.S. is strictly negative whenever $D \neq 0$. This proves that Y^* is the unique optimal solution.

E. Proof of Proposition 2

We show that W constructed in Section VII-C satisfies the conditions in Proposition 1 w.h.p. We need two technical lemmas. First notice that the conditions (6) and (7) in Theorem 1 imply bounds on various quantities.

Lemma 2: Under conditions (6) and (7) in Theorem 1, we have $p(1-q) \ge t(1-t) \ge c\max\left\{\frac{n}{K^2}, \frac{\log^4 n}{K}\right\}$ *and* $\epsilon < \frac{1}{2}$.

Proof: Since $1 > t > 0$, we have $t(1-t) \ge \frac{1}{2}\min\{t, 1-t\}$. Under condition (6) on t, we further have $\min\{t, 1-t\} \ge \frac{1}{4}(p-q)$ and $p(1-q) \ge t(1-t)$. It then follows from condition (7) that

$$t(1-t) \ge \frac{1}{8}(p-q) \ge c'\sqrt{t(1-t)}\max\left\{\frac{\sqrt{n}}{K}, \sqrt{\frac{\log^4 n}{K}}\right\},$$

which implies the inequalities in part (1) of the lemma. Part (2) follows directly from part (1) and the definition of ϵ. □

Due to the randomness of $\mathcal{A}$, W_1, W_2 and W_3 are symmetric random matrices with independent zero-mean entries. The support and variance of their entries are bounded in the following lemma.

Lemma 3: The following holds under the GSBM and the conditions (6) and (7) in Theorem 1.

1) *For $i = 1, 2, 3$, the absolute values of the entries of W_i are bounded by B_i a.s. and their variance is bounded by σ_i^2, where*

$$\begin{aligned}B_1 &:= \frac{1}{pK}, \quad \sigma_1^2 := \frac{1}{pK^2}, \\ B_2 &:= \frac{2}{p}\lambda c_{\mathcal{A}^c}, \quad \sigma_2^2 := \frac{4(1-t)}{p}\lambda^2 c_{\mathcal{A}^c}^2, \\ B_3 &:= \frac{2}{1-q}\lambda c_{\mathcal{A}}, \quad \sigma_3^2 := \frac{4t}{1-q}\lambda^2 c_{\mathcal{A}}^2.\end{aligned}$$

2) *We have $\sigma_i \ge c\frac{B_i \log^2 n}{\sqrt{K}}$ for $i = 1, 2, 3$.*

Proof: The first part of the lemma follows from the definitions of the W_i's, $q \le t \le p$ and $\epsilon < \frac{1}{2}$ (Lemma 2). The second part follows from Lemma 2. □

We now proceed with the proof of Proposition 2, The proof has three sub-steps, corresponding to checking the three conditions in Proposition 1.

(1) Bounding $\|W\|$.

Recall that W_1 is a random matrix with i.i.d. entries, and their absolute values and variance are bounded in Lemma 3. We apply standard bounds on the spectral norm of random matrices (Lemma 4 in the Appendix) to obtain w.h.p.

$$\|W_1\| \le 6\frac{1}{K}\sqrt{\frac{1}{p}}\sqrt{n} \le \frac{1}{4},$$

where the last inequality follows from $p \ge c\frac{n}{K^2}$ (cf. Lemma 2). In a similar manner, we obtain that w.h.p.

$$\begin{aligned}\|W_2\| &\le 6\cdot 2\sqrt{\frac{1-t}{p}}\lambda c_{\mathcal{A}^c}\cdot\sqrt{n} \\ &= 12\sqrt{\frac{(1-t)}{p}}\cdot\frac{1}{48}\sqrt{\frac{t}{(1-t)n}}\cdot\sqrt{n} \le \frac{1}{4},\end{aligned}$$

where the last inequality follows from $p \ge t$, and w.h.p.

$$\begin{aligned}\|W_3\| &\le 6\cdot 2\sqrt{\frac{t}{1-q}}\lambda c_{\mathcal{A}}\cdot\sqrt{n} \\ &= 12\sqrt{\frac{t}{1-q}}\cdot\frac{1}{48}\sqrt{\frac{1-t}{tn}}\cdot\sqrt{n} \le \frac{1}{4},\end{aligned}$$

where the last inequality follows from $1-t \le 1-q$. Finally, since $W_4 = (1+\epsilon)\lambda c_{\mathcal{A}} P_{R^c}(I)$ is a diagonal matrix, we have

$$\|W_4\| \le (1+\epsilon)\lambda c_{\mathcal{A}} \le 2\cdot\frac{1}{48}\sqrt{\frac{1-t}{tn}} \le \frac{1}{4}$$

since $t \ge c\frac{1}{n}$ (cf. Lemma 2). We conclude that $\|W\| \le \sum_{i=1}^4 \|W_i\| \le 1$.

(2) Bounding $\|P_T W\|_\infty$.

Define the sets $R_m := \{(i, j) : i, j \text{ in cluster } m\}$, and recall that r is the number of clusters and $R := \text{support}(Y^*) = \bigcup_{m=1}^r R_m$. We have $Y^* = \sum_{m=1}^r P_{R_m}(\mathbf{1}\mathbf{1}^\top)$, and thus its singular vectors satisfies

$$U_0 U_0^\top = \sum_{m=1}^r \frac{1}{k_m} P_{R_m}(\mathbf{1}\mathbf{1}^\top).$$

Therefore, for $i = 1, 2, 3$, each entry of the matrix $U_0 U_0^\top W_i$ equals $\frac{1}{k_m}$ times the sum of k_m independent zero-mean random variables (which are the entries of W_i), whose absolute values and variance are bounded in Lemma 3. Therefore, $\|U_0 U_0^\top W_i\|_\infty$ can be bounded by applying the standard Bernstein inequality (given as Lemma 5 in the Appendix) to each entry of $U_0 U_0^\top W_i$ and then using the union bound over all the entries. More specifically, by choosing the constant c_0 in Lemma 5 sufficiently large such that c_1 in the same lemma is at least 14, we have the following:

- The matrix W_1 satisfies

$$\begin{aligned}\|U_0 U_0^\top W_1\|_\infty &\le \frac{1}{K}\cdot c_0\sqrt{\frac{1}{pK^2}}\sqrt{K\log n} \\ &= c_0\frac{1}{K}\sqrt{\frac{\log n}{pK}} \le \frac{\log^2 n}{24^2}\sqrt{\frac{1}{Kn}} \quad \text{w.h.p.},\end{aligned}$$

where we use $p \geq c\frac{n}{K^2}$ in the last inequality (cf. Lemma 2) with c sufficiently large.

- Similarly, the matrix W_2 satisfies

$$\begin{aligned}\left\|U_0U_0^\top W_2\right\|_\infty &\leq \frac{1}{K}\cdot c_0\sqrt{\frac{1-t}{p}}\lambda c_{\mathcal{A}^c}\sqrt{K\log n}\\ &= c_0\sqrt{\frac{(1-t)\log n}{pK}}\frac{1}{48}\sqrt{\frac{t}{(1-t)n}}\\ &\leq \frac{\log^2 n}{24^2}\sqrt{\frac{1}{Kn}} \quad \text{w.h.p.},\end{aligned}$$

where we use $p \geq t$ and $\log n$ being sufficiently large in the last inequality.

- The matrix W_3 satisfies

$$\begin{aligned}\left\|U_0U_0^\top W_3\right\|_\infty &\leq \frac{1}{K}\cdot c_0\sqrt{\frac{t}{1-q}}\lambda c_{\mathcal{A}}\sqrt{K\log n}\\ &= c_0\sqrt{\frac{t\log n}{(1-q)K}}\frac{1}{48}\sqrt{\frac{1-t}{tn}}\\ &\leq \frac{\log^2 n}{24^2}\sqrt{\frac{1}{Kn}} \quad \text{w.h.p.},\end{aligned}$$

where we use $1-q \geq 1-t$ and $\log n$ being sufficiently in the last inequality.

- Finally, since W_4 is a diagonal matrix supported on R^c and $U_0U_0^\top$ is supported on R, we have $U_0U_0^\top W_4 = 0$.

On the other hand, we have

$$\begin{aligned}\lambda c_{\mathcal{A}}\epsilon &\geq \frac{1}{48}\sqrt{\frac{1-t}{tn}}\cdot 48\sqrt{\frac{\log^4 n}{Kt(1-t)}}\\ &= \frac{1}{t}\sqrt{\frac{\log^4 n}{Kn}} \geq \frac{1}{24}\sqrt{\frac{\log^4 n}{Kn}}\end{aligned}$$

and

$$\begin{aligned}\lambda c_{\mathcal{A}^c}\epsilon &\geq \frac{1}{48}\sqrt{\frac{t}{(1-t)n}}\cdot 48\sqrt{\frac{\log^4 n}{Kt(1-t)}}\\ &= \frac{1}{(1-t)}\sqrt{\frac{\log^4 n}{Kn}} \geq \frac{1}{24}\sqrt{\frac{\log^4 n}{Kn}},\end{aligned}$$

which implies

$$\frac{1}{24}\epsilon\lambda\min\{c_A, c_{A^c}\} \geq \frac{\log^2 n}{24^2}\sqrt{\frac{1}{Kn}}.$$

Combining with the previous bounds on $\left\|U_0U_0^\top W_i\right\|_\infty$ for $i = 1, 2, 3, 4$, we obtain $\left\|(U_0U_0^\top W_i)\right\|_\infty \leq \frac{1}{24}\epsilon\lambda\min\{c_A, c_{A^c}\}$.

Now observe that since W and $U_0U_0^\top$ are both symmetric, we have $WU_0U_0^\top = \left(U_0U_0^\top W\right)^\top$. Furthermore, we have

$$\begin{aligned}\left\|U_0U_0^\top WU_0U_0^\top\right\|_\infty &\leq \left\|U_0U_0^\top W\right\|_\infty \max_j \sum_i \left|\left(U_0U_0^\top\right)_{ij}\right|\\ &\leq \left\|U_0U_0^\top W\right\|_\infty.\end{aligned}$$

It follows that

$$\begin{aligned}\|P_T W\|_\infty &= \left\|U_0U_0^\top W + WU_0U_0^\top - U_0U_0^\top WU_0U_0^\top\right\|_\infty\\ &\leq \left\|U_0U_0^\top W\right\|_\infty + \left\|WU_0U_0^\top\right\|_\infty + \left\|U_0U_0^\top WU_0U_0^\top\right\|_\infty\\ &\leq 3\left\|U_0U_0^\top W\right\|_\infty \leq 3\sum_{i=1}^4 \left\|U_0U_0^\top W_i\right\|_\infty.\end{aligned}$$

Using the bounds on $\left\|U_0U_0^\top W_i\right\|_\infty$ derived above, we obtain that $\|P_T W\|_\infty \leq 12\cdot\frac{1}{24}\epsilon\lambda\min\{c_{\mathcal{A}}, c_{\mathcal{A}^c}\} = \frac{1}{2}\epsilon\lambda\min\{c_{\mathcal{A}}, c_{\mathcal{A}^c}\}$.

(3) The two equalities in condition (c) in Proposition 1 hold by the definition of W. The two inequalities in condition (c) follow from simple algebra as follows. Because $1-q \geq 1-t$ and $p \leq 4t$, we have $\frac{1-q}{p} \geq \frac{1}{4}\frac{1-t}{t}$. It follows from the conditions in Theorem 1 that

$$\begin{aligned}\frac{p-q}{4} &\geq c\sqrt{p(1-q)}\max\left\{\frac{\sqrt{n}}{K}, \sqrt{\frac{\log^4 n}{K}}\right\}\\ &\geq 8p(1-t)\cdot\frac{48}{\sqrt{t(1-t)}}\max\left\{\frac{\sqrt{n}}{K}, \sqrt{\frac{\log^4 n}{K}}\right\}\\ &= 8p(1-t)\epsilon. \qquad (17)\end{aligned}$$

We thus have

$$p - t \geq p - \left(\frac{3}{4}p + \frac{1}{4}q\right) = \frac{p-q}{4} \geq 8p(1-t)\epsilon.$$

One verifies that this implies $(1+\epsilon)\sqrt{\frac{t}{1-t}\frac{1-p}{p}} \leq (1-2\epsilon)\sqrt{\frac{1-t}{t}}$. Plugging in the values of $c_{\mathcal{A}}$ and $c_{\mathcal{A}^c}$ in (5) yields

$$(1+\epsilon)\frac{c_{\mathcal{A}^c}(1-p)}{p} \leq (1-2\epsilon)c_{\mathcal{A}},$$

Hence, for each $(i, j) \in R \cap \mathcal{A}$, we have

$$\begin{aligned}(U_0U_0^\top + W)_{ij} &= \frac{1}{p}(U_0U_0^\top)_{ij} + (1+\epsilon)\lambda c_{\mathcal{A}^c}\frac{1-p}{p}\\ &\leq \frac{1}{p}(U_0U_0^\top)_{ij} + (1-2\epsilon)c_{\mathcal{A}}. \qquad (18)\end{aligned}$$

We also have

$$\frac{1}{p}(U_0U_0^\top)_{ij} \leq \frac{1}{pK} \overset{(i)}{\leq} \frac{48}{K}\sqrt{\frac{n}{t(1-t)}}\cdot\frac{1}{48}\sqrt{\frac{1-t}{tn}} \leq \epsilon\cdot\lambda c_{\mathcal{A}}, \qquad (19)$$

where (i) follows from $p \geq t$. Combining (18) and (19) proves the first inequality in the condition (c).

Similarly, we have

$$t - q \geq \left(\frac{p}{4} + \frac{3q}{4}\right) - q = \frac{p-q}{4} \overset{(ii)}{\geq} 8p(1-t)\epsilon \overset{(iii)}{\geq} 2t(1-q)\epsilon,$$

where (ii) follows from (17) and (iii) follows from $p \geq t$ and $1-t \geq 1-\frac{3}{4}p-\frac{1}{4}q \geq \frac{1}{4}(1-q)$. This implies $(1+\epsilon)\sqrt{\frac{1-t}{t}\frac{q}{1-q}} \leq (1-\epsilon)\sqrt{\frac{t}{1-t}}$. Therefore, for each $(i, j) \in R^c \cap \mathcal{A}^c$, we have

$$w_{ij} = -(1+\epsilon)\frac{c_{\mathcal{A}}q}{1-q} \geq -(1-\epsilon)c_{\mathcal{A}^c},$$

proving the second inequality in condition (c). This completes the proof of Proposition 2.

VIII. Proof of Theorem 2

We use a standard information theoretic argument via Fano's inequality. For simplicity we assume n_1/K and n_2/K are both integers, and we use $c_1, c_2 \ldots$ to denote positive absolute constants. Let $\mathcal{F}$ be the set of all possible ways of assigning n nodes into n_1/K clusters of equal size K. When $K = \Theta(n_1) = \Theta(n_2)$, the cardinality of $\mathcal{F}$ can be bounded as

$$M := |\mathcal{F}| = \frac{1}{(n_1/K)!}\binom{n}{K}\binom{n-K}{K}\cdots\binom{n_1+K}{K} \geq c_2 \cdot c_1^{\frac{1}{2}n}$$

for some $c_1 > 1$ and $c_2 > 0$.

Suppose the true cluster matrix Y^* is obtained uniformly at random from $\mathcal{F}$, and the graph A is generated from Y^* according to GSBM with uniform edge probabilities. We use $\mathbb{P}_{A|Y^*}$ to denote the distribution of A given Y^*. Let $\hat{Y}$ be any measurable function of A. The Fano's inequality [44] gives

$$\sup_{Y^*\in\mathcal{F}} \mathbb{P}\left[\hat{Y} \neq Y^*|Y^*\right] \geq 1 - \frac{I(A;Y^*) + \log 2}{\log M} \geq 1 - \frac{I(A;Y^*) + \log 2}{c_3 n}$$

for n is sufficiently large, where $I(A;Y^*)$ is the mutual information between A and Y^*. We now bound $I(A;Y^*)$. Let $H(\cdot)$ denote the Shannon entropy and $H(\cdot|Y^*)$ the Shannon entropy conditioned on Y^*. Observe that

$$I(A;Y^*) = H(A) - H(A|Y^*) \leq \sum_{(i,j):i>j} H(a_{ij}) - H(A|Y^*) = \binom{n}{2}H(a_{12}) - \binom{n}{2}H(a_{12}|Y^*) = \binom{n}{2}I(a_{12};Y^*),$$

where in the second equality we have used the symmetry under the uniform distribution of Y^* and the conditional independence between $a'_{ij}s$. By definition of the mutual information, we have

$$I(a_{12};Y^*) = I(a_{12};y^*_{12}) = \mathbb{E}_{y^*_{12}}\left[D\left(\mathbb{P}(a_{12}|y^*_{12})\|\mathbb{P}(a_{12})\right)\right].$$

We can directly compute the divergence on the last RHS. Let $\alpha := \mathbb{P}(y^*_{12} = 1) = \frac{(K-1)n_1}{n^2}$ and $\gamma := \mathbb{P}(a_{11} = 1) = \alpha p + (1-\alpha)q$. It follows that

$$\begin{aligned}
&\mathbb{E}_{y^*_{12}}\left[D\left(\mathbb{P}(a_{12}|y_{12})\|\mathbb{P}(a_{12})\right)\right] \\
&= \alpha p \log\frac{p}{\gamma} + \alpha(1-p)\log\frac{(1-p)}{(1-\gamma)} + (1-\alpha)q\log\frac{q}{\gamma} \\
&\quad + (1-\alpha)(1-q)\log\frac{(1-q)}{(1-\gamma)} \\
&\leq \alpha p\left(\frac{p}{\gamma}-1\right) + \alpha(1-p)\left(\frac{1-p}{1-\gamma}-1\right) + (1-\alpha)q\left(\frac{q}{\gamma}-1\right) \\
&\quad + (1-\alpha)(1-q)\left(\frac{1-q}{1-\gamma}-1\right) \\
&= \frac{\alpha(1-\alpha)(p-q)^2}{\gamma(1-\gamma)} \leq c_4\frac{(p-q)^2}{p(1-q)},
\end{aligned}$$

where in the last inequality we use $\gamma \geq \alpha p$, $1-\gamma \geq (1-\alpha)(1-q)$ and $\alpha, 1-\alpha = \Theta(1)$. Combining pieces, we obtain

$$\sup_{Y\in\mathcal{F}} \mathbb{P}\left[\hat{Y} \neq Y|Y\right] \geq 1 - \frac{c_5\frac{(p-q)^2n^2}{p(1-q)} + \log 2}{c_3 n}.$$

For the last R.H.S. to be less than $\frac{1}{4}$, we need $\frac{(p-q)^2}{p(1-q)} \geq c_6\frac{1}{n}$. This completes the proof of the theorem.

IX. Proof of Theorem 3

Suppose the eigenvalues of the matrix $\mathbb{E}[A]$ are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, whose values are computed in Section IV-D. Observe that the matrix $A - \mathbb{E}A$ is a random symmetric matrix with independent zero-mean entries, each of which is bounded in absolute value by 1 and has variance bounded by $\max\{p(1-p), q(1-q)\} \leq p(1-q)$. Under the condition of Theorem 3, we may apply Lemma 4 to obtain $\|A - \mathbb{E}A\| \leq 4\sqrt{p(1-q)n}$ w.h.p. It then follows from Weyl's inequality [45] that w.h.p.

$$\max_i\left\{\left|\hat{\lambda}_i - \lambda_i\right|\right\} \leq \|A - \mathbb{E}A\| \leq 4\sqrt{p(1-q)n}. \quad (20)$$

In the sequel, we assume we are on the event that (20) holds.

a) Estimation of r : Recall that $\lambda_1 = K(p-q) + nq + (1-p)$, $\lambda_2, \ldots, \lambda_r = K(p-q) + (1-p)$, and $\lambda_{r+1}, \ldots, \lambda_n = 1-p$. The inequality (20) implies that for some universal constant c_1:

- $\hat{\lambda}_1 - \hat{\lambda}_2 \leq \lambda_1 - \lambda_2 + \left|\hat{\lambda}_1 - \lambda_1\right| + \left|\hat{\lambda}_2 - \lambda_2\right| \leq nq + c_1\sqrt{p(1-q)n}$;
- similarly, $\hat{\lambda}_i - \hat{\lambda}_{i+1} \leq c_1\sqrt{p(1-q)n}$ for $i = 2, \ldots r-1$ and $i \geq r+1$;
- $\hat{\lambda}_r - \hat{\lambda}_{r+1} \geq \lambda_r - \lambda_{r+1} - \left|\hat{\lambda}_r - \lambda_r\right| - \left|\hat{\lambda}_{r+1} - \lambda_{r+1}\right| \geq K(p-q) - c_1\sqrt{p(1-q)n}$.

Under the condition (7), we have $K(p-q) \geq c_2\sqrt{p(1-q)n}$ for some constant c_2. This implies $\hat{\lambda}_r - \hat{\lambda}_{r+1} > \frac{K(p-q)}{2} > \hat{\lambda}_i - \hat{\lambda}_{i+1}$ for all $i > 1$ and $i \neq r$. This guarantees $\hat{r} = r$ and thus $\hat{K} = K$.

b) Estimation of p and q: By (20) and the triangle inequality, the estimation error of $\hat{q}$ satisfies

$$|\hat{q} - q| = \left|\frac{\hat{\lambda}_1 - \lambda_1}{n} - \frac{\hat{\lambda}_2 - \lambda_2}{n}\right| \leq c_3\frac{\sqrt{p(1-q)n}}{K}.$$

Similarly, we have

$$\begin{aligned}
|\hat{p} - p| &= \left|\frac{\hat{K}\hat{\lambda}_1 + (n-\hat{K})\hat{\lambda}_2 - n}{n(\hat{K}-1)} - \frac{K\lambda_1 + (n-K)\lambda_2 - n}{n(K-1)}\right| \\
&= \left|\frac{K(\hat{\lambda}_1 - \lambda_1) + (n-K)(\hat{\lambda}_2 - \lambda_2)}{n(K-1)}\right| \\
&\leq c_3\frac{\sqrt{p(1-q)n}}{K}.
\end{aligned}$$

c) Choosing t: Using the above bounds on $\hat{p}$ and $\hat{q}$, we obtain

$$\begin{aligned}
t &= \frac{p+q}{2} + \frac{\hat{p} - p + \hat{q} - q}{2} \leq \frac{p+q}{2} + c_4\frac{\sqrt{p(1-q)n}}{K} \\
&\leq \frac{p+q}{2} + \frac{p-q}{4} = \frac{3}{4}p + \frac{1}{4}q,
\end{aligned}$$

where in the last inequality we use $\frac{p-q}{4} \geq c_4\frac{\sqrt{p(1-q)}}{K}$, satisfied under the condition (7). This proves one side of the interval for t. The other side is proved in a similar way.

X. Conclusion

This work is motivated by clustering large-scale networks such as modern online social networks, where the graphs are often highly noisy and have heterogeneous and non-random structures. We considered a natural and versatile model, namely the semi-random Generalized Stochastic Blockmodel, for clustered random graphs. This model recovers many classical generative models for graph clustering. We presented a convex optimization formulation, essentially a convexification of the maximum likelihood estimator. Our theoretic analysis shows that this method is guaranteed to recover the correct clusters under a wide range of parameters of the problem. In fact, our method order-wise outperforms existing methods in this setting, in the sense that it succeeds under less restrictive conditions. Experiment results also validate the effectiveness of the proposed method.

Possible directions for future work include faster algorithm implementations, developing effective post-processing/rounding schemes when the obtained solution is not an exact cluster matrix, and extension to online clustering settings (e.g., via incremental stochastic optimization [46]). It is also interesting to extend the algorithms and analysis to more general settings beyond the models in Definitions 1 and 2, for example, when the in-cluster and cross-cluster densities are not bounded uniformly and the clusters have overlaps.

Appendix

In this section, we record two technical lemmas that are needed in the proofs of our theoretical results. The first lemma is a standard bound on the spectral norm of a random symmetric matrix.

Lemma 4: Suppose Y is a symmetric $n \times n$ matrix, where Y_{ij}, $1 \leq i, j \leq n$ are independent random variables, each of which has mean 0 and variance at most σ^2 and is bounded in absolute value by B. If $\sigma \geq c_1 \frac{B \log^2 n}{\sqrt{n}}$ for some absolute constant $c_1 > 0$, then with probability at least $1 - n^{-10}$,

$$\|Y\| \leq 4\sigma\sqrt{n}.$$

Proof: Except for Y being symmetric, the proof is the same as that of [47, Th 3.1]. □

The second lemma is a restatement of the standard Bernstein inequality for the sum of independent random variables.

Lemma 5: Let $Y_1, \ldots, Y_N$ be independent random variables, each of which is bounded in absolute value by B a.s. and has variance bounded by σ^2. For any constant $c_1 > 0$, there exists a constant $c_0 > 0$ independent of σ, B, N and n such that for any $n \geq 1$, if $\sigma \geq B\sqrt{\frac{\log n}{N}}$, then we have

$$\left| \sum_{i=1}^{N} Y_i - \mathbb{E}\left[\sum_{i=1}^{N} Y_i \right] \right| \leq c_0 \sigma \sqrt{N \log n}$$

with probability at least $1 - 2n^{-c_1}$.

References

[1] S. Fortunato, "Community detection in graphs," *Phys. Rep.*, vol. 486, nos. 3–5, pp. 75–174, 2010.

[2] J. Leskovec, K. J. Lang, A. Dasgupta, and M. W. Mahoney, "Statistical properties of community structure in large social and information networks," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 695–704.

[3] K. Chaudhuri, F. Chung, and A. Tsiatas, "Spectral clustering of graphs with general degrees in the extended planted partition model," in *Proc. 25th Annu. Conf. Learn. Theory (COLT)*, Edinburgh, Scotland, Jun. 2012.

[4] A. Coja-Oghlan, "Coloring semirandom graphs optimally," in *Proc. 31st Int. Colloq. Automat., Lang. Program.*, 2004, pp. 71–100.

[5] A. Condon and R. M. Karp, "Algorithms for graph partitioning on the planted partition model," *Random Struct. Algorithms*, vol. 18, no. 2, pp. 116–140, Mar. 2001.

[6] P. W. Holland, K. B. Laskey, and S. Leinhardt, "Stochastic blockmodels: First steps," *Social Netw.*, vol. 5, no. 2, pp. 109–137, Jun. 1983.

[7] B. P. W. Ames and S. A. Vavasis, "Convex optimization for the planted k-disjoint-clique problem," *Math. Program.*, vol. 143, nos. 1–2, pp. 299–337, 2014.

[8] N. Alon, M. Krivelevich, and B. Sudakov, "Finding a large hidden clique in a random graph," *Random Struct. Algorithms*, vol. 13, nos. 3–4, pp. 457–466, 1998.

[9] F. McSherry, "Spectral partitioning of random graphs," in *Proc. 42nd IEEE Symp. Found. Comput. Sci.*, Oct. 2001, pp. 529–537.

[10] N. Alon and N. Kahale, "A spectral technique for coloring random 3-colorable graphs," *SIAM J. Comput.*, vol. 26, no. 6, pp. 1733–1748, 1997.

[11] B. Bollobás and A. D. Scott, "Max cut for random graphs with a planted partition," *J. Combinat., Probab. Comput.*, vol. 13, nos. 4–5, pp. 451–474, 2004.

[12] U. Feige and J. Kilian, "Heuristics for semirandom graph problems," *J. Comput. Syst. Sci.*, vol. 63, no. 4, pp. 639–671, 2001.

[13] M. Krivelevich and D. Vilenchik, "Semirandom models as benchmarks for coloring algorithms," in *Proc. 3rd Workshop Anal. Algorithmics Combinat. (ANALCO)*, 2006, pp. 211–221.

[14] R. B. Boppana, "Eigenvalues and graph bisection: An average-case analysis," in *Proc. 28th Annu. Symp. Found. Comput. Sci.*, Oct. 1987, pp. 280–285.

[15] M. Jerrum and G. B. Sorkin, "The metropolis algorithm for graph bisection," *Discrete Appl. Math.*, vol. 82, nos. 1–3, pp. 155–175, Mar. 1998.

[16] T. Carson and R. Impagliazzo, "Hill-climbing finds random planted bisections," in *Proc. 12th Annu. ACM-SIAM Symp. Discrete Algorithms*, 2001, pp. 903–909.

[17] J. Giesen and D. Mitsche, "Reconstructing many partitions using spectral techniques," in *Fundamentals of Computation Theory*. New York, NY, USA: Springer-Verlag, 2005, pp. 433–444.

[18] R. Shamir and D. Tsur, "Improved algorithms for the random cluster graph model," *Random Struct. Algorithms*, vol. 31, no. 4, pp. 418–449, Dec. 2007.

[19] A. Coja-Oghlan, "Graph partitioning via adaptive spectral techniques," *J. Combinat. Probab. Comput.*, vol. 19, no. 2, pp. 227–284, Mar. 2010.

[20] K. Rohe, S. Chatterjee, and B. Yu, "Spectral clustering and the high-dimensional stochastic blockmodel," *Ann. Statist.*, vol. 39, no. 4, pp. 1878–1915, 2011.

[21] S. Oymak and B. Hassibi, "Finding dense clusters via 'low rank + sparse' decomposition," to be published.

[22] B. P. W. Ames, "Guaranteed clustering and biclustering via semidefinite programming," in *Mathematical Programming*. Berlin, Germany: Springer-Verlag, Nov. 2013.

[23] Y. Chen, S. Sanghavi, and H. Xu, "Clustering sparse graphs," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, Inc., 2012, pp. 2204–2212.

[24] A. Anandkumar, R. Ge, D. Hsu, and S. M. Kakade, "A tensor approach to learning mixed membership community models," *J. Mach. Learn. Res.*, vol. 15, pp. 2239–2312, Jun. 2014.

[25] N. Ailon, Y. Chen, and H. Xu, "Breaking the small cluster barrier of graph clustering," in *Proc. 30th Int. Conf. Mach. Learn.*, 2013, pp. 995–1003.

[26] 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*, vol. 84, no. 6, p. 066106, Dec. 2011.

[27] R. R. Nadakuditi and M. E. J. Newman, "Graph spectra and the detectability of community structure in networks," *Phys. Rev. Lett.*, vol. 108, no. 18, p. 188701, 2012.
[28] V. Chandrasekaran, S. Sanghavi, P. A. Parrilo, and A. S. Willsky, "Rank-sparsity incoherence for matrix decomposition," *SIAM J. Optim.*, vol. 21, no. 2, pp. 572–596, 2011.
[29] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *J. ACM*, vol. 58, no. 3, p. 11, May 2011.
[30] Y. Chen, A. Jalali, S. Sanghavi, and C. Caramanis, "Low-rank matrix recovery from errors and erasures," *IEEE Trans. Inf. Theory*, vol. 59, no. 7, pp. 4324–4337, Jul. 2013.
[31] X. Li, "Compressed sensing and matrix completion with constant proportion of corruptions," *Constructive Approx.*, vol. 37, no. 1, pp. 73–99, 2013.
[32] Y. Chen, A. Jalali, S. Sanghavi, and H. Xu, "Clustering partially observed graphs via convex optimization," *J. Mach. Learn. Res.*, vol. 15, pp. 2213–2238, Jun. 2014.
[33] B. Recht, M. Fazel, and P. A. Parrilo, "Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization," *SIAM Rev.*, vol. 52, no. 3, pp. 471–501, 2010.
[34] C. Mathieu and W. Schudy, "Correlation clustering with noisy input," in *Proc. 21st Annu. ACM-SIAM Symp. Discrete Algorithms*, 2010, pp. 712–728.
[35] A. Frieze and C. McDiarmid, "Algorithmic theory of random graphs," *Random Struct. Algorithms*, vol. 10, nos. 1–2, pp. 5–42, 1997.
[36] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: Freeman, 1979.
[37] U. Feige and E. Ofek, "Spectral techniques applied to sparse random graphs," *Random Struct. Algorithms*, vol. 27, no. 2, pp. 251–275, Sep. 2005.
[38] E. Hazan and R. Krauthgamer, "How hard is it to approximate the best Nash equilibrium?" *SIAM J. Comput.*, vol. 40, no. 1, pp. 79–91, 2011.
[39] A. Juels and M. Peinado, "Hiding cliques for cryptographic security," *Designs, Codes Cryptograph.*, vol. 20, no. 3, pp. 269–280, 2000.
[40] O. Shamir and N. Tishby, "Spectral clustering on a budget," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, Apr. 2011, pp. 661–669.
[41] Z. Lin, M. Chen, L. Wu, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," Dept. Elect. Comput. Eng., UIUC, Champaign, IL, USA, Tech. Rep. UILU-ENG-09-2215, 2009.
[42] R. Sibson, "SLINK: An optimally efficient algorithm for the single-link cluster method," *Comput. J.*, vol. 16, no. 1, pp. 30–34, 1973.
[43] U. Von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, 2007.
[44] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, 2nd ed. New York, NY, USA: Wiley, 2006.
[45] R. Bhatia, *Perturbation Bounds for Matrix Eigenvalues*. Essex, U.K.: Longman, 1987.
[46] J. Feng, H. Xu, and S. Yan, "Online robust PCA via stochastic optimization," in *Advances in Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2013, pp. 404–412.
[47] D. Achlioptas and F. Mcsherry, "Fast computation of low-rank matrix approximations," *J. ACM*, vol. 54, no. 2, p. 9, 2007.

Yudong Chen is a postdoctoral scholar at the Department of Electrical Engineering and Computer Sciences at the University of California, Berkeley. He obtained his Ph.D. in electrical and computer engineering from the University of Texas at Austin in 2013. He received his BS and MS degrees from Tsinghua University, Beijing, China. His research interests include statistics, machine learning and applications in large-scale network problems.

Sujay Sanghavi (M'06) is an Associate Professor in Electrical Engineering at the University of Texas, Austin. He obtained two MS degrees and a Phd from the University of Illinois, Urbana-Champaign, and subsequently was a Postdoctoral Fellow at MIT for two years. Sujay is a recipient of the NSF CAREER award and a DTRA young investigator award. Sujay's research interests are in machine learning, optimization and networks.

Huan Xu received the B.Eng. degree in automation from Shanghai Jiaotong University, Shanghai, China in 1997, the M.Eng. degree in electrical engineering from the National University of Singapore in 2003, and the Ph.D. degree in electrical engineering from McGill University, Canada in 2009. From 2009 to 2010, he was a postdoctoral associate at The University of Texas at Austin. Since 2011, he has been an assistant professor at the Department of Mechanical Engineering at the National University of Singapore. His research interests include statistics, machine learning, robust optimization, and planning and control. He is an associate editor of IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and is on the editorial board of *Computational Management Science*.