Clustering from General Pairwise Observations with Applications to Time-varying Graphs[*]

Shiau Hong Lim SHONGLIM@SG.IBM.COM
IBM Research
10 Marina Boulevard
Singapore 018983

Yudong Chen YUDONG.CHEN@CORNELL.EDU
School of Operations Research and Information Engineering
Cornell University
Ithaca, NY 14853, USA

Huan Xu HUAN.XU@ISYE.GATECH.EDU
School of Industrial and Systems Engineering
Georgia Institute of Technology
755 Ferst Drive, NW, Atlanta, GA 30332, USA

Editor: Edo Airoldi

Abstract

We present a general framework for graph clustering and bi-clustering where we are given a general observation (called a label) between each pair of nodes. This framework allows a rich encoding of various types of pairwise interactions between nodes. We propose a new tractable and robust approach to this problem based on convex optimization and maximum likelihood estimators. We analyze our algorithms under a general statistical model extending the planted partition and stochastic block models. Both sufficient and necessary conditions are provided for successful recovery of the underlying clusters. Our theoretical results subsume many existing graph clustering results for a wide range of settings, including planted partition, weighted clustering, submatrix localization and partially observed graphs. Furthermore, our results are applicable to novel settings including time-varying graphs, providing new insights to solutions of these problems. We provide empirical results on both synthetic and real data that corroborate with our theoretical findings.

Keywords: graph clustering, convex optimization, low-rank matrix, information divergence, time-varying graphs, pairwise observation, dynamic graphs

1. Introduction

In the standard formulation of graph clustering, we are given an unweighted graph and seek a partitioning of the nodes into disjoint groups such that members of the same group are more densely connected than those in different groups. Here, the presence of an edge represents certain affinity or similarity between the nodes, and the absence of an edge represents the lack thereof.

In many applications, from chemical interactions to social networks, the interactions between nodes are much richer than a simple “edge” or “non-edge”. Such extra information can be used to improve the clustering quality. We may represent each type of pairwise interaction by a *label*. One

[*] Partial preliminary results are presented in the conference papers Chen et al. (2014b) and Lim et al. (2014).

simple setting of this type is weighted graphs, where instead of a 0-1 graph, we have edge weights representing the strength of the pairwise interaction. In this case the observed label between each pair is a real number. In a more general setting, the label need not be a number. For example, on social networks like Facebook, the label between two persons may be "they are friends", "they went to different schools", "they liked 21 common pages", or the concatenation of them. In such cases different labels carry different information about the underlying community structure. A standard approach that converts these pairwise interactions into a simple edge/non-edge and then applies standard clustering algorithms, often does not work well here, as much of the information may be lost. Even in the simple case of a standard weighted/unweighted graph, it may not be immediately clear how information in the graph should be used in clustering. For example, should the absence of an edge be interpreted as a *neutral* observation carrying no information, or as a *negative* observation indicating dissimilarity between the two nodes?

We emphasize that our notion of labels (types of pairwise observations) is very general. A label can take real, discrete, categorical or even mixed values—we will see several such examples in Section 4. A label can even take the form of a time series, i.e., a record of time varying interactions such as "A and B messaged each other on June 1st, 4th, 15th and 21st", or "they used to be friends, but they stop talking to each other since 2012". The labeled graph model is therefore an immediate tool for analyzing time-varying graphs.

In this paper, we present a new and principled approach for graph clustering that is directly based on general pairwise labels. We assume that between each pair of nodes i and j, one observes a label L_{ij} that takes values in a label set $\mathcal{L}$. The set $\mathcal{L}$ may be continuous, discrete or mixed, and need not have any algebraic or geometric structure. The standard graph model corresponds to a binary label set $\mathcal{L} = \{\text{edge}, \text{non-edge}\}$, and a weighted graph corresponds to $\mathcal{L} = \mathbb{R}$. Given the matrix of observed labels $L = (L_{ij}) \in \mathcal{L}^{n\times n}$ between n nodes, the goal is to partition these nodes into disjoint clusters. Our algorithmic approach is based on finding a partition that optimizes a weighted objective that is appropriately constructed from the observed labels. This formulation leads to a combinatorial optimization problem, and our algorithms use its convex relaxation.

To systematically evaluate clustering performance, we consider a generalization of the stochastic block model and the planted partition model (Holland et al., 1983; Condon and Karp, 2001). Our model assumes that the observed labels are generated based on an underlying set of ground truth clusters, where node pairs from the same cluster generate labels using a distribution μ over $\mathcal{L}$, and pairs from different clusters use a different distribution ν. The standard planted partition model corresponds to the case where μ and ν are Bernoulli distributions with $\mu(\text{edge}) = p$ and $\nu(\text{edge}) = q$, $p \neq q$. We provide theoretical guarantees for our algorithm under this generalized model.

By specializing to concrete examples of the distributions μ and ν, our results cover a wide range of clustering settings—with theoretical guarantees matching or stronger than existing work—including the standard stochastic block model, partially observed graphs, weighted graphs and submatrix localization. Our framework in fact allows us to handle new classes of problems that are not a priori obvious to be a special case of our model, including the clustering of time-varying graphs.

Our framework easily generalizes to the bi-clustering setting, where pairwise labels are observed between two disjoint sets of nodes, potentially from different domains, and the task is to jointly cluster these two sets of nodes given the bipartite label observations. All our algorithmic and statistical results extend to this bi-clustering setting with only minor changes.

Remark 1 *Preliminary versions of some of the results here have appeared in part in Chen et al. (2014b) and Lim et al. (2014). The theory was previously stated with respect to only discrete label sets, but are now extended to general label sets, including continuous and mixed-valued labels, which appear frequently in applications (see, for example, Section 4). We also include a more detailed and systematic discussion on the bi-clustering setting and various special cases. Implementation details of efficient first-order solvers are now included. We also report a much more extensive set of empirical results on both synthetic and real data, including comparison with other methods.*

1.1 Related Work

The planted partition model/stochastic block model (Condon and Karp, 2001; Holland et al., 1983) are standard models for studying graph clustering. Variants of the models cover partially observed graphs (Oymak and Hassibi, 2011; Chen et al., 2014a), weighted graphs and the submatrix localization and bi-clustering problems (Balakrishnan et al., 2011; Kolar et al., 2011). All these models are special cases of ours. Various algorithms have been proposed and analyzed under these models, such as spectral clustering (McSherry, 2001; Chaudhuri et al., 2012; Rohe et al., 2011), convex optimization approaches (Mathieu and Schudy, 2010; Ames and Vavasis, 2011) and tensor decomposition methods (Anandkumar et al., 2014). Ours is based on convex optimization, generalizing the convexified maximum likelihood approach in Chen et al. (2014c).

Time-varying graphs arise in a broad range of applications, and the problem of clustering such graphs has been studied in various context (see Fortunato, 2010; Sun et al., 2007; Chakrabarti et al., 2006; Kawadia and Sreenivasan, 2012; Nguyen et al., 2011, and the references therein). The stochastic block model has also been extended in a number of ways to accommodate time-varying graphs. For example, Han et al. (2015) consider multiple, independent graphs, where edge distributions can differ in each graph and each cluster membership pair. In the work of Xu and Hero (2014); Matias and Miele (2016), the cluster membership of an individual node is allowed to change with time. Another extension is in allowing mixed cluster membership in time-varying networks, as done by Fu et al. (2009). These extensions allow additional flexibilities, but the existing solutions lack the kind of theoretical performance guarantees afforded by our approach. Also note that dealing with time-varying graph is only *one* of the applications of our general theory on clustering based on pairwise labels.

Most related to our setting is the *labelled stochastic block model* proposed by Heimlicher et al. (2012) and Lelarge et al. (2013). A main difference in their model is that they assume each pairwise observation is from a two-step process: first an edge/non-edge is observed; if it is an edge then a label is associated with it. In our model all observations are in the form of labels—in particular, an edge or no-edge is also a label—which covers their setting as a special case. Our model is therefore more general and natural, and as a result our theory covers a broad class of subproblems including time-varying graphs. Moreover, their analysis is mainly restricted to the two-cluster setting with edge probabilities on the order of $\Theta(1/n)$, while we allow for an arbitrary number of clusters and a wide range of edge/label distributions. In addition, we consider the setting where the distributions of the labels are not precisely known. Algorithmically, they use belief propagation (Heimlicher et al., 2012) and spectral methods (Lelarge et al., 2013).

Appearing after the conference versions of this paper, the work by Jog and Loh (2015) studies a form of labelled stochastic block model and thus is also related to ours. Restricting to the homogeneous case with equal-size clusters, they derive recovery conditions in terms of the Renyi divergence. Their results are based on the maximum likelihood decoder, which is not computationally feasible. Another recent work by Chen et al. (2015) also studies the statistical limits of information recovery from pairwise observations. Their model is however quite different from ours, and the results are again based on the intractable maximum likelihood decoder.

2. Problem Setup and Algorithms

We assume n nodes are partitioned into r disjoint clusters of size at least K. The clusters are unknown and considered as the ground truth. For each pair of nodes (i, j), a label $L_{ij} \in \mathcal{L}$ is observed, where $\mathcal{L}$ is the set of all possible values of the label. These labels are generated independently across node pairs according to some distributions μ and ν on $\mathcal{L}$.[1] In particular, if nodes i and j are in the same cluster, the observed label L_{ij} follows the distribution μ, otherwise L_{ij} follows ν. The goal is to recover the ground truth clusters given the pairwise labels, which is represented as a matrix $L = (L_{ij}) \in \mathcal{L}^{n\times n}$. We encode the true clusters as an $n \times n$ *cluster matrix* Y^*, where $Y^*_{ij} = 1$ if nodes i and j belong to the same cluster and $Y^*_{ij} = 0$ otherwise, with the convention that $Y^*_{ii} = 1$ for all i. The problem is therefore to find Y^* given L.

We take an optimization approach to this problem. To motivate our algorithm, first consider the case of clustering a weighted graph, where $\mathcal{L} = \mathbb{R}$ and all labels are real numbers. Suppose that positive weights indicate affinity between node pairs while negative weights indicate dissimilarity. A natural approach is to partition the nodes in a way that maximizes the total weight inside the clusters (this is equivalent to *correlation clustering* by Bansal et al. 2004). Mathematically, this formulation is to find a clustering, represented by a cluster matrix $Y \in \{0, 1\}^{n\times n}$, such that the sum $\sum_{i,j} L_{ij}Y_{ij}$ is maximized. In the setting of general labels, we pick a *weight function* $w : \mathcal{L} \to \mathbb{R}$, which assigns a number $W_{ij} = w(L_{ij})$ to the label L_{ij} observed at each pair (i, j). We then solve the following max-weight problem:

$$\begin{aligned} \max_{Y} \quad & \langle W, Y \rangle \\ \text{s.t.} \quad & Y \text{ is an } n \times n \text{ cluster matrix,} \end{aligned} \tag{1}$$

where $\langle W, Y \rangle := \sum_{i,j} W_{ij}Y_{ij}$ is the trace inner product. Note that once the weight function is specified, this formulation effectively converts the problem of clustering from labels into a weighted clustering problem.

The optimization program (1) is non-convex due to the combinatorial constraint. Our algorithm is based on a convex relaxation of (1), using the fact that any cluster matrix is a symmetric positive semidefinite, block-diagonal 0-1 matrix. Relaxing the constraint in (1) leads to the following convex optimization problem:

$$\begin{aligned} \max_{Y} \quad & \langle W, Y \rangle \\ \text{s.t.} \quad & Y \in \mathcal{S}^n_+ , \\ & 0 \le Y_{ij} \le 1, \forall (i, j), \end{aligned} \tag{2}$$

where $\mathcal{S}^n_+$ denotes the set of $n \times n$ symmetric positive semidefinite matrices. We say that the program (2) recovers the true clusters if it has a unique optimal solution equal to the true cluster matrix Y^*.

[1] More precisely, we assume that there is a σ-algebra $\mathcal{F}$ such that $(\mathcal{L}, \mathcal{F})$ is a measurable space, endowed with probability measures μ and ν.

One has the freedom of choosing the weight function w. Here, the likelihood ratio between label distributions μ and ν will play an important role. We assume that the measures μ and ν are absolutely continuous with each other, as well as with some base measure λ on $\mathcal{L}$. With a slight abuse of notation, the likelihoods of a label $l \in \mathcal{L}$ with respect to μ and ν are given by

$$\mu(l) = \frac{\mathrm{d}\mu}{\mathrm{d}\lambda}(l) \quad \text{and} \quad \nu(l) = \frac{\mathrm{d}\nu}{\mathrm{d}\lambda}(l),$$

where $\frac{\mathrm{d}\mu}{\mathrm{d}\lambda}$ and $\frac{\mathrm{d}\nu}{\mathrm{d}\lambda}$ are the respective Radon-Nikodym derivatives. In the sequel, unless specified otherwise, we will always assume that λ is the counting measure if $\mathcal{L}$ is a discrete label set, in which case we simply have $\mu(l) = \Pr(L_{ij} = l)$. For a continuous set $\mathcal{L}$, we use the Lebesgue base measure and hence $\mu(l)$ is the usual density function of μ. Later we will encounter more general, mixed-valued label sets $\mathcal{L}$.

Intuitively, a weight function w should assign $w(L_{ij}) > 0$ to a label L_{ij} with $\mu(L_{ij}) > \nu(L_{ij})$, so the program (2) is encouraged to place nodes i and j in the same cluster, the more likely possibility; similarly we should have $w(L_{ij}) < 0$ if $\mu(L_{ij}) < \nu(L_{ij})$. In other words, a good weight function should reflect the information in μ and ν. Our theoretical results in Section 3 characterize the performance of the program (2) for *any* given weight function w. Building on this general result, we further derive robust and optimal choices of w. Since μ and ν are often unknown, Section 3 also includes results when w is chosen based on distributions that are different from the true distributions.

For cluster recovery to be possible, the observed labels must contain sufficient information to distinguish different clusters. If $\mu = \nu$, the observed labels for intra- and inter-cluster pairs will have the same distribution, in which case the data L is distributed independently of the underlying clusters and recovery is impossible. In general, the problem becomes harder if μ and ν are more similar to each other. We quantify this relation precisely in Section 3.

2.1 Alternative Formulations and the Bi-clustering Problem

Program (2) uses the fact that Y^* is symmetric positive semidefinite. The following program, which is based on a more relaxed constraint using the nuclear norm $\|\cdot\|_*$,[2] also works and has essentially the same theoretical guarantees:

$$\begin{aligned} \max_{Y} \quad & \langle W, Y \rangle, \\ \text{s.t.} \quad & \|Y\|_* \le \|Y^*\|_* \\ & 0 \le Y_{ij} \le 1, \forall (i,j). \end{aligned} \tag{3}$$

Intuitively, this relaxed problem uses the fact that a cluster matrix always satisfies $\|Y^*\|_* = n$, which is much smaller than a general $n \times n$ binary matrix.

The formulation (3) has the advantage that it applies directly to a bi-clustering setting. In this setting, there are two disjoint sets of nodes $\mathcal{N}_1$ and $\mathcal{N}_2$, where $|\mathcal{N}_1| = n_1$ and $|\mathcal{N}_2| = n_2$. Each of the sets $\mathcal{N}_1$ and $\mathcal{N}_2$ are partitioned into r clusters $\{C_k, k \in [r]\}$ and $\{C'_k, k \in [r]\}$, respectively, where the clusters C_k and C'_k are associated with each other and called a bi-cluster. Similarly to the clustering case, the true cluster matrix $Y^* \in \{0,1\}^{n_1 \times n_2}$ is defined as $Y^*_{ij} = 1$ if and only if

[2] The nuclear norm of a matrix is defined as the sum of its singular values. A cluster matrix is positive semidefinite, so its nuclear norm is equal to its trace and also called the trace norm.

$(i,j) \in C_k \times C'_k$ for some $k \in [r]$. Labels are generated from the distribution μ for each pair (i,j) with $Y^*_{ij} = 1$, and from ν otherwise. Here Y^* is not necessarily a square or positive semidefinite matrix. The program (3) still applies, with the understanding that the matrices W and Y have size $n_1 \times n_2$ instead of $n \times n$.

While we always have $\|Y^*\|_* = n$ in the clustering case, this is not the case in general for bi-clustering. The value of $\|Y^*\|$ used in the constraint of (3) is usually unknown in practice, in which case we can instead solve an equivalent formulation in terms of a Lagrange multiplier:

$$\begin{aligned} \max_{Y} \quad & \langle W, Y\rangle - \eta \|Y\|_* \\ \text{s.t.} \quad & 0 \le Y_{ij} \le 1, \forall (i,j). \end{aligned} \tag{4}$$

The formulations (2)–(4) are semidefinite programs that can be solved in polynomial time by various methods. In Section 5 we describe efficient first order solvers and discuss other implementation details. Note that by convex duality, if the constrained formulation (3) succeeds in recovering the true cluster matrix Y^*, then there exists a multiplier η for which the Lagrangian formulation (4) also succeeds.[3] Therefore, all our theoretical results for the formulations (2) and (3) in Section 3 also hold for the formulation (4) with a suitable η.

3. Theoretical Results

In this section, we provide theoretical analysis for the performance of the convex formulations (2) and (3) under the statistical model described in Section 2. We give sufficient and necessary conditions for cluster recovery, and discuss choices of the weight function. All our results apply to both clustering and bi-clustering settings. In the bi-clustering case, we let $n = \max\{n_1, n_2\}$, and recall that (C_k, C'_k) is the k-th bi-cluster for $k \in [r]$. The minimum cluster size is $K = \min_{k\in[r]} \min\left\{|C_k|, |C'_k|\right\}$. These notations are consistent with the clustering setting, for which n is the total number of nodes and K the minimum cluster size.

Our main result is a general theorem that gives sufficient conditions for the programs (2) and (3) to recover the true cluster matrix Y^*. These conditions are stated in terms of the minimum cluster size K, the label distributions μ and ν, as well as any given weight function $w(\cdot)$ through the quantities

$$\mathbb{E}_\mu w := \int_{\mathcal{L}} w(l)\,\mathrm{d}\mu \quad \text{and} \quad \mathrm{Var}_\mu w := \int_{\mathcal{L}} [w(l) - \mathbb{E}_\mu w]^2\,\mathrm{d}\mu;$$

$\mathbb{E}_\nu w$ and $\mathrm{Var}_\nu w$ are defined similarly. We assume in the sequel that all the relevant integrals and expectations are well-defined. With these notations, we now state our general theorem.

Theorem 2 (Main) *Suppose b is any number that satisfies $|w(l)| \le b$ almost everywhere (a.e.) over $\mathcal{L}$ with respect to μ and ν. There exists a universal constant $c > 0$ such that if*

$$-\mathbb{E}_\nu w \ge c\frac{b\log n + \sqrt{K\log n}\sqrt{\mathrm{Var}_\nu w}}{K}, \tag{5}$$

$$\mathbb{E}_\mu w \ge c\frac{b\log n + \sqrt{n\log n}\sqrt{\max(\mathrm{Var}_\mu w, \mathrm{Var}_\nu w)}}{K}, \tag{6}$$

[3] Both (3) and (4) are strictly feasible, so strong duality holds by Slater's condition.

then Y^* is the unique solution to the programs (2) *and* (3) *with probability at least* $1 - n^{-10}$.[4]

We prove this claim in Section 7. Theorem 2 is in fact a special case of the more general Theorem 9 (Section 3.5), which does not require the boundedness assumption $w(l) \le b$.

Theorem 2 is valid for any given weight function w. Below we discuss how to choose w optimally, and then address the case where w deviates from the optimal choice.

3.1 Optimal Weights

A candidate for a good weight function w can be derived from the maximum likelihood estimator (MLE) of Y^*. Given the observed label matrix L, the log-likelihood of the true cluster matrix taking the value Y is

$$\begin{aligned}\log p\big(L \,|\, Y^* = Y\big) &= \sum_{i,j} \log \left[\mu(L_{ij})^{Y_{ij}} \nu(L_{ij})^{1-Y_{ij}}\right] \\ &= \sum_{i,j} Y_{ij} \underbrace{\log \frac{\mu(L_{ij})}{\nu(L_{ij})}}_{W_{ij}} + \underbrace{\sum_{i,j} \log \nu(L_{ij})}_{c} \\ &= \langle W, Y \rangle + c.\end{aligned}$$

The MLE, which maximizes the above expression over Y, therefore corresponds to using the log likelihood ratio as the weight function:

$$w(l) \leftarrow w^{\text{MLE}}(l) := \log \frac{\mu(l)}{\nu(l)}.$$

Specializing Theorem 2 to the weight function w^{MLE}, we obtain the following theorem that characterizes the performance of using the MLE weights in the convex relaxations. Here $D(\cdot \,\|\, \cdot)$ denotes the KL divergence between two distributions.

Theorem 3 (MLE) *Suppose that $w = w^{MLE}$ is used as the weight function, and b and ζ are any numbers that satisfy with $D(\nu\|\mu) \le \zeta D(\mu\|\nu)$ and $\left|\log \frac{\mu(l)}{\nu(l)}\right| \le b, \forall l \in \mathcal{L}$. There exists a universal constant $c > 0$ such that if*

$$D(\nu\|\mu) \ge c(b+2)\frac{\log n}{K}, \tag{7}$$

$$D(\mu\|\nu) \ge c(\zeta+1)(b+2)\left(\frac{n \log n}{K^2}\right), \tag{8}$$

then with probability at least $1 - n^{-10}$, Y^* is the unique solution to the programs (2) *and* (3). *Moreover, it always holds that $D(\nu\|\mu) \le (2b+3)D(\mu\|\nu)$, so we can take $\zeta = 2b + 3$.*

We prove this claim in Appendix A. The two conditions (7) and (8) are not symmetric due to the fact that there are more cross-cluster pairs than in-cluster pairs in general. The quantity ζ accounts for the asymmetry between $D(\nu\|\mu)$ and $D(\mu\|\nu)$.

[4] In all subsequent results, we use an arbitrary choice of exponent in the failure probability n^{-10}. It is easily seen from Theorem 9 that the constant c scales linearly with the exponent.

Theorem 3 has the intuitive interpretation that the in/cross-cluster label distributions μ and ν should be sufficiently different, measured by their KL divergence, for the underlying clusters to be recovered. Using a classical result in information theory (Topsoe, 2000), we may replace the KL divergences with a quantity called the triangle discrimination that is often easier to work with. This is summarized in the following corollary.

Corollary 4 (MLE 2) *Suppose w^{MLE} is used, and b, ζ are defined as in Theorem 3. There exists a universal constant c such that Y^* is the unique solution to the programs* (2) *and (3) with probability at least* $1 - n^{-10}$ *if*

$$\int_{\mathcal{L}} \frac{(\mu(l) - \nu(l))^2}{\mu(l) + \nu(l)} \,\mathrm{d}\lambda \geq c(\zeta + 1)(b + 2)\left(\frac{n \log n}{K^2}\right). \tag{9}$$

One may take $\zeta = 2b + 3$.

We prove this claim in Appendix A. Note that the left hand side of (9) is the *triangle discrimination* between μ and ν, which lower bounds of the KL-divergence (cf. Lemma 23.) It can be seen that the constant c if Corollary 4 may be trivially chosen such that it only differs from the c in Theorem 3 by a factor of 2. In general, we do not assume any special relationship between universal constants.

The MLE weight function w^{MLE} turns out to be near-optimal, at least in the two-cluster case, in the sense that no other weight function (in fact, no other algorithm) has significantly better statistical performance. This is shown by establishing a necessary condition for *any* algorithm to recover the true clustering Y^*. Here, an algorithm is a measurable function $\hat{Y}$ that maps the data L to a clustering represented by a cluster matrix.

Theorem 5 (Converse) *The following holds for some universal constants $c, c' > 0$. Suppose $K = \frac{n}{2}$, and the quantity b defined in Theorem 3 satisfies $b \leq c'$. If*

$$\int_{\mathcal{L}} \frac{(\mu(l) - \nu(l))^2}{\mu(l) + \nu(l)} \,\mathrm{d}\lambda \leq \frac{c \log n}{n}, \tag{10}$$

then $\inf_{\hat{Y}} \sup_{Y^*} \mathbb{P}\big(\hat{Y}(L) \neq Y^*\big) \geq \frac{1}{2}$, *where the supremum is over all possible cluster matrices.*

We prove this claim in Appendix B.

Under the assumption of Theorem 5, the conditions (9) and (10) match up to a constant factor, showing that program (3) with the MLE weights w^{MLE} is statistically order-wise optimal.

3.2 Monotonicity

We sometimes do not know the exact label distributions μ and ν in computing the MLE weights w^{MLE}. Instead, we may construct the weights using the log likelihood ratios of some "incorrect" distributions $\bar{\mu}$ and $\bar{\nu}$. Our algorithms have a nice *monotonicity* property: as long as the divergence between the true μ and ν is larger than that between $\bar{\mu}$ and $\bar{\nu}$ (hence an "easier" problem), then the algorithm still has the same (if not better) probability of success, even though the wrong weights are used.

More precisely, we say that the pair (μ, ν) is *more divergent* than $(\bar{\mu}, \bar{\nu})$ if, for each $l \in \mathcal{L}$, we have

$$\text{either} \quad \frac{\mu(l)}{\nu(l)} \geq \frac{\mu(l)}{\bar{\nu}(l)} \geq \frac{\bar{\mu}(l)}{\bar{\nu}(l)} \geq 1 \quad \text{or} \quad \frac{\nu(l)}{\mu(l)} \geq \frac{\nu(l)}{\bar{\mu}(l)} \geq \frac{\bar{\nu}(l)}{\bar{\mu}(l)} \geq 1.$$

Theorem 6 (Monotonicity) *Suppose that we use the weight function* $w(l) = \log \frac{\bar{\mu}(l)}{\bar{\nu}(l)}, \forall l \in \mathcal{L}$*, while the actual label distributions are* μ *and* ν*. If the conditions in Theorem 3 or Corollary 4 hold with* μ, ν *replaced by* $\bar{\mu}, \bar{\nu}$*, and* (μ, ν) *is more divergent than* $(\bar{\mu}, \bar{\nu})$*, then with probability at least* $1 - n^{-10}$*,* Y^* *is the unique solution to programs* (2) *and* (3).

We prove this claim in Appendix C.

Theorem 6 suggests that one may choose the weight function by using the log likelihood ratios of a *conservative* estimate (i.e., a less divergent one) of the true label distribution pair.

3.3 Using Inaccurate Weights

We now consider a more general way of choosing the weight function w, which need not be conservative, but is only required to be not too far from the true log-likelihood ratios w^{MLE}. Let

$$\varepsilon(l) := w(l) - w^{\text{MLE}}(l) = w(l) - \log \frac{\mu(l)}{\nu(l)}$$

be the weighting error for each label $l \in \mathcal{L}$. Then the quantities $\Delta_\mu := \int_{\mathcal{L}} \varepsilon(l) \, \mathrm{d}\mu$ and $\Delta_\nu := \int_{\mathcal{L}} \varepsilon(l) \, \mathrm{d}\nu$ represent the average errors with respect to μ and ν. Note that Δ_μ and Δ_ν can be positive or negative. The theorem below characterizes the performance of using such an inaccurate weight function w.

Theorem 7 (Inaccurate Weights) *Let* b *and* ζ *be defined as in Theorem 3. Suppose that the weight function* w *satisfies*

$$|w(l)| \le \alpha \left| \log \frac{\mu(l)}{\nu(l)} \right|, \; \forall l \in \mathcal{L}, \quad |\Delta_\mu| \le \gamma D(\mu \| \nu), \quad \textit{and} \quad |\Delta_\nu| \le \gamma D(\nu \| \mu)$$

for some numbers $\alpha > 0$ *and* $\gamma < 1$*. Then* Y^* *is unique solution to the programs* (2) *and* (3) *with probability at least* $1 - n^{-10}$ *provided that*

$$D(\nu \| \mu) \ge c \frac{\alpha^2}{(1-\gamma)^2} (b+2) \frac{\log n}{K} \quad \textit{and} \quad D(\mu \| \nu) \ge c \frac{\alpha^2}{(1-\gamma)^2} (\zeta + 1)(b+2) \left(\frac{n \log n}{K^2} \right).$$

We prove this claim in Appendix D.

Therefore, as long as the errors Δ_μ and Δ_ν in w are not too large, the condition for recovery is order-wise similar to that of the MLE weight given in Theorem 3. The numbers α and γ measure the level of inaccuracy in the weight function w with respect to the ideal choice w^{MLE}. The last two conditions in Theorem 7 thus quantify the relation between the inaccuracy in w and the statistical price to pay for using such a weight.

3.4 Linear Weights

We next consider a weight function that is an alternative to the MLE weights, and may be preferable in certain scenarios. It is based on a linear approximation to the MLE weight function, hence referred to as the *linear weights*:

$$w^{\text{LIN}}(l) := \frac{\mu(l) - \nu(l)}{\mu(l) + \nu(l)}.$$

It is straightforward to show that

$$\mathbb{E}_\mu w^{\text{LIN}} = -\mathbb{E}_\nu w^{\text{LIN}} = \frac{\text{Var}_\mu w^{\text{LIN}} + \text{Var}_\nu w^{\text{LIN}}}{2} = \frac{1}{2}\int_{\mathcal{L}} \frac{(\mu(l)-\nu(l))^2}{\mu(l)+\nu(l)}\,\mathrm{d}\lambda.$$

Using this observation, we immediately obtain the following corollary for w^{LIN} from Theorem 2.

Corollary 8 (Linear) *Suppose the weight function w^{LIN} is used. There exists a universal constant c such that Y^* is the unique solution to the program* (3) *with probability at least* $1 - n^{-10}$ *if*

$$\int_{l\in\mathcal{L}} \frac{(\mu(l)-\nu(l))^2}{\mu(l)+\nu(l)}\,\mathrm{d}\lambda \geq c\frac{n\log n}{K^2}. \tag{11}$$

Note the the left hand side of (11) coincides with the triangle discrimination term that has appeared in Corollary 4 as a lower bound of the KL-divergence. Comparing Corollary 8 above to the converse results in Theorem 5, we see that, at least for the case of two equal-size clusters, the linear weight function w^{LIN} provides order-wise optimal recovery guarantees.

The linear weights $w^{\text{LIN}}(l)$ are always between -1 and 1 and thus well bounded. Empirically, it is observed that w^{LIN} performs slightly worse than the MLE weight function w^{MLE} in general. However, in certain cases w^{LIN} can actually outperform w^{MLE}. In particular, comparing Corollaries 4 and 8 suggests that this most likely happens when the quantity ζ is large, i.e., $D(\nu\|\mu) \gg D(\mu\|\nu)$. We demonstrate this phenomenon in our empirical results in Section 6.2, where ζ is large in the case of dense graphs.

3.5 Unbounded Weights

The general result in Theorem 2 is valid for any weight function that is uniformly bounded, i.e., $|w(l)| \leq b$ a.e. on $\mathcal{L}$. (The theorem becomes vacuous if $b \to \infty$.) We now give a more general result, which allows some of the weights to have arbitrarily large magnitudes. Unbounded weights arise when some of the pairwise observations are highly certain, or given as hard constraints. For example, the label between nodes i and j may have the form $l_{\text{same}} =$ "these nodes are known to be in the same cluster", in which case assigning an unbounded weight $w(l_{\text{same}}) \to \infty$ in the program (1) forces the nodes i and j to be clustered together. Similarly, for two nodes that are known to be in different clusters, a large negative weight would be desirable.

We identify the set of labels associated with unbounded weights as

$$\mathcal{L}_\infty := \{l \in \mathcal{L} : |w(l)| > b_\infty\},$$

where b_∞ should be thought of as a very large positive number. For all $l \notin \mathcal{L}_\infty$, we assume that $|w(l)| \leq b$ for some $b < b_\infty$. We further assume that the weight function w is $\mathcal{L}_\infty$*-consistent* in the sense that

$$\mu(l) = 0, \;\; \forall l \in \mathcal{L}_\infty : w(l) < 0 \qquad \text{and} \qquad \nu(l) = 0, \;\; \forall l \in \mathcal{L}_\infty : w(l) > 0.$$

This condition ensures that whenever a node pair is assigned with unbounded weight, the sign of the weight is consistent with the true cluster relation between the pair, that is, with probability one,

$$W_{ij}(2Y^*_{ij} - 1) > 0, \quad \forall (i,j) : L_{ij} \in \mathcal{L}_\infty. \tag{12}$$

Denote by $s_\mu := \int_{\mathcal{L}_\infty} \mathrm{d}\mu$ the total probability of the labels in $\mathcal{L}_\infty$ under the distribution μ, and

$$\tilde{w}_b(l) := \begin{cases} 0 & \text{if } l \in \mathcal{L}_\infty, \\ w(l) & \text{otherwise}, \end{cases}$$

the weight function restricted to labels with bounded weights. Finally, in the bi-clustering setting, we need an additional parameter ξ that measures bi-cluster skewness:

$$\xi := \max_{k \in [r]} \left\{ \sqrt{\frac{|C_k|}{|C'_k|}}, \sqrt{\frac{|C'_k|}{|C_k|}} \right\}.$$

In the standard clustering case we simply have $\xi = 1$.

Under the above setting, we have the following result valid for unbounded weights with arbitrarily large b_∞.

Theorem 9 *Suppose that the weight function w is $\mathcal{L}_\infty$-consistent, and $\beta > 0$ is any fixed number. There exists a universal constant $c > 0$ such that the following is true. If*

$$\mathbb{E}_\mu \tilde{w}_b \geq 0, \tag{13}$$

$$-\mathbb{E}_\nu \tilde{w}_b > c\, \frac{b\beta \log n + \sqrt{K\beta \log n}\sqrt{\mathrm{Var}_\nu \tilde{w}_b}}{K}, \tag{14}$$

and at least one of the following two inequalities holds:

$$\mathbb{E}_\mu \tilde{w}_b > c\, \frac{b\beta \log n + \sqrt{n\beta \log n}\sqrt{\max(\mathrm{Var}_\mu \tilde{w}_b, \mathrm{Var}_\nu \tilde{w}_b)}}{K}, \tag{15}$$

$$s_\mu + \frac{\mathbb{E}_\mu \tilde{w}_b}{b_\infty} > c \max \left\{ \max\left\{1, \frac{b\xi}{b_\infty}\right\} \frac{\beta \log n}{K}, \frac{n \max(\mathrm{Var}_\mu \tilde{w}_b, \mathrm{Var}_\nu \tilde{w}_b)}{K b_\infty^2} \right\}, \tag{16}$$

then with probability at least $1 - n^{-\beta}$ the solution to the programs (2) *and* (3) *is unique and equals Y^*.*

We prove this claim in Section 7.

Theorem 9 is particularly useful when we take $b_\infty \to \infty$ while keeping b, n and K finite. In this case the condition (16) simplifies to

$$s_\mu > c \frac{\beta \log n}{K}, \tag{17}$$

a lower bound on the probability of observing a label from $\mathcal{L}_\infty$ between two nodes in the same cluster. The conditions (17) together with (14) are sufficient for successful recovery of Y^*. Significantly, these conditions only depend *logarithmically* on n, which is in contrast to the dependence on $\sqrt{n}$ in Theorem 2. This improvement demonstrates the benefit of assigning unbounded weights to labels in $\mathcal{L}_\infty$ (i.e., label with high certainty).

Also note that Theorem 2 is a special case of Theorem 9, as we show in Section 7.

4. Consequences and Applications

We now apply the general results in the last section to different concrete cases. In sections 4.1 and 4.2, we consider two clustering settings with sub-Gaussian weights or non-uniform edge probabilities, both of which generalize the standard stochastic block model. In these settings two immediate corollaries of our main theorems recover, and in fact improve upon, existing results. In sections 4.3 and 4.4, we turn to the more complicated setting of clustering time-varying graphs and derive several new results.

Throughout this section we use c, c_0 etc. to denote universal positive constants.

4.1 Clustering a Sub-Gaussian Matrix with Partial Observations

Analogous to the planted partition and stochastic block models for unweighted graphs, the *submatrix localization* problem concerns clustering a weighted graph whose adjacency matrix has sub-Gaussian entries. This problem is often used as an idealized model for the bi-clustering of drugs and proteins, species and gene sequences, customers and products, and other forms of object-feature pairs (see Kolar et al., 2011, and references therein).

Here we consider a generalization of this problem, where some of the entries are unobserved. This setting arises when the relations between some node pairs are costly to determine, so it is unknown whether or not they are connected by an edge, nor is the weight of the edge (Shamir and Tishby, 2011). For simplicity, we state our results assuming that the adjacency matrix and its submatrices are symmetric, and that the entries of the matrix are Gaussian. Our theoretical results apply to the general sub-Gaussian case without change.

Specifically, we observe a matrix $L \in (\mathbb{R} \cup \{?\})^{n\times n}$, where $L_{ij} = ?$ with probability $1 - s$, and otherwise L_{ij} follows the Gaussian distribution $\mathcal{N}(u_{ij}, 1)$.[5] Here $L_{ij} = ?$ denotes that the relation between the i-th and j-th nodes is unobserved. The means $\{u_{ij}\}$ of the Gaussians satisfy the following: within L there are r submatrices of size at least $K \times K$ with disjoint row and column support; we have $u_{ij} = \bar{u}$ if the pair (i, j) is inside one of the submatrices, and $u_{ij} = \underline{u}$ if outside, where $\bar{u} > \underline{u} \geq 0$. The goal is to locate these submatrices with elevated means given the large matrix L. We may think of this problem as finding clusters in a weighted graph with the partially observed adjacency matrix L.

This problem is a special case of our general framework with a mixed-valued label set $\mathcal{L} = \mathbb{R} \cup \{?\}$. The base measure λ is chosen to be such that it equals the Lebesgue measure on $\mathbb{R}$ and 1 on $\{?\}$. Then the density functions of μ and ν are

$$\mu(l) = \begin{cases} 1-s, & \text{if } l = ?, \\ s \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(l-\bar{u})^2}{2}\right), & \text{if } l \in \mathbb{R}, \end{cases} \quad \text{and} \quad \nu(l) = \begin{cases} 1-s, & \text{if } l = ?, \\ s \cdot \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(l-\underline{u})^2}{2}\right), & \text{if } l \in \mathbb{R}. \end{cases}$$

The MLE weight function therefore has the form

$$w^{\text{MLE}}(l) \propto \begin{cases} 0, & \text{if } l = ?, \\ l - (\bar{u} + \underline{u})/2, & \text{if } l \in \mathbb{R}. \end{cases}$$

[5] Equivalently, L_{ij} is generated according to $\mathcal{N}(u_{ij}, 1)$, and then is replaced by ? with probability $1 - s$.

This submatrix localization problem is most interesting when $\bar{u} - \underline{u} \le c_0\sqrt{\log n}$ for some universal constant $c_0 > 0$,[6] which we assume to hold in the sequel. Observe that $D\left(\mu \| \nu\right) = D\left(\nu \| \mu\right) = \frac{1}{4}s(\bar{u} - \underline{u})^2$. In Appendix E we apply the general result in Theorem 2 to derive the following recovery guarantee.

Corollary 10 (Gaussian Graphs) *Under the above setting, if*

$$s\left(\bar{u} - \underline{u}\right)^2 \ge c\frac{n\log^3 n}{K^2}, \tag{18}$$

then with probability at least $1 - 2n^{-10}$, Y^* *is the unique solution to the programs* (2) *and* (3) *with the weight function* $w = w^{MLE}$.

In the full observation case $s = 1$, Corollary 10 recovers the results in Ames (2014); Balakrishnan et al. (2011); Kolar et al. (2011) up to log factors. Our results are more general as we allow for partial observation, which is not considered in previous work. Corollary 10 allows the observation probability to be as small as $s = \Omega(\frac{n}{K^2})$, or the signal strength to be $\bar{u} - \underline{u} = \Omega(\frac{\sqrt{n}}{K})$ (ignoring log factors). In general we see a quadratic tradeoff between these two quantities: with four times more observations, the signal strength can be 50% smaller.

4.2 Planted Partition with Non-Uniform Edge Probabilities

Recall that in the standard planted partition model, the observed unweighted graph is generated by independently connecting each node pair with probability p if they are in the same cluster, or with probability q if they are in different clusters, where $p, q \in [0, 1]$ are two fixed numbers. Here we consider a more general setting, where the in/cross-cluster edge probabilities can be different across node pairs. This setting has a range of applications as discussed at the end of this sub-section.

Concretely, suppose that each node pair (i, j) is associated with two numbers P_{ij} and Q_{ij}. Independently for each (i, j), the values of P_{ij} and Q_{ij} are generated randomly according to some distribution ψ on $[0, 1] \times [0, 1]$, which is assumed to have a density function with respect to some reference measure λ_0. Conditioned on P_{ij} and Q_{ij}, the nodes i and j are connected by an edge with probability P_{ij} if they are in the same cluster, or with probability Q_{ij} if they are in different clusters. For each (i, j), one knows the values of P_{ij} and Q_{ij}, but not which of them is the probability that generates the edge. Assuming that there are r ground-truth clusters of size at least K, the goal is to find these clusters given the graph adjacency matrix $A \in \{0, 1\}^{n\times n}$ and the edge probabilities (P_{ij}) and $(Q_{ij}) \in [0, 1]^{n\times n}$.

This model can be cast as a special case of our labeled framework, in which each pairwise label is a triplet of the form $L_{ij} = (A_{ij}, P_{ij}, Q_{ij}) \in \mathcal{L} = \{0, 1\} \times [0, 1] \times [0, 1]$. Note that the labels take a combination of discrete and real values. If we choose the base measure λ to be the product of the counting measure on $\{0, 1\}$ and λ_0, then the density functions of μ and ν are given as follows: for each $l = (a, p, q) \in \{0, 1\} \times [0, 1] \times [0, 1]$,

$$\mu(l) = \begin{cases} p\psi(p, q), & \text{if } a = 1, \\ (1 - p)\psi(p, q), & \text{if } a = 0, \end{cases} \qquad \text{and} \qquad \nu(l) = \begin{cases} q\psi(p, q), & \text{if } a = 1, \\ (1 - q)\psi(p, q), & \text{if } a = 0. \end{cases}$$

[6] Otherwise, if $\bar{u} - \underline{u} > c_0\sqrt{\log n}$ for some sufficiently large constant $c_0 > 0$, then with high probability the smallest entry inside the submatrices will dominate the largest entry outside, so simple element-wise thresholding finds the submatrices (Balakrishnan et al., 2011; Kolar et al., 2011).

The MLE weight function therefore has the form

$$w^{\text{MLE}}(l) = w^{\text{MLE}}\big((a,p,q)\big) = a\log\frac{p}{q} + (1-a)\log\frac{1-p}{1-q}. \tag{19}$$

Applying Corollary 4 for MLE weights, we immediately obtain the following recovery guarantee:

Corollary 11 (Non-uniform Edge Probabilities, I) *Under the above setting, suppose that there exist ζ and b such that $D(\nu\|\mu) \le \zeta D(\mu\|\nu)$ and $|\log\frac{\mu(l)}{\nu(l)}| \le b$ almost everywhere. If*

$$\mathbb{E}_{(P,Q)\sim\psi}\left[\frac{(P-Q)^2}{(P+Q)(2-P-Q)}\right] \ge c\,(\zeta+1)(b+2)\frac{n\log n}{K^2}.$$

then with probability at least $1-n^{-10}$, Y^* is the unique solution to the programs (2) *and* (3) *with the MLE weight function in* (19)*. One may take $\zeta = 2b+3$.*

Here the notation $(P,Q)\sim\psi$ means that (P,Q) is a pair of numbers sampled from the distribution ψ.

We can simplify the above condition considerably by considering a conservative weight function (cf. Section 3.2): on the RHS of (19) we replace p and q with $\bar{p} = \frac{3}{4}p + \frac{1}{4}q$ and $\bar{q} = \frac{1}{4}p + \frac{3}{4}q$, respectively, which leads to the weight function

$$\bar{w}^{\text{MLE}}(l) = \bar{w}^{\text{MLE}}\big((a,p,q)\big) = a\log\frac{3p+q}{p+3q} + (1-a)\log\frac{4-3p-q}{4-p-3q}. \tag{20}$$

These weights are clearly bounded by a constant. Applying Theorem 6 for monotonicity together with Corollary 4 for MLE weights, we obtain the following recovery guarantee:

Corollary 12 (Non-uniform Edge Probabilities, II) *Under the above setting, suppose that $P \ge Q$ almost surely with respect to ψ. If*

$$\mathbb{E}_{(P,Q)\sim\psi}\left[\frac{(P-Q)^2}{P(1-Q)}\right] \ge c\,\frac{n\log n}{K^2},$$

then with probability at least $1-n^{-10}$, Y^* is the unique solution to the programs (2) *and* (3) *with the conservative MLE weight function $\bar{w}^{\text{MLE}}$ in* (20)*.*

As a passing observation, we note the $\log n$ factor in the above two corollaries can be removed by a slightly more careful analysis; see Remark 20.

Below we discuss two applications of Corollary 12, which leads to generalization and improvement over existing results.

4.2.1 CLUSTERING PARTIALLY OBSERVED UNWEIGHTED GRAPHS

In Section 4.1 we discussed clustering Gaussian weighted graphs with partial observations. Here we consider such a similar setting for unweighted graphs under the planted partition model.[7]

[7] The planted partition model satisfies the sub-Gaussian assumption, so the results in Section 4.1 in fact apply to this setting. Here we take into account the variance information to derive tighter bounds.

Specifically, suppose that the edge probabilities $(P_{ij}, Q_{ij}) \sim \psi$ is distributed as $(P_{ij}, Q_{ij}) = (p, q)$ with some probability s, and $P_{ij} = Q_{ij} = \frac{1}{2}$ with probability $1 - s$, where $p > q$ are two fixed numbers. This setting extends the standard planted partition model to partial observation: with probability $1 - s$, the connection A_{ij} between a pair (i, j) is known to be purely random and uninformative about the cluster membership of i and j—having such a purely random observation is equivalent to saying that one does not observe whether or not i and j are connected by an edge.[8] This setting is sometimes called the planted partition model with partial or *censored* edge observation. Note that the conservative MLE weight function (20) assigns zero weight to unobserved/uninformative pairs, as should be expected.

By calculating the left hand side of the condition in Corollary 12, we immediately obtain the following guarantee.

Corollary 13 *Under the above planted partition model with partial observation, if*

$$\frac{s(p-q)^2}{p(1-q)} \geq c\,\frac{n \log n}{K^2},$$

then with probability at least $1 - n^{-10}$*,* Y^* *is the unique solution to the programs* (2) *and* (3) *with the conservative MLE weight function* $\bar{w}^{\textit{MLE}}$ *in* (20)*.*

In the full observation setting $s = 1$, the above result matches the best existing bounds for standard planted partition (e.g., in Chen et al., 2014c; Anandkumar et al., 2014), up to a $\log n$ factor that can be removed (see the remark after Corollary 12). In the partial observation setting $s < 1$, the work by Vinayak et al. (2014); Chen et al. (2014a) gives a similar bound under the additional assumption $p > 0.5 > q$, which is not required by Corollary 13. For general p and q, the best existing bounds can be found in Oymak and Hassibi (2011); Chen et al. (2014c), where unobserved entries in the adjacency matrix A are replaced with 0 and recovery is guaranteed under the condition $\frac{s(p-q)^2}{p(1-sq)} \gtrsim \frac{n \log n}{K^2}$.[9] Our result is tighter when p and q are close to 1. Finally, we note that our result is non-asymptotic and valid under any scaling of the parameters n, K, r, p, q, s.

4.2.2 Planted Partition with Non-uniform Uncertainty

We next consider an application of Corollary 12 to a further generalization of the partial observation setting above. In the above setting, if an entry A_{ij} of the adjacency matrix is unobserved or purely random, the cluster relation between the nodes i and j is completely uncertain (based on only A_{ij}). Here we assume that each A_{ij} is associated with a different, continuous level of uncertainty. For some node pairs, the observation A_{ij} of the existence of an edge (or respectively, the absence of an edge) may be generated by accurate measurements, and therefore a strong indicator that the nodes i and j should be assigned to the same (or respectively, different) clusters.

For example, in *crowd-clustering* a number of users are asked whether or not they think a pair of nodes (e.g., movies or images) are similar, and the final graph is constructed by aggregating the users' answers by say majority voting (Gomes et al., 2011; Yi et al., 2012). The uncertainty levels are naturally non-uniform across pairs: a pair receiving a large number of unanimous votes

[8] This model can be rephrased in the following equivalent form: given a partially observed adjacency matrix $A \in \{0, 1, ?\}^{n \times n}$, one replaces each unobserved entry ? in A with an independent Bernoulli random variable.

[9] We use $\gtrsim$ when the inequality is up to a constant factor.

is associated with high confidence, whereas those with a few votes or divergent votes have low confidence, and a pair receiving no votes is completely uncertain.

In such a setting, each pairwise observation A_{ij} should be treated differently according to its level of uncertainty. Often, one has prior knowledge on the uncertainty levels, for example when the graph is built from a known process as in the crowd-clustering example. Intuitively, using such knowledge improves clustering performance. It is, however, less clear how this knowledge can be used and how much improvement it can provide. Below we use Corollary 12 to obtain a quantitative answer.

For simplicity, we focus on a special case of the setting in Corollary 12: we assume that the distribution ψ of edge probabilities is symmetric, in the sense that we always have $Q_{ij} \equiv 1 - P_{ij}$, where $Q_{ij} \in [0, \frac{1}{2}]$.[10] In this case, the quantity Q_{ij} can be interpreted as the probability of an erroneous observation—namely, a no-edge between two nodes in the same cluster (a false negative), or an edge between two nodes in different clusters (a false positive). Therefore, Q_{ij} measures the level of uncertainty in the observation A_{ij} at (i, j). In particular, $Q_{ij} = 0$ means that A_{ij} is a noiseless observation of the cluster relation between i and j, whereas $Q_{ij} = \frac{1}{2}$ corresponds to a completely uncertain observation with no information.

For each observation $A_{ij} \in \{0, 1\}$ with an uncertainty level $Q_{ij} \in [0, \frac{1}{2}]$, the conservative MLE weight function (20) assigns the weight

$$W_{ij} = (2A_{ij} - 1) \log \frac{3 - 2Q_{ij}}{1 + 2Q_{ij}}.$$

Note that the magnitude $|W_{ij}|$ of the weight is small for a high uncertainty level Q_{ij}; in particular, one has $W_{ij} = 0$ for a completely uncertain observation with $Q_{ij} = \frac{1}{2}$.

Using these weight, Corollary 12 guarantees that the programs (2) and (3) recover the clusters with high probability provided that

$$\mathbb{E}\left[\left(\frac{1}{2} - Q\right)^2\right] \gtrsim \frac{n \log n}{K^2},$$

where the expectation is with respect to $(P, Q) \sim \psi$. We can compare this result with the presumably sub-optimal unweighted approach, which ignores the non-uniformity of the uncertainty level and uses uniform weights $W_{ij} = 2A_{ij} - 1 \in \{-1, 1\}$. By Theorem 2 this unweighted approach succeeds if

$$\left(\frac{1}{2} - \mathbb{E}Q\right)^2 \gtrsim \frac{n \log n}{K^2}.$$

The difference between the left hand sides of the last two conditions is $\mathbb{E}[Q^2] - (\mathbb{E}Q)^2 = \text{Var}[Q]$, the variance of the uncertainty level.[11] This is therefore evidence that the weighted approach is indeed better than the unweighted one, and the gain is large precisely when the uncertainty level is very non-uniform with a high variance. We refer the readers to Chen et al. (2014b) for empirical verification of this gain as well as further discussion and applications.

Of course our results are not limited to the symmetric setting $Q_{ij} \equiv 1 - P_{ij}$. Corollary 12 is applicable under a general distribution ψ of P_{ij} and Q_{ij}, in which case a larger value of $|P_{ij} - Q_{ij}|$ corresponds to a lower level of uncertainty. We omit discussion of this general setting due to space limit.

[10] That is, the density function $\psi(p, q)$ is supported on the line segment $\{(p, q) \in \mathbb{R}^2 : q = 1 - p, 0 \leq q \leq \frac{1}{2}\}$.

[11] Asymptotically the difference between the right hand sides can be made arbitrarily small, regardless of the contants. Furthermore, empirically the constants involved are very similar, depending on how the weights are bounded.

4.3 Clustering Time-varying Multiple-snapshot Graphs

Standard graph clustering concerns the partition of a single, static graph. We now consider a setting where the graph is time-varying. Specifically, for each time interval $t = 1, 2, \ldots, T$, one observes a snapshot of the (label) graph $L^{(t)} \in \mathcal{L}^{n\times n}$. In this subsection, we assume that each snapshot is generated by the distributions μ and ν independently of other snapshots. In the next subsection we consider the more general setting of Markov snapshots.

We can map this problem into our labeled framework, by considering the whole time sequence of $\bar{L}_{ij} := (L_{ij}^{(1)}, \ldots, L_{ij}^{(T)})$ as a single label observed at the pair (i, j). In this case the label set is the set of all possible sequences, i.e., $\bar{\mathcal{L}} = \mathcal{L}^T$, and the label distributions are (with a slight abuse of notation) given by the products $\mu(\bar{L}_{ij}) = \prod_{t=1}^{T} \mu(L_{ij}^{(t)})$ and $\nu(\bar{L}_{ij}) = \prod_{t=1}^{T} \nu(L_{ij}^{(t)})$. The MLE weight (normalized by T) is therefore the average log-likelihood ratio:

$$w^{\text{MLE}}(\bar{L}_{ij}) = \frac{1}{T}\log\frac{\mu(\bar{L}_{ij})}{\nu(\bar{L}_{ij})} = \frac{1}{T}\sum_{t=1}^{T}\log\frac{\mu(L_{ij}^{(t)})}{\nu(L_{ij}^{(t)})}.$$

Since the weight $w^{\text{MLE}}(\bar{L}_{ij})$ is the average of T independent random variables, its variance scales as $\frac{1}{T}$. Applying Theorem 2 and following almost identical arguments as in the proof of Theorem 3, we obtain the following guarantee for clustering a time-varying graph with independent snapshots.

Corollary 14 (Independent Snapshots) *Suppose that* $|\log\frac{\mu(l)}{\nu(l)}| \le b, \forall l \in \mathcal{L}$ *and* $D(\nu\|\mu) \le \zeta D(\mu\|\nu)$. *If*

$$D(\nu\|\mu) \ge c(b+2)\frac{\log n}{K} \qquad \textit{and} \tag{21}$$

$$D(\mu\|\nu) \ge c(b+2)\max\Big\{\frac{\log n}{K}, (\zeta+1)\frac{n\log n}{TK^2}\Big\}, \tag{22}$$

then with probability at least $1 - n^{-10}$, Y^* *is the unique solution to the programs* (2) *and* (3) *with the MLE weights given above.*

Note that setting $T = 1$ above recovers the result in Theorem 3 for clustering a single label graph.

The second term on the right hand side of (22) usually dominates. In this case, Corollary 14 says that the clustering problem becomes easier if T is larger (i.e., more snapshots of the graph are observed), and the relation between T and cluster recovery is quantified precisely. As a concrete example, suppose that each snapshot is generated by the standard planted partition model with edge probabilities p and q, where $q = p/2$. By Corollary 14 we need $p \gtrsim \frac{n\log n}{TK^2}$ to guarantee cluster recovery. Therefore, if four times more snapshots are available, then one can recover clusters whose sizes K are 50% smaller. Similarly, we see a tradeoff between the number of snapshots T and the graph sparsity p.

4.4 Markov Sequence of Snapshots

We now consider a more general and useful setting for time-varying graphs, where the graph snapshots are not assumed to be independent but instead form a Markov chain.

For simplicity we assume that the Markov chain is time-invariant and has a finite state space and a unique stationary distribution that is also the initial distribution. Therefore, the observations

$L_{ij}^{(t)}, t = 1, 2, \ldots$ at each pair (i, j) are generated by first drawing a label $L_{ij}^{(1)}$from the stationary distribution μ_0 (or ν_0, according to the cluster membership of i and j) over the finite set $\mathcal{L}$ at $t = 1$, then applying a one-step transition to obtain the label at each subsequent t. In particular, given the previously observed label l, let the intra-cluster and inter-cluster conditional distributions of the next observation be $\mu(\cdot|l)$ and $\nu(\cdot|l)$. We assume that the Markov chains $\{L_{ij}^{(1)}, L_{ij}^{(2)}, \ldots\}$ with respect to both μ and ν are geometrically ergodic, in the sense that for each integer $\tau \geq 1$ and pair $L_{ij}^{(1)}, L_{ij}^{(\tau+1)}$,

$$\left| \Pr_\mu \left(L_{ij}^{(\tau+1)} | L_{ij}^{(1)} \right) - \mu_0\big(L_{ij}^{(\tau+1)}\big) \right| \leq \kappa\phi^\tau \text{ and } \left| \Pr_\nu \left(L_{ij}^{(\tau+1)} | L_{ij}^{(1)} \right) - \nu_0\big(L_{ij}^{(\tau+1)}\big) \right| \leq \kappa\phi^\tau \quad (23)$$

for some constants $\kappa \geq 1$ and $0 < \phi < 1$ that only depend on μ and ν. Let $D_l(\mu\|\nu)$ denote the KL-divergence between $\mu(\cdot|l)$ and $\nu(\cdot|l)$; $D_l(\nu\|\mu)$ is similarly defined. Let

$$\mathbb{E}_{\mu_0} D_l(\mu\|\nu) := \textstyle\sum_{l\in\mathcal{L}} \mu_0(l) D_l(\mu\|\nu)$$

and similarly for $\mathbb{E}_{\nu_0} D_l(\nu\|\mu)$. As in the previous subsection, we use the average log-likelihood ratio $w^{\text{MLE}}(\bar{L}_{ij}) = \frac{1}{T}\log\frac{\mu(\bar{L}_{ij})}{\nu(\bar{L}_{ij})}$ as the weight, where $\mu(\bar{L}_{ij})$ is the joint distribution of the sequence $\bar{L}_{ij} = \big(L_{ij}^{(1)}, \ldots, L_{ij}^{(T)}\big)$ under the above Markov chain; similarly for $\nu(\bar{L}_{ij})$. Finally, define the quantity

$$\Phi := \frac{\kappa}{(1-\phi)\min_{l\in\mathcal{L}}\{\mu_0(l), \nu_0(l)\}}.$$

With these notations, we have the following corollary of Theorem 2.

Corollary 15 (Markov Snapshots) *Under the above setting, suppose that for each label pair* $(l, l') \in \mathcal{L}\times\mathcal{L}$, *we have* $\left|\log\frac{\mu_0(l)}{\nu_0(l)}\right| \leq b$, $\left|\log\frac{\mu(l'|l)}{\nu(l'|l)}\right| \leq b$, $D(\nu_0\|\mu_0) \leq \zeta D(\mu_0\|\nu_0)$ *and* $\mathbb{E}_{\nu_0} D_l(\nu\|\mu) \leq \zeta\mathbb{E}_{\mu_0} D_l(\mu\|\nu)$ *for some numbers* b *and* ζ. *If*

$$\frac{1}{T}D(\nu_0\|\mu_0) + \Big(1-\frac{1}{T}\Big)\mathbb{E}_{\nu_0}D_l(\nu\|\mu) \geq c(b+2)\frac{\log n}{K}, \quad (24)$$

$$\frac{1}{T}D(\mu_0\|\nu_0) + \Big(1-\frac{1}{T}\Big)\mathbb{E}_{\mu_0}D_l(\mu\|\nu) \geq c(b+2)\max\Big\{\frac{\log n}{K}, (\zeta+1)\Phi\frac{n\log n}{TK^2}\Big\}, \quad (25)$$

then with probability at least $1 - n^{-10}$, Y^* *is the unique solution to the programs* (2) *and* (3) *with MLE weight function* w^{MLE}.

See Appendices F for the proof of this claim, and G for additional discussion of the assumptions.

As an illuminating example, consider the case where $\mu_0 \approx \nu_0$, i.e., the marginal distributions for individual snapshots are very close or even identical in and across clusters. This means that the information about cluster membership is not contained in the labels themselves in individual snapshots, but instead in the *change* of labels between snapshots. This point is made evident in the left hand sides of (24) and (25), as first terms therein are approximately zero. In this case, *it is necessary to use the temporal information* in order to perform clustering. Such information would be lost if one disregards the ordering of the snapshots (for example, by averaging the snapshots) and then applies a single-snapshot clustering algorithm. This example therefore highlights a crucial difference between clustering time-varying graphs and static graphs.

5. Implementation

The convex programs (2) and (4) used in our clustering approach are semidefinite programs and can be solved efficiently by the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011). We provide the pseudocode for a complete implementation of the programs (2) and (4) in Algorithms 1 and 2 below.

Algorithm 1 ADMM solver for Program (2)

```
Input: Weight matrix $W \in \mathbb{R}^{n \times n}$ symmetric, convergence threshold $\epsilon > 0$
Output: $Y$

1. $\rho \leftarrow 1, k \leftarrow 0$
2. $Y^k \leftarrow 0, Q^k \leftarrow 0, (Y^k, Q^k \in \mathbb{R}^{n \times n})$
3. $X^{k+1} \leftarrow U \max\{\Lambda, 0\} U^\top$, where $U \Lambda U^\top$ is an eigen-decomposition of $(Y^k - Q^k + \frac{1}{\rho} W)$.
4. $Y^{k+1} \leftarrow \min\left\{\max\left\{X^{k+1} + Q^k, 0\right\}, 1\right\}$
5. $Q^{k+1} \leftarrow Q^k + X^{k+1} - Y^{k+1}$
6. If $\|X^{k+1} - Y^{k+1}\|_F \leq \epsilon \max\{\|X^{k+1}\|_F, \|Y^{k+1}\|_F\}$ and $\|Y^{k+1} - Y^k\|_F \leq \epsilon \|Q^{k+1}\|_F$ then
   stop and output $Y = Y^{k+1}$.
7. (Optional) Update $\rho$ and $Q^{k+1}$
8. $k \leftarrow k + 1$, go to step 3.
```

The inputs and outputs of Algorithms 1 and 2 are the same terms used in the programs (2) and (4), respectively. We find that in practice, using the tuning parameter $\eta = \sqrt{2n}$ for the program (4) works well. The criterion for convergence is specified by the threshold $\epsilon > 0$, and using $\epsilon = 10^{-4}$ provides a good tradeoff between the convergence time and the quality of the solution. All our experiment results in Section 6 are based on these choices of η and ϵ.

In both Algorithms 1 and 2, an optional Step 7 for updating ρ can be used to potentially improve the speed of convergence. Boyd et al. (2011) suggest one such updating rule, which takes an additional parameter τ and aims to balance the primal and dual residuals. In particular, if $\|X^{k+1} - Y^{k+1}\|_F > \tau\rho\|Y^{k+1} - Y^k\|_F$, then set $\rho \leftarrow 2\rho$ and $Q^{k+1} \leftarrow Q^{k+1}/2$. On the other hand, if $\tau\|X^{k+1} - Y^{k+1}\|_F < \rho\|Y^{k+1} - Y^k\|_F$, then set $\rho \leftarrow \rho/2$ and $Q^{k+1} \leftarrow 2Q^{k+1}$. Typically $\tau = 10$ is a stable choice, which we use in all our experiments. For further details of this updating rule we refer the reader to Boyd et al. (2011).

In practice, due to finite precision and numerical errors, the output matrix Y will not in general have entries that are exactly 0 or 1, even if the true cluster matrix Y^* is in fact the unique optimal solution. If this is the case, a simple rounding of Y will give the correct solution, from which the clusters can then be obtained by sorting the rows. If Y can not be rounded into a cluster matrix, we use a simple k-means algorithm to the rows of Y to extract a desired number of clusters.

Algorithm 2 ADMM solver for Program (4)

```
Input: Weight matrix $W \in \mathbb{R}^{n_1 \times n_2}$, tuning parameter $\eta > 0$, convergence threshold $\epsilon > 0$
Output: $Y$

  1. $\rho \leftarrow 1, k \leftarrow 0$
  2. $Y^k \leftarrow 0, Q^k \leftarrow 0, (Y^k, Q^k \in \mathbb{R}^{n_1 \times n_2})$
  3. $X^{k+1} \leftarrow U \max\{\Sigma - \frac{\eta}{\rho}, 0\} V^\top$, where $U \Sigma V^\top$ is an SVD of $(Y^k - Q^k)$.
  4. $Y^{k+1} \leftarrow \min\left\{\max\left\{X^{k+1} + Q^k + \frac{1}{\rho} W, 0\right\}, 1\right\}$
  5. $Q^{k+1} \leftarrow Q^k + X^{k+1} - Y^{k+1}$
  6. If $\|X^{k+1} - Y^{k+1}\|_F \le \epsilon \max\{\|X^{k+1}\|_F, \|Y^{k+1}\|_F\}$ and $\|Y^{k+1} - Y^k\|_F \le \epsilon \|Q^{k+1}\|_F$ then
     stop and output $Y = Y^{k+1}$.
  7. (Optional) Update $\rho$ and $Q^{k+1}$
  8. $k \leftarrow k + 1$, go to step 3.
```

6. Empirical results

In this section we report empirical results by applying Algorithm 1 to a variety of both synthetic and real data sets.[12] In our experiments, unless specified otherwise, we report the "full recovery rate" based on 100 repeated trials, i.e., the fraction of trials where the output of Algorithm 1 (after rounding to the nearest 0 and 1) equals Y^* exactly. Error bars show 95% confidence intervals.

6.1 Clustering with General Labels

We first evaluate graph clustering performance on a generic graph model with 5 labels. We use $n = 200$ with 4 equal-size clusters. In each experiment, two distributions μ and ν are randomly chosen from a uniform prior over all distributions as the in-cluster and the cross-cluster label distributions. Then, 100 random graphs are generated using this (μ, ν) pair, and each clustering outcome is checked against the corresponding ground truth. This is repeated 500 times to get a large variety of pairs. In other words, 500 random (μ, ν) pairs are generated, each tested on 100 random graphs. According to Theorem 3, the KL-divergence between μ and ν is the key deciding factor for the successful recovery of the underlying true clusters. The results are shown in Figure 1. In the left panel of Figure 1, we use the sum $D(\mu\|\nu) + D(\nu\|\mu)$ as the predictor (the horizontal axis is cut off at 2 since beyond this range all recovery rates are essentially 1). In the right panel, we use $\min\{D(\mu\|\nu), D(\nu\|\mu)\}$ as the predictor. The results indeed support the theoretical prediction as given by conditions (7) and (8) of Theorem 3.

[12] Although we only report results from Algorithm 1 and for standard clustering, we note that similar results are obtained with Algorithm 2 and in the bi-clustering case as well.

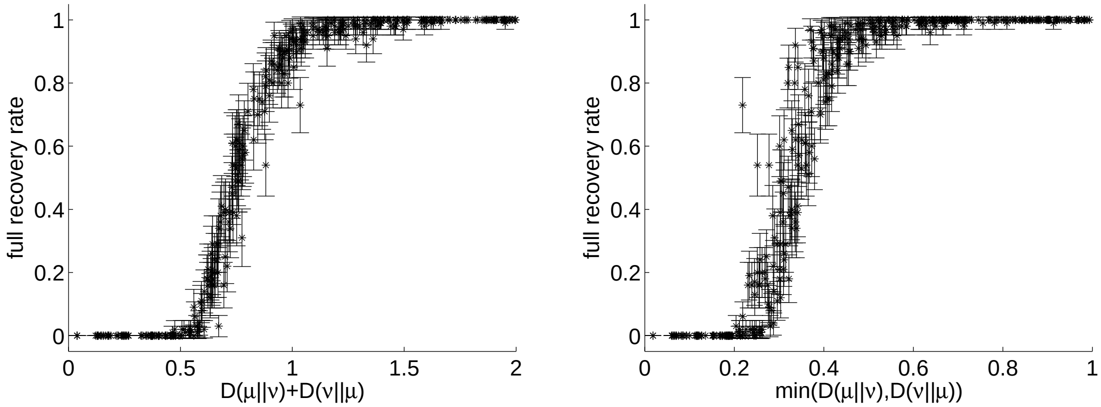


Figure 1: Clustering performance under different label distributions

6.2 Clustering Sparse/Dense Graphs with Partial Observations

In the next experiment, we test the planted partition model with partial observations. The model is as described in Section 4.2. Each pair is observed with probability s. For each observed in-cluster pairs, an edge is generated with probability p while for each observed cross-cluster pairs, an edge is generated with probability q. We consider both sparse graphs (p and q close to 0) and dense graphs (p and q close to 1). For the sparse case, we fix $q = 0.02$ and vary p, whereas for the dense case, we fix $p = 0.98$ and vary q.

Figures 2 and 3 show the results for graphs with $n = 200$, and Figures 4 and 5 show the results for $n = 1000$, with 4 equal-size clusters in both cases. For $n = 200$, we set the observation probability to $s = 0.8$; for $n = 1000$, we use $s = 0.5$ since the problem is significantly easier.

For comparison, we include results for the MLE weights (w^{MLE}), the linear weights (w^{LIN}), and the uniform weights. An imputation scheme labeled “MLE (no partial)”, where all unobserved entries are treated as “no edge”, is also included.

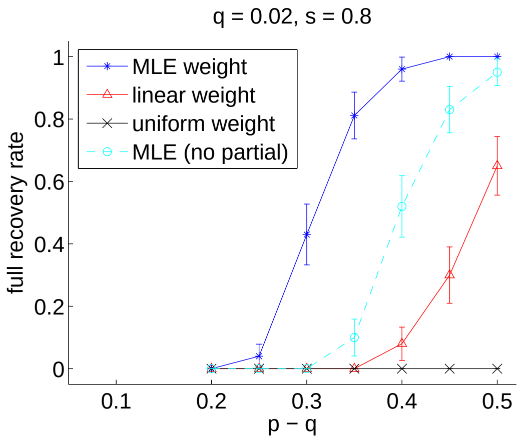


Figure 2: Clustering sparse graph ($n = 200$)

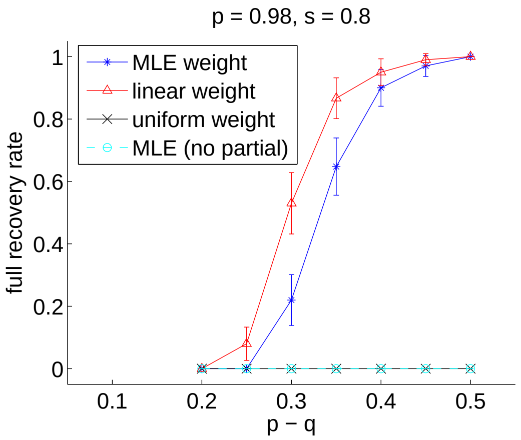
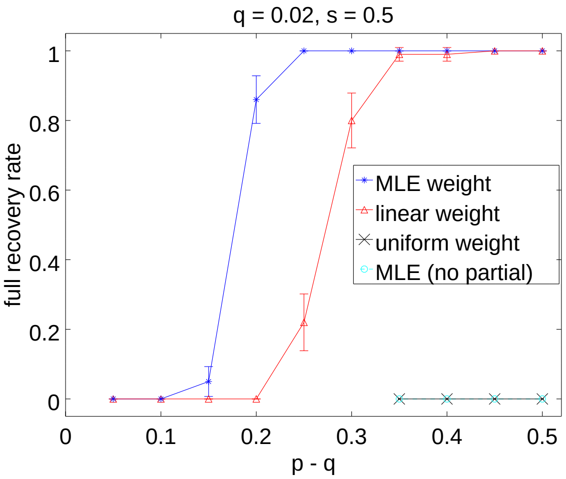


Figure 3: Clustering dense graph ($n = 200$)



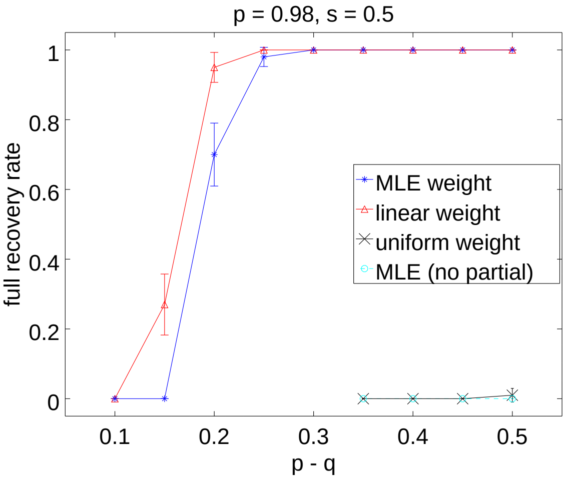


Figure 4: Clustering sparse graph ($n = 1000$)

Figure 5: Clustering dense graph ($n = 1000$)

Corollary 11 predicts more success as the ratio $\frac{s(p-q)^2}{p(1-q)}$ gets larger. All else being the same, label distributions with small ζ (corresponding to sparse graphs in the planted partition setting, where $\frac{D(\nu\|\mu)}{D(\mu\|\nu)}$ is small) are easier to solve. Note that these predictions are with respect to the MLE weights. Both predictions are consistent with the empirical results given in Figures 2–5. The results also indicate that the MLE weights outperform the other weights in the sparse settings. On the other hand, in the dense case, we observe that the linear weights outperform the MLE weights by a small margin. This empirical observation is consistent with the prediction given in Section 3.4.

To give an idea of the computation time involved, Figures 6 and 7 plot the average CPU time needed to solve program (2) with Algorithm 1 on a typical quad-core desktop machine in Matlab. A commonly observed trend is that the number of iterations needed to converge is usually small when the problem is either too “hard” ($p - q$ small) or too “easy” ($p - q$ large). Note that we did not attempt to optimize the algorithm in terms of speed. Improvement in the computational aspect is certainly an interesting direction to explore.

We next examine the effect of varying cluster size K on the performance, with the total number of nodes held fixed. Figures 8 and 9 show clustering results with various values of K for $n = 400$. We choose a particular value of (p, q, s) that shows an interesting region. As expected, the success rates improve when K grows. The results remain qualitatively similar for other values of p, q and s.

6.2.1 COMPARISON WITH SPECTRAL METHODS

Lelarge et al. (2013) proposed a labeled stochastic block model, which is a special case of our model (cf. Section 1.1). In particular, their setup is restricted to only the two-cluster case, with balanced cluster sizes (i.i.d. uniform cluster membership). They proposed a spectral method specially tailored for this case. In this section, we compare our approach with theirs on the exact same setup that they use. In this setup, each pairwise observation can be an edge/non-edge. Each edge can take one of two possible labels. This is equivalent to a 3-label case in our setup. Figure 10 shows the results in terms of the “overlap” (as used by Lelarge et al. (2013)), which is a measure of how closely the resulting clusters match the ground truth. In Fig. 10, “convex” refers to results

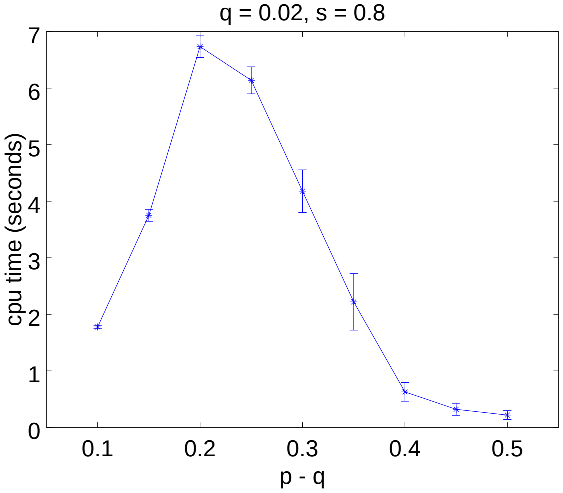


Figure 6: CPU time (sparse graph, $n = 200$)

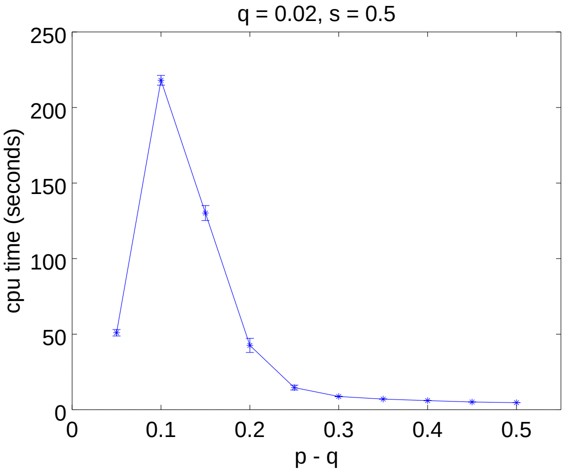


Figure 7: CPU time (sparse graph, $n = 1000$)

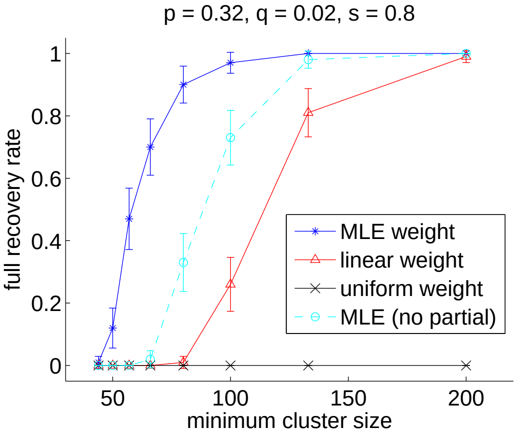


Figure 8: Clustering sparse graph ($n = 400$)

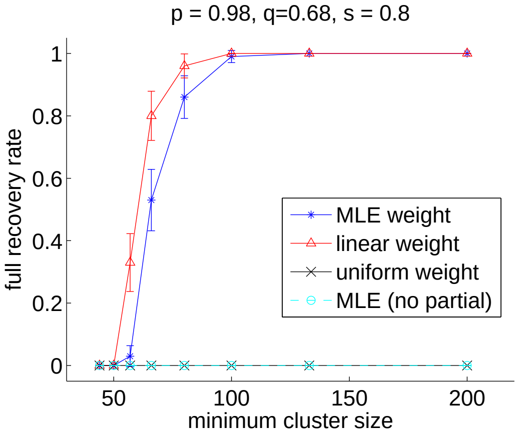


Figure 9: Clustering dense graph ($n = 400$)

based on Algorithm 1, where the final clustering is obtained by running K-means on the normalized rows of the output matrix. “Spectral” refers to results based on the proposed spectral method by Lelarge et al. (2013). We can observe that both approaches produce comparable results. Although computationally more costly,[13] our approach can handle a much wider range of problem setups, and as shown in Section 6.5.1, can significantly outperform the spectral method in many cases.

[13] We note that computationally the cost is typically dominated by the SVD operations, especially for large n. For spectral clustering, only 1 SVD needs to be performed, whereas the ADMM solver performs 1 SVD per iteration. The number of iterations may range from just a few to several hundreds, depending on the problem.

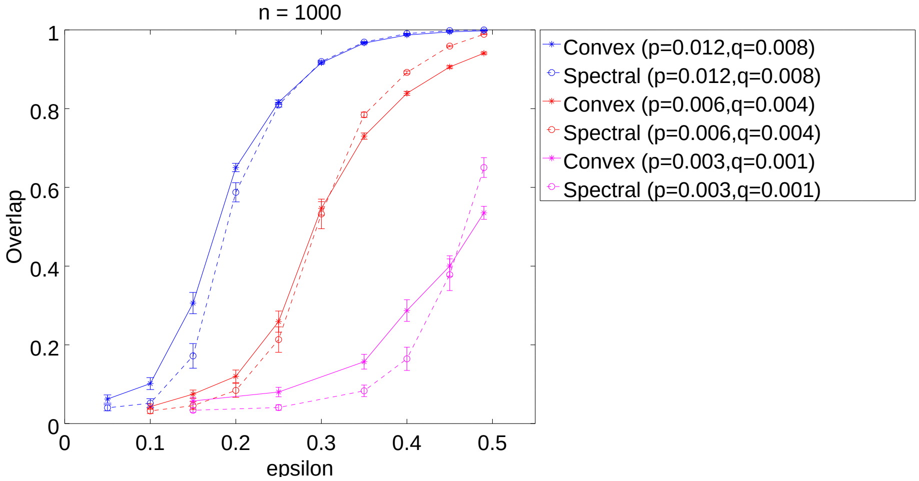


Figure 10: Comparison with spectral method. p and q are within-cluster and between-cluster edge probabilities, respectively. ϵ denotes the difference in within/between-cluster label distributions, where 0 is the smallest and 0.5 is the largest.

6.3 Gaussian Graphs and Inaccurate Weights

In this experiment, we evaluate real-valued labels drawn from Gaussian distributions. We also evaluate the effect of using weights that deviate from the MLE weights. Figure 11 shows clustering results on graphs with $n = 200$ and 8 equal-size clusters. The cross-cluster label distribution has mean 0 and variance 1. The in-cluster label distribution has the same variance but with elevated mean of $\mu = 2$ (blue) and $\mu = 1.5$ (red). Obviously, the case of $\mu = 2$ is an easier problem. In each case, we run Algorithm 1 using MLE weights that correspond to an in-cluster distribution with mean $\mu + \Delta$ instead of μ. Figure 11 shows that clustering performance drops when the deviation gets sufficiently large. Observe that negative Δ (conservative weights) performs better—consistent with Theorem 6. Figure 12 shows results based on the same settings, except that random noise is added to the (true) MLE weight, where the noise is uniformly distributed within the range $[-\Delta, \Delta]$. We observe a gradual drop of performance with respect to Δ, as predicted by Theorem 7.

6.4 Clustering with Highly Certain Observations

We empirically test the prediction of Theorem 9 regarding highly certain observations. For partially observed graphs with uncertain observations, the sufficient condition for recovery given by Corollary 13 requires that the fraction of observed entries, s, grows linearly with n (ignoring logarithmic term) if the other parameters p, q and K are held fixed. On the other hand, if all observed entries are highly certain, Theorem 9 predicts that s needs to grow only logarithmically with respect to n.

We run two experiments, each with $n = 600, 1200, 1800, 2400$. In all cases, the cluster size K is fixed at 150, and the labels are either “edge”, “no edge” or “unobserved”. In the first experiment we consider uncertain observations, where each observed entry is equal to the corresponding entry of Y^* with probability 0.7, and otherwise it is flipped. In the second experiment with highly certain

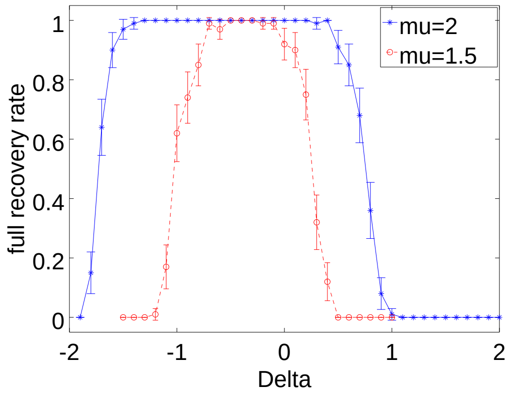


Figure 11: Systematic weight deviation

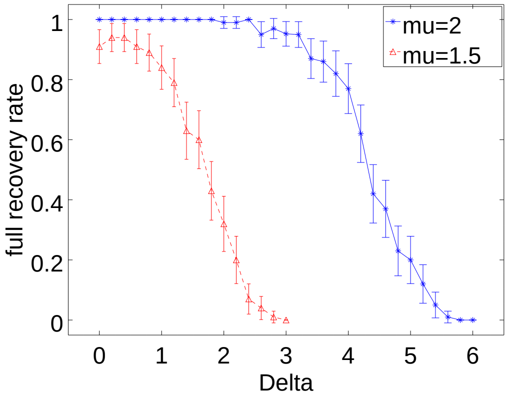


Figure 12: Random weight deviation

observations, all observed entries are equal to Y^* with probability 1. In both experiments, we search for the smallest fraction of observed entries needed to achieve a 90% full-recovery rate.

The results are shown in Figure 13. The plots indeed suggest a linear dependency on n for the uncertain observations, but a sub-linear dependency on n for the highly certain case.

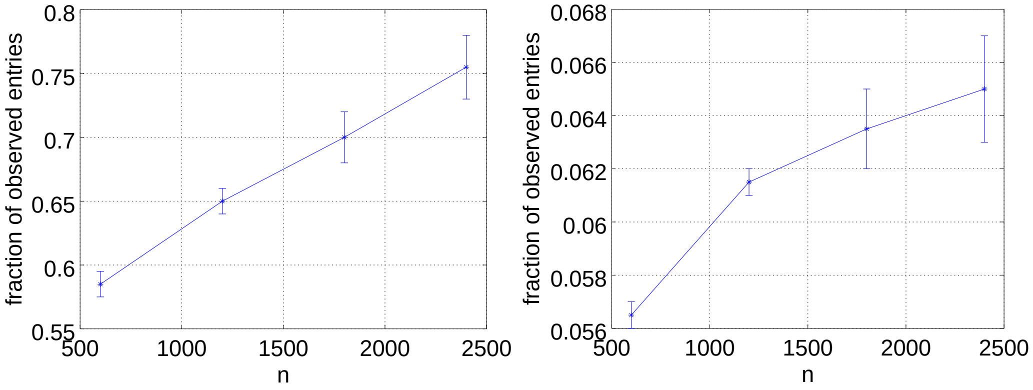


Figure 13: Fraction of observed entries needed for 90% full-recovery rate. Left panel: uncertain observations. Right panel: highly-certain observations.

6.5 Clustering Time-varying Graphs

We next investigate clustering performance on time-varying graphs. Figure 14 shows results based on multiple independent snapshots of partially observed graphs. Each graph is generated according to the planted partition model with partial observation (20% observed) as described in Section 4.2.1, with uniform error rate $q = 1 - p$ for all node pairs. We tested a wide range of error rates ($q \in [0.006, 0.325]$), and the horizontal axis tracks the corresponding KL divergence between μ and ν. We used $n = 200$ with 8 equal-size clusters. As predicted by Corollary 14, the clustering performance improves as the number of snapshots T grows.

Figure 15 shows results for Markov label sequences. Here, we test a simple model with two labels “interaction” and “no-interaction” (equivalent to “edge” and “no-edge”). For within-cluster pairs, the probability of interaction is greater (i.e., equal to $0.5 + \epsilon$) in the next time-step if there is no interaction in the current time-step, and vice versa. For inter-cluster pairs, the occurrence of interaction/no-interaction is completely random (i.e., $\epsilon = 0$) and therefore independent across snapshots. In this setting, both the marginal and stationary distributions for μ and ν are identical in every time step, and therefore at least two consecutive time steps are needed for informative clustering (cf. Section 4.4). The figure shows results for 200 nodes partitioned into 8 equal-size clusters, with the horizontal axis tracking the average KL-divergence between the conditional distributions (by varying ϵ). As predicted by Corollary 15, the performance improves as the number of snapshots T increases.

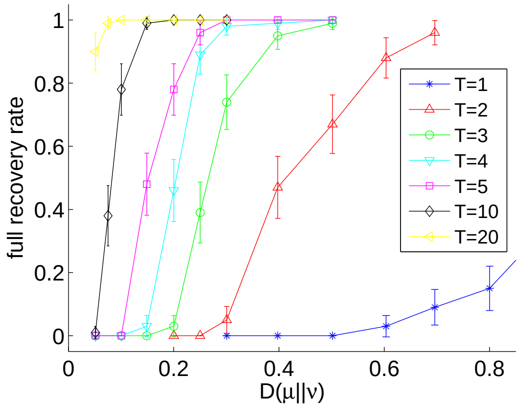


Figure 14: Independent snapshots

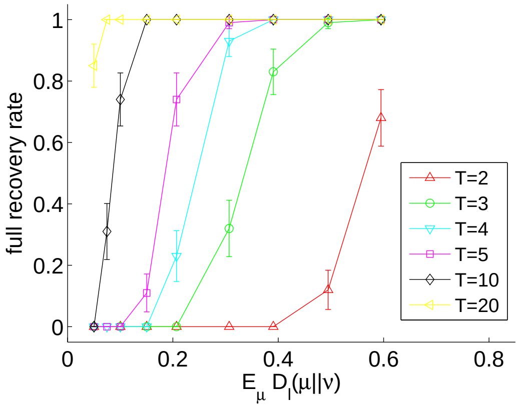


Figure 15: Markov snapshots

6.5.1 COMPARISON WITH EVOLUTIONARY SPECTRAL CLUSTERING

In this section, we compare our approach with an existing approach in clustering time-varying graphs based on evolutionary spectral clustering, as proposed by Chi et al. (2009). In particular, to handle time-varying graphs, one can assign a weight to each time step and compute a weighted average of the normalized graph Laplacian in each time step, then perform standard spectral clustering using this averaged graph Laplacian. We use the same settings as in Figure 14, except that we use the fully-observed case, to make sure that the comparison is fair.[14] For our approach, the final cluster assignments are obtained via running K-means on the normalized rows of the output matrix of the convex program. For spectral clustering, the final cluster assignments are based on K-means on the normalized rows of the top-K left singular vectors of the averaged graph Laplacian. For reference, we also include the results based on simply running K-means on the normalized rows of the averaged graph Laplacian. Figure 16 shows the results for $T = 1$, $T = 2$ and $T = 3$ in terms

[14] Our method has extra advantage in the partially observed case, since it is generally unclear what weight to use for spectral clustering in this case. We tested a few imputation scheme for spectral clustering but the results are qualitatively the same.

of full clusters recovery rate. Figure 17 shows the pairwise error rate, which is the fraction of all node pairs that have been wrongly classified as either belonging to the same or different clusters.

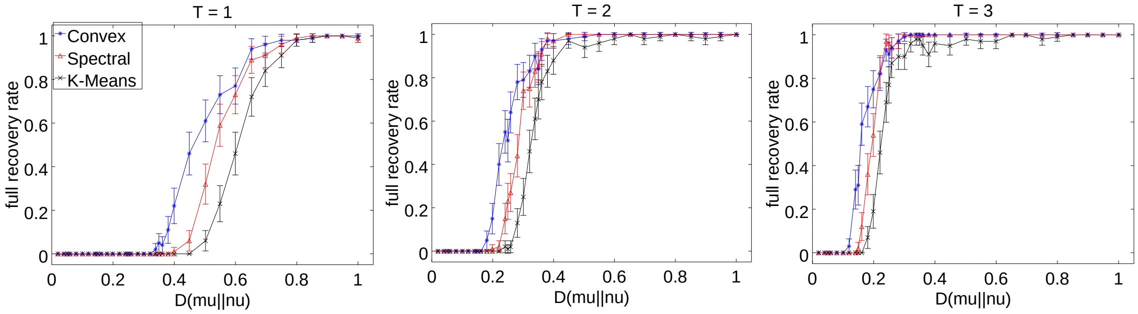


Figure 16: Comparison with evolutionary spectral clustering (full recovery rate)

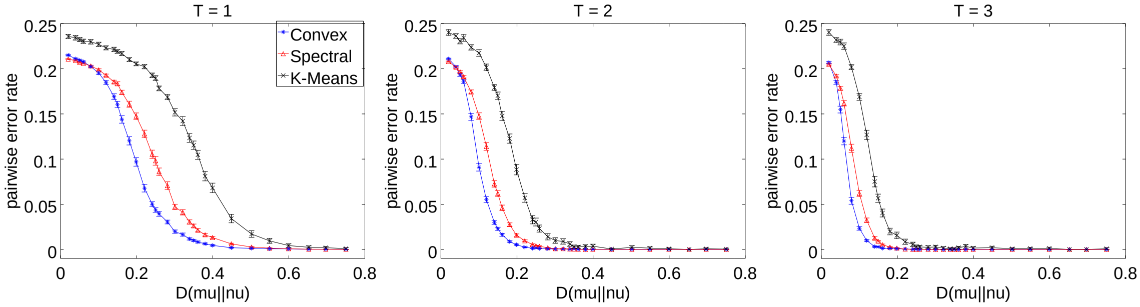


Figure 17: Comparison with evolutionary spectral clustering (pairwise error rate)

As expected, the overall performance improves as more snapshots are included. Among the three approaches, ours performs the best in terms of clustering error rates, but at the expense of higher computational cost relative to the other two.

6.6 Real-world Data Set

We consider three real-world data sets for which reliable ground truth is available. All three sets involve interactions among different social groups, with different clustering structures and different temporal granularity.

The *Reality Mining* data set (Eagle and Pentland, 2006) contains individuals from two main groups, the MIT Media Lab and the Sloan Business School, which we use as the ground-truth clusters. The data set records when two individuals interact, i.e., become proximal of each other or make a phone call, over a 9-month period. We choose a window of 14 weeks (the Fall semester) during which most individuals have non-empty interaction data. This sub-dataset consists of $n = 85$ individuals with 25 of them from Sloan and 60 from the Media Lab. We represent the data as a time-varying graph with 14 snapshots (one per week), each with two observations $\mathcal{L}_t = \{\text{“interact”}, \text{“no-interact”}\}$: “interact” if a pair of individuals interact within the week, and “no-interact” otherwise. Note that with T snapshorts, the full label set $\mathcal{L} = \mathcal{L}_1 \times \ldots \times \mathcal{L}_T$ consists of all possible binary sequences of length T.

We compare three models on this data set: the Markov snapshot model in Section 4.4, the independent snapshot model in Section 4.3, and a single snapshot approach (i.e., our approach with $T = 1$) applied to a static graph generated by taking the union of all snapshots (i.e., two individuals are connected if they interact during any of the 14 weeks). In each trial, the in/cross-cluster label distributions are estimated from a fraction of randomly selected pairwise interaction data. In particular, each parameter (i.e., the interaction probability and the transition kernel) is estimated using the corresponding empirical frequency in the selected data, regularized by adding 1 to each count. We use the MLE weights in all instances. To ensure that a valid clustering is always obtained, we run K-means on the normalized rows of the output matrix of Algorithm 1.

Figure 18 shows the results with respect to varying number of snapshots (left panel) and varying number of training pairs used for parameter estimation (right panel). For comparison, we also added result based on evolutionary spectral clustering (Chi et al., 2009). The vertical axis shows the fraction of pairs whose cluster relationships are correctly identified. The accuracy generally improves when more snapshots are used. However, the improvement with respect to T is lower than expected, most likely due to the fact that the snapshots are not independent. The union model is expected to improve with T for *sparse* graphs—due to increase in the KL-divergence between the in/cross-cluster distributions. Overall, the Markov model managed to achieve higher accuracy than both the union and independent model, though the best accuracy is achieved by the evolutionary spectral clustering approach. Further inspection of the data suggests that the snapshots are non-stationary in the sense that the earlier snapshots have rather different label distributions than the later snapshots.

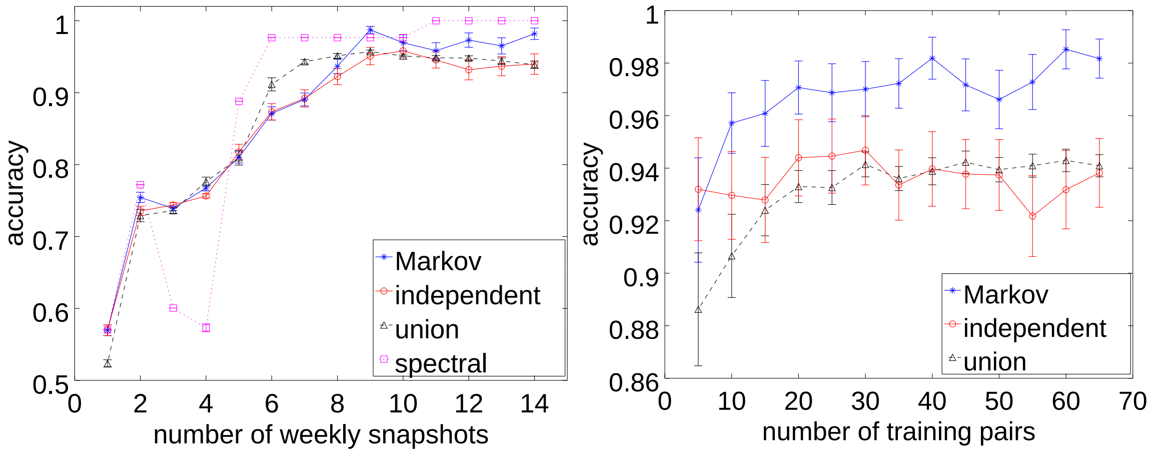


Figure 18: Reality Mining data set. Left: Varying number of snapshots (50 training pairs). Right: Varying number of training pairs (14 snapshots).

The next two data sets, *Workplace* and *Primary-school*, are from Genois et al. (2015) and Stehl et al. (2011), respectively. The Workplace data involves human contacts among 92 employees in an office building, where the ground truth clusters correspond to 5 different departments. We split this data set (over 10 days) into 10 daily snapshots, each with a binary {“interact”, “no-interact”} observation set. The Primary-school data involves contacts among 232 children in a primary school (teachers are omitted), where the ground truth clusters correspond to 10 different classes. We split this data set (over 2 days) into 18 hourly snapshots, with the same binary observation per snapshot.

Figures 19 and 20 show the empirical results using the Markov snapshot, independent snapshot and single snapshot models described above, as well as evolutionary spectral clustering. All 3 models achieve comparable clustering accuracy in the Workplace data set, while the union model performs significantly worse in the Primary-school data set. The evolutionary spectral clustering approach performs worse overall—with large variations across varying number of snapshots. We believe that this is due to its sensitivity to the distribution of the top eigenvectors of the graph Laplacian.

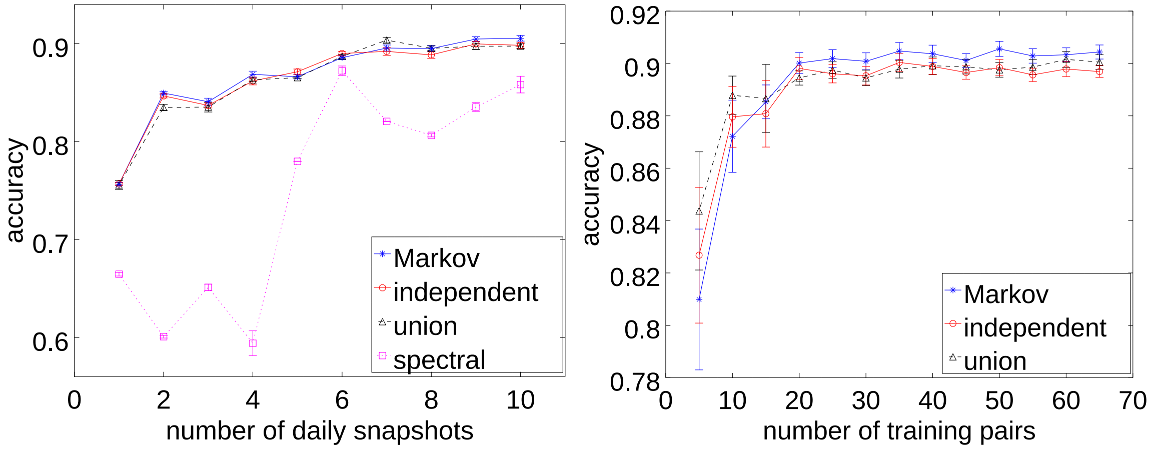


Figure 19: Workplace data set. Left: Varying number of snapshots (40 training pairs). Right: Varying number of training pairs (10 snapshots).

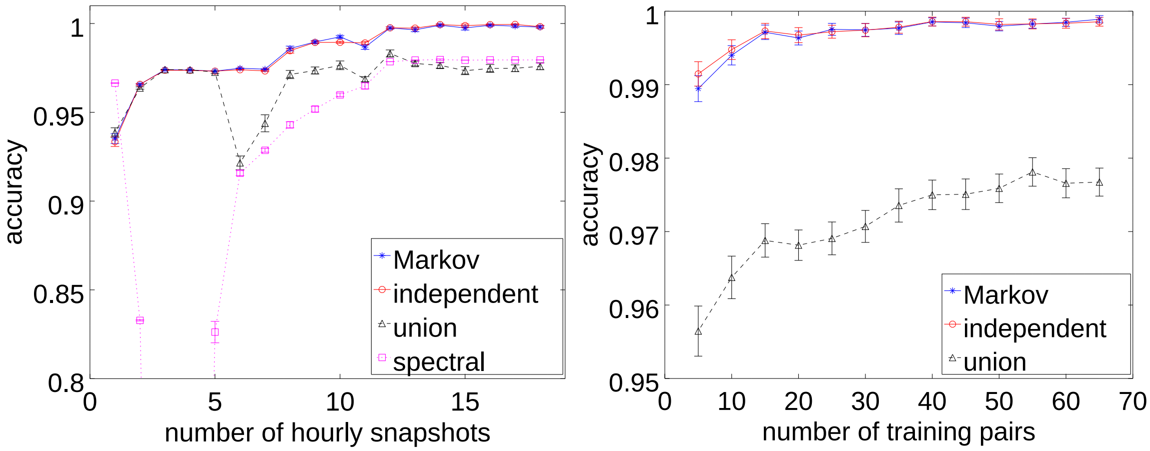


Figure 20: Primary-school data set. Left: Varying number of snapshots (50 training pairs). Right: Varying number of training pairs (18 snapshots).

7. Proof of Theorems 2 and 9

In this section, we prove the general results in the two main theorems. Observe that Theorem 2 is a special case of Theorem 9 with $\beta = 10$: if the weight function w is bounded, then there exists some b_∞ such that the set $\mathcal{L}_\infty$ is empty, in which case we have $\tilde{w}_b \equiv w$ and the conditions (14) and (15) in Theorem 9 reduce to (5) and (6) in Theorem 2. Therefore, it suffices to prove Theorem 9.

7.1 Notations

We need some additional notation to account for the bi-clustering setting. Recall that $|\mathcal{N}_1| = n_1$ and $\mathcal{N}_2 = n_2$, and set $n = \max\{n_1, n_2\}$. For a node $i \in \mathcal{N}_1$, let $\mathcal{C}_i$ be the cluster that contains node i and $K_i = |\mathcal{C}_i|$. Similarly, $\mathcal{C}'_j$ is the cluster that contains a node j in $\mathcal{N}_2$, and $K'_j = |\mathcal{C}'_j|$. Each cluster $\mathcal{C}_i$ in $\mathcal{N}_1$ is associated with exactly one cluster $\mathcal{C}'_j$ in $\mathcal{N}_2$, which is denoted by $\mathcal{C}_i \sim \mathcal{C}'_j$. These notations are consistent with those for clusteirng, with the understanding that $\mathcal{N}_1 = \mathcal{N}_2$, $n_1 = n_2 = n$, $\mathcal{C}_i = \mathcal{C}'_i$ and $K_i = K'_i$.

Let $\Omega = \{(i,j) : L_{ij} \in \mathcal{L}_\infty\}$ and $R = \{(i,j) : Y^*_{ij} = 1\}$. Let $\mathcal{P}_\Omega$ be the projection operator on matrices such that

$$(\mathcal{P}_\Omega Z)_{ij} = \begin{cases} Z_{ij} & \text{if } (i,j) \in \Omega, \\ 0 & \text{otherwise.} \end{cases}$$

The projections $\mathcal{P}_R$, $\mathcal{P}_{\Omega\cap R}$ etc are defined similarly.

Let $U\Sigma V^\top$ be the rank-r SVD of Y^*, where r is the number of groud-truth clusters and the rank of Y^*. Note that

$$(UU^\top)_{ij} = \begin{cases} \frac{1}{K_i} & \text{if } \mathcal{C}_i = \mathcal{C}_j, \\ 0 & \text{otherwise,} \end{cases} \tag{26}$$

$$(VV^\top)_{ij} = \begin{cases} \frac{1}{K'_i} & \text{if } \mathcal{C}'_i = \mathcal{C}'_j, \\ 0 & \text{otherwise,} \end{cases} \tag{27}$$

and

$$(UV^\top)_{ij} = \begin{cases} \frac{1}{\sqrt{K_i K'_j}} & \text{if } \mathcal{C}_i \sim \mathcal{C}'_j, \\ 0 & \text{otherwise.} \end{cases} \tag{28}$$

Define the projection operator

$$\mathcal{P}_T Z := UU^\top Z + ZVV^\top - UU^\top ZVV^\top \tag{29}$$

and its complementary projection

$$\mathcal{P}_{T^\perp} Z := Z - \mathcal{P}_T Z.$$

Denote by $\|\cdot\|$ the spectral norm (largest singular value) of a matrix. For any matrix X with $\|X\| \le \eta_0$, the matrix $UV^\top + \eta_0^{-1}\mathcal{P}_{T^\perp}X$ is a subgradient of the nuclear norm $\|\cdot\|_*$ at Y^* (Recht et al., 2010). It follows that for any feasible solution Y to the program (3), we have

$$0 \ge \|Y\|_* - \|Y^*\|_* \ge \langle UV^\top + \eta_0^{-1}\mathcal{P}_{T^\perp}X, Y - Y^*\rangle,$$

which implies

$$\langle X, Y^* - Y\rangle \ge \langle \mathcal{P}_T X - \eta_0 UV^\top, Y^* - Y\rangle. \tag{30}$$

7.2 Preliminary Lemmas

The proof of Theorem 9 builds on the following four lemmas. The first lemma gives a closed form expression for the projected matrix $\mathcal{P}_T Z$.

Lemma 16 *For any matrix Z and each index (i,j), we have*

$$(\mathcal{P}_T Z)_{ij} = \frac{1}{K_i}\sum_{k\in\mathcal{C}_i} Z_{kj} + \frac{1}{K'_j}\sum_{l\in\mathcal{C}'_j} Z_{il} - \frac{1}{K_i K'_j}\sum_{k\in\mathcal{C}_i}\sum_{l\in\mathcal{C}'_j} Z_{kl}. \tag{31}$$

Proof The lemma is immediate by the definition (29) of the projection $\mathcal{P}_T$ and the expressions (26) and (27) for the matrices $UU^\top$ and $VV^\top$. ■

Recall that W is the weight matrix used in program (3). The second lemma controls the spectral norm and the element magnitudes of the matrix $\mathcal{P}_\Omega UV^\top$.

Lemma 17 *Define*

$$\eta_1 := c_1 \frac{\beta \log n + \sqrt{K s_\mu (1-s_\mu)\beta \log n}}{K}.$$

With probability at least $1-n^{-\beta}$ the followings hold:

$$\|\mathcal{P}_\Omega UV^\top - \mathbb{E}[\mathcal{P}_\Omega UV^\top]\| \le \eta_1 \tag{32}$$

and for all i,j

$$|(\mathcal{P}_T(\mathcal{P}_\Omega UV^\top - \mathbb{E}[\mathcal{P}_\Omega UV^\top]))_{ij}| \le \begin{cases} \frac{\eta_1}{\sqrt{K_i K'_j}} & \textit{if } \mathcal{C}_i \sim \mathcal{C}'_j \\ 0 & \textit{otherwise} \end{cases} \tag{33}$$

Proof For the first inequality (32), consider $\mathcal{P}_\Omega UV^\top - \mathbb{E}[\mathcal{P}_\Omega UV^\top]$ as the sum of independent, zero-mean random matrices:

$$\mathcal{P}_\Omega UV^\top - \mathbb{E}[\mathcal{P}_\Omega UV^\top] = \sum_{(i,j),\mathcal{C}_i\sim\mathcal{C}'_j} X_{i,j},$$

where, noting the expression (28), we define

$$X_{i,j} := \mathcal{P}_\Omega\left(\frac{1}{\sqrt{K_i K'_j}}(e_i e_j^\top)\right) - \mathbb{E}\mathcal{P}_\Omega\left(\frac{1}{\sqrt{K_i K'_j}}(e_i e_j^\top)\right)$$

with e_i denoting the i-th standard basis vector. Note that

$$\|X_{i,j}\| \le \frac{1}{K}, \quad \forall i,j,$$

and

$$\left\|\sum_{(i,j),\mathcal{C}_i\sim\mathcal{C}'_j} \mathbb{E}X_{i,j}X_{i,j}^\top\right\| = \left\|\sum_{(i,j),\mathcal{C}_i\sim\mathcal{C}'_j} \frac{s_\mu(1-s_\mu)}{K_i K'_j}(e_i e_i^\top)\right\| \le \frac{s_\mu(1-s_\mu)}{K}.$$

Similarly we have $\left\|\sum_{(i,j),\mathcal{C}_i\sim\mathcal{C}'_j}\mathbb{E}X_{i,j}^\top X_{i,j}\right\| \leq \frac{s_\mu(1-s_\mu)}{K}$. By applying the matrix Bernstein inequality (Tropp, 2012), we obtain the desired inequality (32).

Turning to the inequality (33), let $Z := \mathcal{P}_\Omega UV^\top - \mathbb{E}[\mathcal{P}_\Omega UV^\top]$. Note that each entry of Z is an independent zero-mean random variable. From the expression (31), let $\hat{z}$ be the first term of its RHS, which is an average of K_i independent zero-mean random variables with $|Z_{kj}| \leq \frac{1}{\sqrt{K_iK'_j}}$ and

$$\mathrm{Var}(Z_{kj}) = \frac{s_\mu(1-s_\mu)}{K_iK'_j}.$$

By the standard Bernstein's inequality, we obtain that $|\hat{z}| \leq \frac{\eta_1}{\sqrt{K_iK'_j}}$ with probability at least $1-n^{-\beta-2}$. The same reasoning can be used to arrive at the same bound for the second and the third RHS term of expression (31). Applying a union bound over all (i,j) we obtain the desired inequality (33). ■

The next lemma is an analogue of Lemma 17 and controls the matrix $\mathcal{P}_{\Omega^c\cap R}W$.

Lemma 18 *Define*

$$\eta_2 := c_2\left(b\beta\log n + \sqrt{n\mathrm{Var}_\mu(\tilde{w}_b)\beta\log n}\right),$$
$$\eta_3 := c_3\left(b\xi\beta\log n + \sqrt{n\mathrm{Var}_\mu(\tilde{w}_b)\beta\log n}\right).$$

With probability at least $1-n^{-\beta}$ *the followings hold:*

$$\|\mathcal{P}_{\Omega^c\cap R}W - \mathbb{E}[\mathcal{P}_{\Omega^c\cap R}W]\| \leq \eta_2 \tag{34}$$

and for all i, j

$$|(\mathcal{P}_T(\mathcal{P}_{\Omega^c\cap R}W - \mathbb{E}[\mathcal{P}_{\Omega^c\cap R}W]))_{ij}| \leq \begin{cases} \min\left\{\frac{\eta_2}{K}, \frac{\eta_3}{\sqrt{K_iK'_j}}\right\} & \textit{if } \mathcal{C}_i\sim\mathcal{C}'_j, \\ 0 & \textit{otherwise}. \end{cases} \tag{35}$$

Proof This claim follows the same arguments as in the proof of Lemma 17. ■

The last lemma is again analogous to Lemma 17 and controls the matrix $\mathcal{P}_{\Omega^c\cap R^c}W$.

Lemma 19 *Define*

$$\eta_4 := c_4\left(b\beta\log n + \sqrt{n\mathrm{Var}_\nu(\tilde{w}_b)\beta\log n}\right),$$
$$\eta_5 := c_5\frac{b\beta\log n + \sqrt{K\mathrm{Var}_\nu(\tilde{w}_b)\beta\log n}}{K}.$$

With probability at least $1-n^{-\beta}$ *the followings hold:*

$$\|\mathcal{P}_{\Omega^c\cap R^c}W - \mathbb{E}[\mathcal{P}_{\Omega^c\cap R^c}W]\| \leq \eta_4 \tag{36}$$

and for all i, j

$$|(\mathcal{P}_T(\mathcal{P}_{\Omega^c\cap R^c}W - \mathbb{E}[\mathcal{P}_{\Omega^c\cap R^c}W]))_{ij}| \leq \begin{cases} 0 & \textit{if } \mathcal{C}_i\sim\mathcal{C}'_j, \\ \eta_5 & \textit{otherwise}, \end{cases} \tag{37}$$

Proof This claim follows the same arguments as in the proof of Lemma 17. ■

Remark 20 *The $\sqrt{\log n}$ factors in the expressions of η_2, η_3 and η_4 can be removed via a more careful analysis, for example by using the results in Bandeira and van Handel (2016). We do not delve into the details here.*

7.3 Proof of Theorem 9

We are now ready to complete the proof of Theorem 9. We will show that with probability at least $1 - n^{-\beta}$ the following inequality holds for all $Y \neq Y^*$ feasible to the program (3):

$$\langle W, Y^* - Y \rangle > 0,$$

which implies that Y^* is the unique solution of program (3). Note that the feasible set of the program (2) is a subset of the feasible set of the program (3) (but always contains Y^*). This means that whenever Y^* is the unique solution of the program (3), it is also the unique solution of the program (2).

To proceed, we decompose $\langle W, Y^* - Y \rangle$ as follows:

$$\begin{aligned}\langle W, Y^* - Y \rangle =& \langle \mathcal{P}_{\Omega \cap R} W, Y^* - Y \rangle + \langle \mathcal{P}_{\Omega \cap R^c} W, Y^* - Y \rangle + \langle \mathcal{P}_{\Omega^c \cap R} W, Y^* - Y \rangle \\ &+ \langle \mathcal{P}_{\Omega^c \cap R^c} W, Y^* - Y \rangle.\end{aligned} \tag{38}$$

We analyze separately the four terms on the RHS above.

For the first RHS term, note that by the assumption (12), all entries of the matrix $\mathcal{P}_{\Omega \cap R} W$ are positive with probability one. We therefore have

$$\begin{aligned}
&\langle \mathcal{P}_{\Omega \cap R} W, Y^* - Y \rangle \\
&\geq b_\infty K \langle \mathcal{P}_\Omega U V^\top, Y^* - Y \rangle \\
&= b_\infty K \left(\langle \mathbb{E}[\mathcal{P}_\Omega U V^\top], Y^* - Y \rangle + \langle \mathcal{P}_\Omega U V^\top - \mathbb{E}[\mathcal{P}_\Omega U V^\top], Y^* - Y \rangle \right) \\
&= b_\infty K \left(\langle s_\mu U V^\top, Y^* - Y \rangle + \langle \mathcal{P}_\Omega U V^\top - \mathbb{E}[\mathcal{P}_\Omega U V^\top], Y^* - Y \rangle \right) \\
&\overset{(a)}{\geq} b_\infty K \left(\langle s_\mu U V^\top, Y^* - Y \rangle + \langle \mathcal{P}_T(\mathcal{P}_\Omega U V^\top - \mathbb{E}[\mathcal{P}_\Omega U V^\top]) - \eta_1 U V^\top, Y^* - Y \rangle \right) \\
&\overset{(b)}{\geq} b_\infty K \left(\langle s_\mu U V^\top, Y^* - Y \rangle + \langle -2\eta_1 U V^\top, Y^* - Y \rangle \right) \\
&= b_\infty K (s_\mu - 2\eta_1) \langle U V^\top, Y^* - Y \rangle,
\end{aligned}$$

where we apply the bounds (30) and (32) in the step (a), and the bound (33) in the step (b). Again by assumption (12) we have $\langle \mathcal{P}_{\Omega \cap R} W, Y^* - Y \rangle \geq 0$. We conclude that

$$\langle \mathcal{P}_{\Omega \cap R} W, Y^* - Y \rangle \geq \max\{0, b_\infty K (s_\mu - 2\eta_1)\} \cdot \langle U V^\top, Y^* - Y \rangle. \tag{39}$$

For the second RHS term in (38), it follows immediately from the assumption (12) that

$$\langle \mathcal{P}_{\Omega \cap R^c} W, Y^* - Y \rangle \geq 0. \tag{40}$$

Turning to the third RHS term in (38), we have

$$\begin{aligned}
&\langle \mathcal{P}_{\Omega^c\cap R} W, Y^* - Y\rangle \\
=&\langle \mathbb{E}[\mathcal{P}_{\Omega^c\cap R} W], Y^* - Y\rangle + \langle \mathcal{P}_{\Omega^c\cap R} W - \mathbb{E}[\mathcal{P}_{\Omega^c\cap R} W], Y^* - Y\rangle \\
=&\langle (\mathbb{E}_\mu \tilde{w}_b) Y^*, Y^* - Y\rangle + \langle \mathcal{P}_{\Omega^c\cap R} W - \mathbb{E}[\mathcal{P}_{\Omega^c\cap R} W], Y^* - Y\rangle \\
\overset{(a)}{\geq}&\langle (\mathbb{E}_\mu \tilde{w}_b) Y^*, Y^* - Y\rangle + \langle \mathcal{P}_T(\mathcal{P}_{\Omega^c\cap R} W - \mathbb{E}[\mathcal{P}_{\Omega^c\cap R} W]) - \eta_2 UV^\top, Y^* - Y\rangle \\
\overset{(b)}{\geq}& \max\left\{ \left(\mathbb{E}_\mu \tilde{w}_b - \frac{2\eta_2}{K}\right) \langle Y^*, Y^* - Y\rangle, \left(K\mathbb{E}_\mu \tilde{w}_b - 2\eta_3\right) \langle UV^\top, Y^* - Y\rangle \right\}, \qquad (41)
\end{aligned}$$

where we use the bounds (30) and (34) in the step (a), and the bound (35) in the step (b); also in (b) we use the inequality $\langle (\mathbb{E}_\mu \tilde{w}_b) Y^*, Y^* - Y\rangle \geq \langle (K\mathbb{E}_\mu \tilde{w}_b) UV^\top, Y^* - Y\rangle$, which follows from the assumption (13).

Finally, letting J denote the all one matrix, we can bound the last RHS term in (38) as

$$\begin{aligned}
&\langle \mathcal{P}_{\Omega^c\cap R^c} W, Y^* - Y\rangle \\
=\ & \langle \mathbb{E}[\mathcal{P}_{\Omega^c\cap R^c} W], Y^* - Y\rangle + \langle \mathcal{P}_{\Omega^c\cap R^c} W - \mathbb{E}[\mathcal{P}_{\Omega^c\cap R^c} W], Y^* - Y\rangle \\
=\ & (-\mathbb{E}_\nu \tilde{w}_b)\langle J - Y^*, Y\rangle + \langle \mathcal{P}_{\Omega^c\cap R^c} W - \mathbb{E}[\mathcal{P}_{\Omega^c\cap R^c} W], Y^* - Y\rangle \\
\overset{(a)}{\geq}\ & (-\mathbb{E}_\nu \tilde{w}_b)\langle J - Y^*, Y\rangle + \langle \mathcal{P}_T(\mathcal{P}_{\Omega^c\cap R^c} W - \mathbb{E}[\mathcal{P}_{\Omega^c\cap R^c} W]) - \eta_4 UV^\top, Y^* - Y\rangle \\
\overset{(b)}{\geq}\ & (-\mathbb{E}_\nu \tilde{w}_b - \eta_5)\langle J - Y^*, Y\rangle - \langle \eta_4 UV^\top, Y^* - Y\rangle, \qquad (42)
\end{aligned}$$

where we use the bounds (30) and (36) in the step (a), and the bound (37) in the step (b).

Applying the above inequalities (39), (40), (41) and (42) to the equation (38), we obtain that

$$\begin{aligned}
\langle W, Y^* - Y\rangle \geq & \max\Bigg\{ \left(\mathbb{E}_\mu \tilde{w}_b - \frac{2\eta_2 + \eta_4}{K}\right) \langle Y^*, Y^* - Y\rangle, \\
& \qquad \left(b_\infty K(s_\mu - 2\eta_1) + K\mathbb{E}_\mu \tilde{w}_b - 2\eta_3 - \eta_4\right) \langle UV^\top, Y^* - Y\rangle \Bigg\} \\
& \qquad + (-\mathbb{E}_\nu \tilde{w}_b - \eta_5)\langle J - Y^*, Y\rangle \\
= & \max\Bigg\{ \left(\mathbb{E}_\mu \tilde{w}_b - \frac{2\eta_2 + \eta_4}{K}\right) \langle Y^*, Y^* - Y\rangle, \\
& \qquad b_\infty K \left(s_\mu + \frac{\mathbb{E}_\mu \tilde{w}_b}{b_\infty} - \left(2\eta_1 + \frac{2\eta_3 + \eta_4}{b_\infty K}\right)\right) \langle UV^\top, Y^* - Y\rangle \Bigg\} \\
& \qquad + (-\mathbb{E}_\nu \tilde{w}_b - \eta_5)\langle J - Y^*, Y\rangle. \qquad (43)
\end{aligned}$$

We see that the condition (14) in the theorem statement ensures that $-\mathbb{E}_\nu \tilde{w}_b - \eta_5 > 0$. On the other hand, the condition (15) ensures that

$$\mathbb{E}_\mu \tilde{w}_b - \frac{2\eta_2 + \eta_4}{K} > 0,$$

whereas the condition (16) ensures that

$$s_\mu + \frac{\mathbb{E}_\mu \tilde{w}_b}{b_\infty} - \left(2\eta_1 + \frac{2\eta_3 + \eta_4}{b_\infty K}\right) > 0.$$

Either of last two inequalities is sufficient to guarantee that the right hand side of the equation (43) is strictly positive. This completes the proof of Theorem 9.

8. Conclusion

In this paper we presented a general framework for graph clustering by assuming that all pairwise observations are in the form of labels. The algorithm involves solving a tractable convex optimization problem with an appropriately weighted objection function based on the observed labels.

A key contribution of our theoretical results is in showing that the MLE weights are order-wise optimal under a generalized version of the stochastic block model, thus providing a principled way of assigning weights to the observed graph. Our main results also identify the relevant parameters that are crucial to the successful recovery of the underlying clusters. These include the minimum cluster size, the distance between the label distributions, as well as properties of the observations such as the number of snapshots in a time-varying graph.

This framework recovers as special cases a broad range of existing results in graph clustering, and in fact provides substantial improvement for many of them. Important features such as partial observability and non-uniform uncertainties can be readily analyzed using our results, which provide new insights in many applications. Moreover, our framework is powerful enough to yield new results on novel settings such as the clustering of time-varying graphs.

An interesting future direction is to extend the proposed approach to problems with more complex structures such as overlapping clusters. Another important direction is to develop scalable solution applicable to very large data sets, both in terms of storage and computational complexity.

Acknowledgments

S.H. Lim and H. Xu were supported by the Ministry of Education of Singapore through AcRF Tier Two grant R-265-000-443-112. H. Xu was also supported by the School of Industrial and System Engineering of Georgia Institute of Technology. Y. Chen was supported by the National Science Foundation CRII award 1657420 and the School of Operations Research and Information Engineering at Cornell University.

Appendix A. Proofs of Theorem 3 and Corollary 4

In this section, we prove Theorem 3 and Corollary 4 for using MLE weights.

A.1 Proof of Theorem 3

Throughout this subsection, w always means the MLE weight function w^{MLE}. The proof is based on several lemmas. The first lemma, proved in Section A.1.1 to follow, bounds the log-likelihood ratios.

Lemma 21 *Suppose that* $\left|\log \frac{\mu(l)}{\nu(l)}\right| \leq b, \forall l \in \mathcal{L}$. *Then, for any* $l \in \mathcal{L}$,

$$\left|\log \frac{\mu(l)}{\nu(l)}\right| \leq (b+2)\left|\frac{\mu(l)-\nu(l)}{\mu(l)+\nu(l)}\right|.$$

The second lemma, proved in Section A.1.2 to follow, controls the variance terms.

Lemma 22 *Suppose that* $|\log\frac{\mu(l)}{\nu(l)}| \le b, \forall l \in \mathcal{L}$ *and* $D(\nu\|\mu) \le \zeta D(\mu\|\nu)$*, then*

$$\mathrm{Var}_\nu w \le 3(b+2)D(\nu\|\mu) \tag{44}$$

and

$$\max(\mathrm{Var}_\mu w, \mathrm{Var}_\nu w) \le (\zeta+1)(b+2)D(\mu\|\nu). \tag{45}$$

The last lemma is a classical result in information theory. The lemma bounds the KL divergence $D(\mu\|\nu)$ and $D(\nu\|\mu)$ in terms of the triangle discrimination between μ and ν.

Lemma 23 (Topsoe 2000) *The following holds for any distributions* μ *and* ν*, assuming that* μ *and* ν *are absolutely continuous with each other and with respect to a base measure* λ*:*

$$\min\{D(\mu\|\nu), D(\nu\|\mu)\} \ge \frac{1}{2}\int_{\mathcal{L}} \frac{(\mu(l)-\nu(l))^2}{\mu(l)+\nu(l)}\,\mathrm{d}\lambda.$$

We are now ready to prove Theorem 3. To this end, we verify that the conditions (5) and (6) in Theorem 2 are satisfied under the assumption of Theorem 3. Note that

$$-\mathbb{E}_\nu w = -\int_{\mathcal{L}} \log\frac{\mu(l)}{\nu(l)}\,\mathrm{d}\nu = D(\nu\|\mu).$$

Combining the assumption (7) in Theorem 3 with the previous bound (44), the first condition (5) in Theorem 2 is satisfied as follows:

$$\begin{aligned}\frac{b\log n}{K} + \frac{\sqrt{K\log n}\sqrt{\mathrm{Var}_\nu w}}{K} &\le \frac{(b+2)\log n}{K} + \sqrt{\frac{\log n}{K}}\sqrt{3(b+2)D(\nu\|\mu)} \\ &\le cD(\nu\|\mu) + c'\sqrt{D(\nu\|\mu)}\sqrt{D(\nu\|\mu)} \\ &\le c''D(\nu\|\mu) = -c''\mathbb{E}_\nu w.\end{aligned}$$

Turning to the condition (6), we note that $\mathbb{E}_\mu w = D(\mu\|\nu)$. Combining the assumption (8) in Theorem 3 with the previous bound (45) in a similar manner establishes the condition (6).

Finally, the last sentence in the statement of Theorem 3 is a special case of the following more general result, which is useful later in the proof of Theorem 5.

Lemma 24 *Suppose that* $\left|\log\frac{\mu(l)}{\nu(l)}\right| \le b, \forall l \in \mathcal{L}$*. Then we have*

$$D(\nu\|\mu) + D(\mu\|\nu) \le (b+2)\int_{\mathcal{L}} \frac{(\nu(l)-\mu(l))^2}{\nu(l)+\mu(l)}\,\mathrm{d}\lambda$$

and

$$\frac{1}{2b+3} \le \frac{D(\nu\|\mu)}{D(\mu\|\nu)} \le 2b+3.$$

Proof We have

$$\begin{aligned}\frac{D(\nu\|\mu)}{D(\mu\|\nu)}+1&=\frac{D(\nu\|\mu)+D(\mu\|\nu)}{D(\mu\|\nu)}\\&=\frac{\int_{\mathcal{L}}(\nu(l)-\mu(l))\log\frac{\nu(l)}{\mu(l)}\,\mathrm{d}\lambda}{D(\mu\|\nu)}\\&\overset{(a)}{\leq}\frac{(b+2)\int_{\mathcal{L}}\frac{(\nu(l)-\mu(l))^2}{\nu(l)+\mu(l)}\,\mathrm{d}\lambda}{D(\mu\|\nu)},\end{aligned}$$

where we use Lemma 21 in the inequality (a). This proves the first equation in the lemma. Bounding the right side of (a) using Lemma 23, we prove the upper bound in the second equation of the lemma. The lower bound in the second equation follows from switching the roles of μ and ν. ■

A.1.1 PROOF OF LEMMA 21

Consider the function $g(p)=\log\frac{1-p}{p}$ for $p\in[\frac{1}{e^b+1},0.5]$. By the convexity of $g(p)$ in this range we can linearly upper-bound it and show that $\log\frac{1-p}{p}\leq b\left(\frac{e^b+1}{e^b-1}\right)(1-2p)$. Taking $p=\frac{\nu}{\mu+\nu}$, we then have

$$\left|\log\frac{\mu}{\nu}\right|=\left|\log\frac{1-p}{p}\right|\leq b\left(\frac{e^b+1}{e^b-1}\right)|1-2p|=b\left(\frac{e^b+1}{e^b-1}\right)\left|\frac{\mu-\nu}{\mu+\nu}\right|\leq(b+2)\left|\frac{\mu-\nu}{\mu+\nu}\right|.$$

A.1.2 PROOF OF LEMMA 22

We need a version of the Padé approximation for logarithms:

$$\log\frac{1}{x}\geq\frac{(1-x)(5+x)}{2+4x},\quad\forall x>0,$$

which follows from that fact that the function $g(x)=\log\frac{1}{x}-\frac{(1-x)(5+x)}{2+4x}$ has a unique minimum $g(1)=0$. Using this inequality, we obtain that

$$\begin{aligned}3\int_{\mathcal{L}}\log\frac{\nu(l)}{\mu(l)}\,\mathrm{d}\nu-\int_{\mathcal{L}}\left|\frac{\mu(l)-\nu(l)}{\mu(l)+\nu(l)}\right|\left|\log\frac{\mu(l)}{\nu(l)}\right|\,\mathrm{d}\nu&=2\int_{\mathcal{L}}\frac{2\mu(l)+\nu(l)}{\mu(l)+\nu(l)}\log\frac{\nu(l)}{\mu(l)}\,\mathrm{d}\nu\\&\geq2\int_{\mathcal{L}}\frac{2\mu(l)+\nu(l)}{\mu(l)+\nu(l)}\frac{(1-\frac{\mu(l)}{\nu(l)})(5+\frac{\mu(l)}{\nu(l)})}{2+4\frac{\mu(l)}{\nu(l)}}\,\mathrm{d}\nu\\&=\int_{\mathcal{L}}\frac{(\nu(l)-\mu(l))(5\nu(l)+\mu(l))}{\mu(l)+\nu(l)}\,\mathrm{d}\lambda\\&=-4+8\int_{\mathcal{L}}\frac{\mu(l)^2}{\mu(l)+\nu(l)}\,\mathrm{d}\lambda\\&\overset{(a)}{\geq}0,\end{aligned}$$

where the last inequality (a) follows from the fact that $\int_{\mathcal{L}} \frac{\mu(l)^2}{\mu(l)+\nu(l)} \,\mathrm{d}\lambda \geq \frac{1}{2}$. The first inequality (44) in the lemma then follows from

$$\begin{aligned}
\mathrm{Var}_\nu w &\leq \mathbb{E}_\nu w^2 \\
&= \int_{\mathcal{L}} \left(\log \frac{\mu(l)}{\nu(l)} \right)^2 \mathrm{d}\nu \\
&\overset{(a)}{\leq} (b+2) \int_{\mathcal{L}} \left| \frac{\mu(l)-\nu(l)}{\mu(l)+\nu(l)} \right| \left| \log \frac{\mu(l)}{\nu(l)} \right| \mathrm{d}\nu \\
&\overset{(b)}{\leq} 3(b+2) \int_{\mathcal{L}} \log \frac{\nu(l)}{\mu(l)} \,\mathrm{d}\nu \\
&= 3(b+2) D(\nu \| \mu),
\end{aligned}$$

where the step (a) follows from Lemma 21 and the step (b) follows from the inequality above.

For the second inequality (45), again by using Lemma 21 we obtain that

$$\begin{aligned}
\max(\mathrm{Var}_\mu w, \mathrm{Var}_\nu w) &\leq \mathbb{E}_\mu w^2 + \mathbb{E}_\nu w^2 \\
&= \int_{\mathcal{L}} (\mu(l)+\nu(l)) \left(\log \frac{\mu(l)}{\nu(l)} \right)^2 \mathrm{d}\lambda \\
&\leq (b+2) \int_{\mathcal{L}} (\mu(l)+\nu(l)) \left| \frac{\mu(l)-\nu(l)}{\mu(l)+\nu(l)} \right| \left| \log \frac{\mu(l)}{\nu(l)} \right| \mathrm{d}\lambda \\
&= (b+2) \big(D(\mu\|\nu) + D(\nu\|\mu) \big) \\
&\leq (\zeta+1)(b+2) D(\mu\|\nu).
\end{aligned}$$

A.2 Proof of Corollary 4

The corollary follows immediately from Theorem 3, by lower bounding the left hand sides of the equations (7) and (8) using Lemma 23.

Appendix B. Proof of Theorem 5

In this section, we prove the converse result in Theorem 5. We use a standard technique of converting a statistical estimation problem to multiple hypothesis testing—in particular, we shall apply Theorem 2.5 of Tsybakov (2009). We will consider the standard clustering case where Y^* is symmetric. Set $M = n$ and let $\theta_0 \in \{0,1\}^{n\times n}$ be a fixed cluster matrix corresponding to two equal sized clusters. For $k = 1, \ldots, \frac{M}{2}$, let θ_k be the cluster matrix of a new clustering by swapping the 1st member of cluster 1 with the k-th member of cluster 2. Likewise, for $k = \frac{M}{2}+1, \ldots, M$, θ_k is obtained by swapping the 2nd member of cluster 1 with the k-th member of cluster 2. Let $L^0, L^1, \ldots, L^M$ be the random label matrices generated by the corresponding clustering.

Since the label of each pair (i, j) is generated independently, we have:

$$\begin{aligned}
D(L^j \| L^0) &= \sum_{i<j} D(L_{ij}^k \| L_{ij}^0) \\
&\overset{(a)}{=} (n-2)D(\mu\|\nu) + (n-2)D(\nu\|\mu) \\
&\overset{(b)}{\leq} (n-2)(b+2)\int_{\mathcal{L}} \frac{(\nu(l)-\mu(l))^2}{\nu(l)+\mu(l)}\,\mathrm{d}\lambda \\
&\overset{(c)}{\leq} (c'+2)c\log n;
\end{aligned}$$

here in step (a) we use the fact that due to the membership swap, exactly $n-2$ intra-cluster pairs in θ_0 become inter-cluster pairs in θ_j and vise-versa, step (b) follows from the first equation in Lemma 24, and step (c) holds due to the assumption (10) of the theorem and that b is bounded by a universal constant.

The result then follows from taking c sufficiently small and applying Theorem 2.5 of Tsybakov (2009).

Appendix C. Proof of Theorem 6

In this section we prove the monotonicity property in Theorem 6. Let $E = \{l \in \mathcal{L} : \bar{\mu}(l) \geq \bar{\nu}(l)\}$ and $E^c = \mathcal{L} \setminus E$. Since (μ, ν) is strictly more divergent than $(\bar{\mu}, \bar{\nu})$, we have that for all $l \in E$, $\mu(l) \geq \bar{\mu}(l) \geq \bar{\nu}(l) \geq \nu(l)$ and for all $l \in E^c$, $\nu(l) \geq \bar{\nu}(l) > \bar{\mu}(l) \geq \mu(l)$.

Suppose that the label matrix L is generated using the following two-stage procedure:

1. First, generate a matrix $\bar{L}$ from $(\bar{\mu}, \bar{\nu})$. Set $L \leftarrow \bar{L}$.

2. Second:

   - For each (i, j) where $Y_{ij}^* = 0$, if $L_{ij} \in E$, then with probability $1 - \frac{\nu(L_{ij})}{\bar{\nu}(L_{ij})}$, set $L_{ij} \leftarrow l$ where l is drawn from the set E^c with distribution $\frac{\nu(l)-\bar{\nu}(l)}{\int_{l' \in E^c} \nu(l')-\bar{\nu}(l')\,\mathrm{d}\lambda(l')}$. Let Ω_- be the set of all such entries, i.e. where L_{ij} has switched from E to E^c.

   - For each (i, j) where $Y_{ij}^* = 1$, if $L_{ij} \in E^c$, then with probability $1 - \frac{\mu(L_{ij})}{\bar{\mu}(L_{ij})}$, set $L_{ij} \leftarrow l$ where l is drawn from the set E with distribution $\frac{\mu(l)-\bar{\mu}(l)}{\int_{l' \in E} \mu(l')-\bar{\mu}(l')\,\mathrm{d}\lambda(l')}$. Let Ω_+ be the set of all such entries, i.e. where L_{ij} has switched from E^c to E.

It is straightforward to verify that the resulting distribution of L is identical to that generated by the pair (μ, ν).

Consider the program (3) with $\bar{L}$ as input, and let $\bar{W}$ be the corresponding MLE weights. Since the pair $(\bar{\mu}, \bar{\nu})$ satisfies the condition of Theorem 3, we have that with probability at least $1 - n^{-10}$, the matrix Y^* is the unique optimal solution and hence satisfies $\langle \bar{W}, Y^* \rangle > \langle \bar{W}, Y \rangle$ for any feasible solution $Y \neq Y^*$. Now, consider the program (3) with L as the input, using the corresponding MLE weights W based on $(\bar{\mu}, \bar{\nu})$. We have that for any feasible $Y \neq Y^*$,

$$\begin{aligned}
\langle W, Y^* \rangle - \langle \bar{W}, Y^* \rangle &= \sum_{(i,j)\in\Omega_+} (W_{ij} - \bar{W}_{ij}) Y^*_{ij} \\
&\geq \sum_{(i,j)\in\Omega_+} (W_{ij} - \bar{W}_{ij}) Y_{ij} \\
&\geq \sum_{(i,j)\in\Omega_-} (W_{ij} - \bar{W}_{ij}) Y_{ij} + \sum_{(i,j)\in\Omega_+} (W_{ij} - \bar{W}_{ij}) Y_{ij} \\
&= \langle W, Y \rangle - \langle \bar{W}, Y \rangle,
\end{aligned}$$

which implies that

$$\langle W, Y^* \rangle - \langle W, Y \rangle \geq \langle \bar{W}, Y^* \rangle - \langle \bar{W}, Y \rangle > 0.$$

Therefor, Y^* is still the unique optimal solution.

Appendix D. Proof of Theorem 7

In this section, we prove Theorem 7 for using inaccurate weights. Our strategy is to apply Theorem 2. To avoid confusion we use b' and c' to denote the constants b and c in Theorem 2.

To show that the condition (5) in Theorem 2 is satisfied, we upper bound its right hand side as follows:

$$\begin{aligned}
c' \frac{b' \log n + \sqrt{K \log n}\sqrt{\mathrm{Var}_\nu w}}{K} &\overset{(a)}{\leq} \frac{c' b' (1-\gamma)^2}{c \alpha^2 (b+2)} D(\nu \| \mu) + \sqrt{3 c'^2 \alpha^2 (b+2) D(\nu \| \mu) \frac{\log n}{K}} \\
&\leq \frac{c'(1-\gamma)^2}{c\alpha} D(\nu \| \mu) + \sqrt{\frac{3 c'^2 (1-\gamma)^2}{c} D(\nu \| \mu)^2} \\
&\overset{(b)}{\leq} (1-\gamma) D(\nu \| \mu) \\
&\overset{(c)}{\leq} D(\nu \| \mu) - \Delta_\nu \\
&= -\mathbb{E}_\nu w,
\end{aligned}$$

where in step (a) we use $\mathrm{Var}_\nu w \leq \alpha^2 \mathrm{Var}_\nu w^{\mathrm{MLE}}$ due to the condition $|w| \leq \alpha |\log \frac{\mu}{\nu}|$ and apply the bound (44), step (b) holds by choosing an appropriately large c, and in step (c) we use the assumption $|\Delta_\nu| \leq \gamma D(\nu \| \mu)$ in the statement Theorem 7.

We can use similar arguments, using the bound (45) instead of (44), to prove that the condition (6) in Theorem 2 is satisfied. Theorem 7 then follows from applying Theorem 2.

Appendix E. Proof of Corollary 10

In this section, we prove Corollary 10 for clustering Gaussian graphs. Ideally we would like apply Theorem 3 as we are using the MLE weight. A minor technical difficulty is that the boundedness condition $|w^{\mathrm{MLE}}(L_{ij})| \leq b$ is not satisfied as the Gaussian entries of L are unbounded. To overcome this we use a standard truncation argument.

Without loss of generality assume that $\underline{u} = 0$. Define a truncated version $\tilde{w}$ for the weight function w^{MLE} by

$$\tilde{w}(l) = \begin{cases} 0, & \text{if } |l| > c'\sqrt{\log n} \text{ or } l = ?, \\ l - (\bar{u} + \underline{u})/2, & \text{if } |l| < c'\sqrt{\log n}, \end{cases}$$

where $c' > 4c_0$ is a universal constant to be chosen later. With the goal of applying Theorem 2 to this weight function, we verify that the conditions of the theorem are satisfied. By the assumption $\bar{u} \leq \underline{u} + c_0\sqrt{\log n} = c_0\sqrt{\log n}$, the weight function $\tilde{w}$ satisfies $\tilde{w}(l) \leq b = (c' + c_0)\sqrt{\log n}, \forall l \in \mathcal{L}$, so the boundedness condition holds. We next verify the condition (6). Letting $\phi(x) = (2\pi)^{-1/2}e^{-x^2/2}$ be the density function of the standard normal, we note that

$$\begin{aligned} \mathbb{E}_\mu \tilde{w} = \int_{l:|l|<c'\sqrt{\log n}} w^{\text{MLE}}(l)\,\mathrm{d}\mu &= \int_{\mathcal{L}} w^{\text{MLE}}(l)\,\mathrm{d}\mu - \int_{l:|l|>c'\sqrt{\log n} \text{ or } l=?} w^{\text{MLE}}(l)\,\mathrm{d}\mu \\ &= \frac{s\bar{u}}{2} - s\int_{|l|>c'\sqrt{\log n}} (l - \bar{u}/2)\phi(l - \bar{u})\,\mathrm{d}l \\ &\geq \frac{s\bar{u}}{2} - s\int_{|l|>c'\sqrt{\log n}} (|l| + \bar{u}/2)\phi(l - \bar{u})\,\mathrm{d}l \\ &\geq \frac{s\bar{u}}{2} - 4s\underbrace{\int_{c'\sqrt{\log n}/2}^{+\infty} l\phi(l)\,\mathrm{d}l}_{T}, \end{aligned}$$

where the last inequality follows from the fact that $\bar{u} \leq c_0\sqrt{\log n} < \frac{c'\sqrt{\log n}}{4}$. We control the term T as

$$T = (2\pi)^{-\frac{1}{2}}\int_{c'\sqrt{\log n}/2}^{+\infty} xe^{-x^2/2}\,\mathrm{d}x \overset{(a)}{\leq} (2\pi)^{-\frac{1}{2}}\int_{c'\sqrt{\log n}/2}^{+\infty} e^{-(x-1/2)^2/2}\,\mathrm{d}x \overset{(b)}{\leq} \frac{1}{n^2} \overset{(c)}{\leq} \frac{\bar{u}}{16},$$

where the step (a) follows from $x \leq e^{x-1/2}$, (b) follows from the standard Gaussian tail bound $1 - \Phi(t) \leq e^{-t^2/2}$ and the fact that c' is sufficiently large, and (c) follows from the assumption (18) of Corollary 10. It follows that $\mathbb{E}_\mu \tilde{w} \geq \frac{s\bar{u}}{4}$. On the other hand, since $\tilde{w}$ is a truncated version of w^{MLE}, we have

$$\max\{\text{Var}_\mu \tilde{w}, \text{Var}_\nu \tilde{w}\} \leq \max\{\text{Var}_\mu w^{\text{MLE}}, \text{Var}_\nu w^{\text{MLE}}\} \leq s(\bar{u}^2 + 1) \leq 2c_0^2 s\log n.$$

Combining the above bounds, it is easy to check that the condition (6) of Theorem 2 is satisfied under the assumption (18) of Corollary 10. Similar lines of arguments establish the condition (5) of Theorem 2. This theorem therefore guarantees that the program (2) with the weight function $w = \tilde{w}$ recovers the true cluster matrix Y^* with probability at least $1 - n^{-10}$.

Now, by choosing the constant c' to be sufficiently large and using the Gaussian tail bound and the union bound, we are guaranteed that with probability at least $1 - n^{-10}$, $\tilde{w}(L_{ij}) = w^{\text{MLE}}(L_{ij})$ for all (i, j). In the intersection of the above two events (which occurs with probability at least $1 - 2n^{-10}$), the program (2) with the weight function $\tilde{w}$ is identical to that with the weight function w^{MLE}, both of which recover the true Y^*. The same holds for the program (3).

Appendix F. Proof of Corollary 15

In this section we prove Corollary 15 for clustering with Markov snapshots. The MLE weight in this case is given by

$$\begin{aligned} w^{\text{MLE}}(\bar{L}_{ij}) =& \frac{1}{T}\log\frac{\mu_0(L_{ij}^{(1)})\mu(L_{ij}^{(2)}|L_{ij}^{(1)})\dots\mu(L_{ij}^{(T)}|L_{ij}^{(T-1)})}{\nu_0(L_{ij}^{(1)})\nu(L_{ij}^{(2)}|L_{ij}^{(1)})\dots\nu(L_{ij}^{(T)}|L_{ij}^{(T-1)})} \\ =& \frac{1}{T}\log\frac{\mu_0(L_{ij}^{(1)})}{\nu_0(L_{ij}^{(1)})} + \frac{1}{T}\sum_{t=2}^{T}\log\frac{\mu(L_{ij}^{(t)}|L_{ij}^{(t-1)})}{\nu(L_{ij}^{(t)}|L_{ij}^{(t-1)})}. \end{aligned}$$

In the sequel, we will focus on an in-cluster pair (i,j) with label distribution μ and drop the subscript ij in L_{ij}. The same analysis holds for the cross-cluster pair (i,j).

It is convenient to consider an auxiliary Markov chain $X_1,\dots X_T$ where each state is characterized by a label pair $X_t = (L^{(t-1)}, L^{(t)})$ for $t>1$, and $X_1 = L^{(1)}$. We define the function f on the domain of X_t such that

$$f(L) = \log\frac{\mu_0(L)}{\nu_0(L)} \quad \text{and} \quad f(L,L') = \log\frac{\mu(L'|L)}{\nu(L'|L)}.$$

We therefore have

$$w^{\text{MLE}}(\bar{L}) = \frac{1}{T}\sum_{t=1}^{T} f(X_t).$$

It is straightforward to show that

$$\mathbb{E}_\mu(f(X_1)) = D(\mu_0\|\nu_0) \quad \text{and} \quad \mathbb{E}_\mu(f(X_t)) = \mathbb{E}_{\mu_0}D_l(\mu\|\nu) \ \text{ for } t>1,$$

whence

$$\mathbb{E}_\mu(w^{\text{MLE}}) = \frac{1}{T}D(\mu_0\|\nu_0) + \Big(1-\frac{1}{T}\Big)\mathbb{E}_{\mu_0}D_l(\mu\|\nu). \tag{46}$$

The rest of the proof concerns bounding $\text{Var}_\mu(w^{\text{MLE}})$. Following the proof of Lemma 22, the variance of $f(X_t)$ can be bounded by

$$\text{Var}_\mu(f(X_1)) \le 3(b+2)D(\mu_0\|\nu_0) \quad \text{and} \quad \text{Var}_\mu(f(X_t)) \le 3(b+2)\mathbb{E}_{\mu_0}D_l(\mu\|\nu) \quad (t>1).$$

We now bound the covariance $\text{Cov}_\mu(f(X_t), f(X_{t+\tau+1}))$ for $t\ge 2$ and $\tau \ge 0$:

$$\begin{aligned} &\text{Cov}_\mu(f(X_t), f(X_{t+\tau+1})) \\ =& \mathbb{E}_\mu\left[\log\frac{\mu(L^{(t)}|L^{(t-1)})}{\nu(L^{(t)}|L^{(t-1)})}\log\frac{\mu(L^{(t+\tau+1)}|L^{(t+\tau)})}{\nu(L^{(t+\tau+1)}|L^{(t+\tau)})}\right] - \mathbb{E}_{\mu_0}D_l(\mu\|\nu)^2 \\ =& \sum_{L^{(t-1)}}\mu_0(L^{(t-1)})\sum_{L^{(t)}}\mu(L^{(t)}|L^{(t-1)})\log\frac{\mu(L^{(t)}|L^{(t-1)})}{\nu(L^{(t)}|L^{(t-1)})}\sum_{L^{(t+\tau)}}\Pr_\mu(L^{(t+\tau)}|L^{(t)})D_{L^{(t+\tau)}}(\mu\|\nu) \\ &- \mathbb{E}_{\mu_0}D_l(\mu\|\nu)^2 \\ \overset{(a)}{\le}& \sum_{L^{(t-1)}}\mu_0(L^{(t-1)})\sum_{L^{(t)}}\mu(L^{(t)}|L^{(t-1)})\left|\log\frac{\mu(L^{(t)}|L^{(t-1)})}{\nu(L^{(t)}|L^{(t-1)})}\right|\sum_{L^{(t+\tau)}}\kappa\phi^\tau D_{L^{(t+\tau)}}(\mu\|\nu) \\ \le& \frac{\kappa\phi^\tau}{\min_l \mu_0(l)} b\mathbb{E}_{\mu_0}D_l(\mu\|\nu), \end{aligned}$$

where in the step (a) we use the geometric ergodicity assumption of μ, i.e.,

$$|\Pr_\mu(L^{(t+\tau)}|L^{(t)}) - \mu_0(L^{(t+\tau)})| \leq \kappa\phi^\tau.$$

The same bound also applies to the case $t = 1$. Note that the covariance bound is independent of t and only dependent on τ.

We proceed to bound $\mathrm{Var}_\mu(w^{\mathrm{MLE}})$ as follows:

$$\begin{aligned}
\mathrm{Var}_\mu(w^{\mathrm{MLE}}) &= \frac{1}{T^2}\sum_{t=1}^{T}\mathrm{Var}_\mu(f(X_t)) + \frac{2}{T^2}\sum_{t=1}^{T-1}\sum_{t'=t+1}^{T}\mathrm{Cov}_\mu(f(X_t), f(X_{t'})) \\
&\leq \frac{1}{T^2}\sum_{t=1}^{T}\mathrm{Var}_\mu(f(X_t)) + \frac{2}{T}\sum_{\tau=0}^{T-2}\frac{T-1-\tau}{T}\frac{\kappa\phi^\tau}{\min_l \mu_0(l)} b\mathbb{E}_{\mu_0}D_l(\mu\|\nu) \\
&\leq \frac{3(b+2)}{T}\left(\frac{1}{T}D(\mu_0\|\nu_0) + \frac{T-1}{T}\mathbb{E}_{\mu_0}D_l(\mu\|\nu)\right) + \frac{2\kappa}{(1-\phi)\min_l \mu_0(l)} b\frac{\mathbb{E}_{\mu_0}D_l(\mu\|\nu)}{T} \\
&\leq c\frac{(b+2)\Phi}{T}\left(\frac{1}{T}D(\mu_0\|\nu_0) + \frac{T-1}{T}\mathbb{E}_{\mu_0}D_l(\mu\|\nu)\right). \qquad (47)
\end{aligned}$$

With the above bounds (46) and (47), we complete the proof of Corollary 15 by applying Theorem 2.

Appendix G. Example of Markov Chain with Explicit Bound on Φ

The snapshots in the Markov model are not independent. Therefore, given T snapshots we do not expect a T-fold increase in the information as in the independent snapshot model. In the conditions given in Corollary 15, this penalty is characterized by the parameter Φ defined in Section 4.4. To provide a sense of what values it may take, we now derive an explicit bound for a simple class of 2-state sequences (i.e., $|\mathcal{L}| = 2$).

As in Section F, we first focus on an in-cluster pair (i, j) with label distribution μ, and drop the subscript ij in L_{ij}. Suppose the transition matrix for the distribution μ is

$$\begin{bmatrix} 1-p_0 & p_0 \\ p_1 & 1-p_1 \end{bmatrix},$$

where $0 < p_0, p_1 < 1$. If we identify the label set $\mathcal{L}$ with $\{\text{edge}, \text{non-edge}\}$, then p_0 can be thought of as the probability that an edge "flips" into a non-edge, and p_1 as the probability that a non-edge flips into an edge. Let $\rho := 1 - p_0 - p_1$. By eigen-decomposition we can show that the transition matrix after t transitions is

$$\begin{bmatrix} \frac{p_1}{p_0+p_1}\left(1 + \frac{p_0}{p_1}\rho^t\right) & \frac{p_0}{p_0+p_1}(1-\rho^t) \\ \frac{p_1}{p_0+p_1}\left(1-\rho^t\right) & \frac{p_0}{p_0+p_1}(1 + \frac{p_1}{p_0}\rho^t) \end{bmatrix},$$

and the stationary distribution on the two states, written as a vector, is given by

$$\mu_0 = \begin{bmatrix} \frac{p_1}{p_0+p_1} & \frac{p_0}{p_0+p_1} \end{bmatrix}.$$

This Markov chain of the observed sequence of an in-cluster pair satisfies the inequality

$$|\mathrm{Pr}_\mu(L^{(1+t)}|L^{(1)}) - \mu_0(L^{(1+t)})| \leq |\rho|^t$$

for all integer $t \geq 1$.

Now suppose that the cross-cluster distribution ν is of a similar form as μ, i.e., has the transition matrix

$$\begin{bmatrix} 1 - p_0' & p_0' \\ p_1' & 1 - p_1' \end{bmatrix},$$

for some p_0' and p_1'. By similar arguments, ν has the stationary distribution $\nu_0 = [p_1'/(p_0' + p_1'), p_0'/(p_0' + p_1')]$, and satisfies $|\mathrm{Pr}_\nu(L^{(1+t)}|L^{(1)}) - \nu_0(L^{(1+t)})| \leq |\rho'|^t$, where $\rho' := 1 - p_0' - p_1'$. Therefore, the geometric ergodicity condition in (23) of Section 4.4 holds with parameters $\kappa = 1$ and $\phi = \max\{|\rho|, |\rho'|\}$. The parameter Φ, having the value

$$\Phi = \frac{1}{\left(1 - \max\{|\rho|, |\rho'|\}\right) \min\left\{\frac{p_0}{p_0+p_1}, \frac{p_1}{p_0+p_1}, \frac{p_0'}{p_0'+p_1'}, \frac{p_1'}{p_0'+p_1'}\right\}}$$

is a constant independent of the parameters n, K, T etc.

The value of Φ determines how much new information is contained in a new snapshot. For example, suppose that $p_1 = p_0 \in (0, \frac{1}{2}]$ and $p_1' = p_0' \in (0, \frac{1}{2}]$, in which case the stationary distributions $\mu_0 = \nu_0 = [\frac{1}{2}\ \frac{1}{2}]$ are fixed, and hence

$$\Phi = \frac{1}{\min\{p_0, p_0'\}}.$$

The value of Φ is large when the flipping probabilities p_0, and p_0' are close to zero. In this case the next snapshot is almost always the same as the current snapshot, hence providing little extra information. On the other hand, if these probabilities are closer to $\frac{1}{2}$, say $p_0 = p_1 = \frac{1}{2}$ and $p_0' = p_1' = \frac{1}{4}$, then $\Phi = 4$ is small. In this case the snapshots have more independence and thus the next snapshot provides fresh information.

Also note that in the above case, the marginal distributions of in- and cross- cluster pairs are the same: $\mu_0 = \nu_0 = [\frac{1}{2}\ \frac{1}{2}]$. In this case, information about the clustering only comes from the "flipping" pattern in the snapshots. For example, if $p_0 = p_1 = \frac{1}{2}$ and $p_0' = p_1' = \frac{1}{4}$, then in a long sequence of snapshots, the fractions of "edge" and "non-edge" labels in a in-cluster pair and a cross-cluster pair are both close to 50%, but we will see more flippings on pairs in the same cluster.

References

Brendan P. W. Ames. Guaranteed clustering and biclustering via semidefinite programming. *Mathematical Programming*, 147(1–2):429–465, 2014. ISSN 0025-5610.

Brendan P. W. Ames and Stephen Vavasis. Nuclear norm minimization for the planted clique and biclique problems. *Mathematical Programming*, 129(1):69–89, 2011.

Anima Anandkumar, Rong Ge, Daniel Hsu, and Sham M. Kakade. A tensor spectral approach to learning mixed membership community models. *Journal of Machine Learning Research*, 15: 2239–2312, June 2014.

Sivaraman Balakrishnan, Mladen Kolar, Alessandro Rinaldo, Aarti Singh, and Larry Wasserman. Statistical and computational tradeoffs in biclustering. In *NIPS Workshop on Computational Trade-offs in Statistical Learning*, 2011.

Afonso S. Bandeira and Ramon van Handel. Sharp nonasymptotic bounds on the norm of random matrices with independent entries. *Annals of Probability*, 44(4):2479–2506, 2016.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1), 2004.

Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, January 2011.

Deepayan Chakrabarti, Ravi Kumar, and Andrew Tomkins. Evolutionary clustering. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 554–560. ACM, 2006.

Kamalika Chaudhuri, Fan Chung, and Alexander Tsiatas. Spectral clustering of graphs with general degrees in the extended planted partition model. In *Proceedings of the 25th Annual Conference on Learning Theory (COLT)*, pages 35.1–35.23, 2012.

Yudong Chen, Ali Jalali, Sujay Sanghavi, and Huan Xu. Clustering partially observed graphs via convex optimization. *Journal of Machine Learning Research*, 15:2213–2238, June 2014a.

Yudong Chen, Shiau Hong Lim, and Huan Xu. Weighted graph clustering with non-uniform uncertainties. In *International Conference on Machine Learning*, 2014b.

Yudong Chen, Sujay Sanghavi, and Huan Xu. Improved graph clustering. *IEEE Transactions on Information Theory*, 60(10):6440–6455, 2014c.

Yuxin Chen, Changho Suh, and Andrea J. Goldsmith. Information Recovery from Pairwise Measurements: A Shannon-Theoretic Approach. *ArXiv e-prints*, April 2015.

Yun Chi, Xiaodan Song, Dengyong Zhou, Koji Hino, and Belle L. Tseng. On evolutionary spectral clustering. *ACM Transactions on Knowledge Discovery from Data*, 3(4):17:1–17:30, December 2009.

Anne Condon and Richard M. Karp. Algorithms for graph partitioning on the planted partition model. *Random Structures and Algorithms*, 18(2):116–140, 2001.

Nathan Eagle and Alex (Sandy) Pentland. Reality mining: Sensing complex social systems. *Personal and Ubiquitous Computing*, 10(4):255–268, March 2006. ISSN 1617-4909.

Santo Fortunato. Community detection in graphs. *Physics Reports*, 486(3-5):75–174, 2010.

Wenjie Fu, Le Song, and Eric P. Xing. Dynamic mixed membership blockmodel for evolving networks. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 329–336, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-516-1.

Mathieu Genois, Christian L. Vestergaard, Julie Fournet, Andre Panisson, Isabelle Bonmarin, and Alain Barrat. Data on face-to-face contacts in an office building suggest a low-cost vaccination strategy based on community linkers. *Network Science*, 3:326–347, September 2015.

Ryan G. Gomes, Peter Welinder, Andreas Krause, and Pietro Perona. Crowdclustering. In *Advances in Neural Information Processing Systems*, 2011.

Qiuyi Han, Kevin S. Xu, and Edoardo M. Airoldi. Consistent estimation of dynamic and multi-layer block models. In *International Conference on Machine Learning*, 2015.

Simon Heimlicher, Marc Lelarge, and Laurent Massoulié. Community detection in the labelled stochastic block model. In *NIPS Workshop on Algorithmic and Statistical Approaches for Large Social Networks*, 2012.

Paul W. Holland, Kathryn B. Laskey, and Samuel Leinhardt. Stochastic blockmodels: Some first steps. *Social Networks*, 5:109–137, 1983.

Varun Jog and Po-Ling Loh. Information-theoretic bounds for exact recovery in weighted stochastic block models using the Renyi divergence. *ArXiv e-prints*, September 2015.

Vikas Kawadia and Sameet Sreenivasan. Sequential detection of temporal communities by estrangement confinement. *Scientific Reports*, 2, 2012.

Mladen Kolar, Sivaraman Balakrishnan, Alessandro Rinaldo, and Aarti Singh. Minimax localization of structural information in large noisy matrices. In *Advances in Neural Information Processing Systems*, pages 909–917, 2011.

Marc Lelarge, Laurent Massoulié, and Jiaming Xu. Reconstruction in the Labeled Stochastic Block Model. In *IEEE Information Theory Workshop*, Seville, Spain, September 2013. URL `http://hal.inria.fr/hal-00917425`.

Shiau Hong Lim, Yudong Chen, and Huan Xu. Clustering from labels and time-varying graphs. In *Advances in Neural Information Processing Systems*, pages 1188–1196, 2014.

Claire Mathieu and Warren Schudy. Correlation clustering with noisy input. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms*, page 712, 2010.

Catherine Matias and Vincent Miele. Statistical clustering of temporal networks through a dynamic stochastic block model. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 2016.

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

Nam P. Nguyen, Thang N. Dinh, Ying Xuan, and My T. Thai. Adaptive algorithms for detecting community structure in dynamic social networks. In *INFOCOM*, pages 2282–2290. IEEE, 2011.

Samet Oymak and Babak Hassibi. Finding dense clusters via low rank + sparse decomposition. arXiv:1104.5186v1, 2011.

Benjamin Recht, Maryam Fazel, and Pablo A. Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM Review*, 52(471), 2010.

Karl Rohe, Sourav Chatterjee, and Bin Yu. Spectral clustering and the high-dimensional stochastic block model. *Annals of Statistics*, 39:1878–1915, 2011.

Ohad Shamir and Naftali Tishby. Spectral clustering on a budget. In *AISTATS*, 2011.

Juliette Stehl, Nicolas Voirin, Alain Barrat, Ciro Cattuto, Lorenzo Isella, Jean-Franois Pinton, Marco Quaggiotto, Wouter Van den Broeck, Corinne Rgis, Bruno Lina, and Philippe Vanhems. High-resolution measurements of face-to-face contact patterns in a primary school. *PLOS ONE*, 6(8):e23176, 08 2011.

Jimeng Sun, Christos Faloutsos, Spiros Papadimitriou, and Philip S. Yu. Graphscope: parameter-free mining of large time-evolving graphs. In *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2007.

Flemming Topsoe. Some inequalities for information divergence and related measures of discrimination. *IEEE Transactions on Information Theory*, 46(4):1602–1609, Jul 2000.

Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of Computational Mathematics*, 12(4):389–434, 2012.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, 2009.

Ramya Korlakai Vinayak, Samet Oymak, and Babak Hassibi. Graph clustering with missing data: Convex algorithms and analysis. In *Advances in Neural Information Processing Systems*, pages 2996–3004, 2014.

Kevin S. Xu and Alfred O. Hero. Dynamic stochastic blockmodels for time-evolving social networks. *IEEE Journal of Selected Topics in Signal Processing*, 8(4):552–562, Aug 2014.

Jinfeng Yi, Rong Jin, Anil K. Jain, and Shaili Jain. Crowdclustering with sparse pairwise labels: A matrix completion approach. In *AAAI Workshop on Human Computation*, 2012.