A Convex Optimization Framework for Bi-Clustering

Shiau Hong Lim SHONGLIM@GMAIL.COM
National University of Singapore, 9 Engineering Drive 1, Singapore 117575

Yudong Chen YUDONG.CHEN@EECS.BERKELEY.EDU
University of California, Berkeley, CA 94720, USA

Huan Xu MPEXUH@NUS.EDU.SG
National University of Singapore, 9 Engineering Drive 1, Singapore 117575

Abstract

We present a framework for biclustering and clustering where the observations are general labels. Our approach is based on the maximum likelihood estimator and its convex relaxation, and generalizes recent works in graph clustering to the biclustering setting. In addition to standard biclustering setting where one seeks to discover clustering structure simultaneously in two domain sets, we show that the same algorithm can be as effective when clustering structure only occurs in one domain. This allows for an alternative approach to clustering that is more natural in some scenarios. We present theoretical results that provide sufficient conditions for the recovery of the true underlying clusters under a generalized stochastic block model. These are further validated by our empirical results on both synthetic and real data.

1. Introduction

In a regular clustering task, we look for clustering structure within a set $\mathcal{F}$ through observing the pairwise interactions between elements in this set. In biclustering, we instead have two sets $\mathcal{F}$ and $\mathcal{G}$ and the observed pairwise interactions are between object pairs (i, j) with $i \in \mathcal{F}$ and $j \in \mathcal{G}$. The aim is to discover clustering structure within $\mathcal{F}$, $\mathcal{G}$ or both through these observations. For example, in a recommender system, $\mathcal{F}$ consists of customers and $\mathcal{G}$ a set of products. In DNA microarray analysis, $\mathcal{F}$ could be biological samples and $\mathcal{G}$ a set of genes. The standard clustering task can be viewed as a special case of biclustering with $\mathcal{F} = \mathcal{G}$.

While many biclustering algorithms have been proposed and applied to a variety of problems, most are without formal guarantees in terms of the actual clustering performance. A typical approach begins with an objective function that measures the quality or cost of a candidate clustering, and then searches for one that optimizes the objective. Many interesting objective functions have intractable computational complexity and the focus of many past works have been on finding efficient approximate solutions to these problems.

We propose a tractable biclustering algorithm based on convex optimization. The algorithm can be viewed as a convex relaxation to the computationally intensive problem of finding a maximum likelihood solution under a generalized stochastic block model. The stochastic block model (Holland et al., 1983; Rohe et al., 2011) has been widely used in graph clustering. In this model, it is assumed that a true but unknown underlying clustering exists in both $\mathcal{F}$ and $\mathcal{G}$. A probabilistic generative model is defined for the observations and the performance of the clustering algorithm is evaluated in terms of the ability to recover the true underlying clusters.

Our main contribution is in extending the current advances in standard graph clustering to the biclustering setting. We provide the conditions under which our biclustering algorithm can recover the true clusters with high probability. Our theoretical results are consistent with existing results in graph clustering, which have been shown to be optimal in some cases.

One novel aspect of our result is in providing new insight on the case where clustering structure only occurs in one domain set, say $\mathcal{F}$ but not necessarily in $\mathcal{G}$. We show that under reasonable assumptions, the same algorithm can be used to recover the clusters in $\mathcal{F}$ regardless of the clustering structure in $\mathcal{G}$. This provides an alternative approach to

Proceedings of the 32^{nd} International Conference on Machine Learning, Lille, France, 2015. JMLR: W&CP volume 37. Copyright 2015 by the author(s).

standard graph clustering, where instead of relying on pairwise interactions within $\mathcal{F}$, we cluster objects in $\mathcal{F}$ through their interactions with elements of $\mathcal{G}$.

We employ the observation model as proposed in (Lim et al., 2014) where each observation is a label $\Lambda_{ij} \in \mathcal{L}$. The label set $\mathcal{L}$ can be very general. In standard graph clustering, $\mathcal{L}$ would consist of two labels "edge" and "no-edge". An additional "unknown" label may be included for partially observed graphs. We refer the reader to (Lim et al., 2014) for examples of other, more complex label sets, which include observations from time-varying graphs.

The paper is organized as follows. After discussing related works in the next section, we present the formal setup of our approach in Section 3. The main algorithm and its theoretical results are presented in Section 4. An implementation of the main algorithm is provided in Section 5. Empirical results on both synthetic and real-world data are presented in Section 6. The proofs of the theoretical results are given in the supplementary materials.

2. Related Work

A comparative study of many biclustering algorithms in the domain of analyzing gene expression data is provided in (Eren et al., 2012), and a comprehensive survey can be found in (Tanay et al., 2005). Beginning with the work of Hartigan (1972) and Cheng & Church (2000), many approaches to biclustering are based on optimizing certain combinatorial objective functions, which are typically NP-Hard, and heuristic and approximate algorithms have been developed but with no formal performance guarantees. One notable exception that is related to our settings with labels, is the work of Wulff et al. (2013). They proposed a very intuitive monochromatic cost function, proved its NP-hardness and developed a polynomial-time approximate algorithm. Another related approach is correlation clustering (Bansal et al., 2004), which was originally developed for clustering but can be extended to biclustering; results on computational complexity and approximate algorithms can be found in, e.g., (Demaine et al., 2005; Swamy, 2004; Puleo & Milenkovic, 2014). A popular approach to clustering and biclustering is spectral clustering and its variants (Chaudhuri et al., 2012; Rohe et al., 2011; Anandkumar et al., 2013; Kannan et al., 2000; Shamir & Tishby, 2011; Lelarge et al., 2013; McSherry, 2001).

Here we focus on average-case performance under a probabilistic generative model for generalized graphs with labels, and our algorithms are inspired by recent convex optimization approaches to graph clustering (Mathieu & Schudy, 2010; Ames & Vavasis, 2011; Lim et al., 2014; Chen et al., 2012; 2014; Cai & Li, 2014; Vinayak et al., 2014). For biclustering, the work by Ames (2013); Kolar et al. (2011) considers the setting with a "block-diagonal" structure (in the matrix B to be defined below). The recent work by Xu et al. (2014) studies a more general setting with "block-constant" structure. Both these settings are special cases of ours with 3 labels (1, -1 and "unobserved") and with clusters in both the rows and columns.

3. Problem Setup

A *bicluster* is defined as a cluster-pair (C, D) with $C \subseteq \mathcal{F}$ and $D \subseteq \mathcal{G}$. We say that (i, j) is a member-pair of (C, D) if $i \in C$ and $j \in D$. The key property that is shared by member-pairs of the same bicluster is the label distribution. Ideally, if two pairs (i, j) and (i', j') belong to the same bicluster, then their respective labels Λ_{ij} and $\Lambda_{i'j'}$ should have similar distributions.

Let $n_1 = |\mathcal{F}|$, $n_2 = |\mathcal{G}|$ and $n = \max\{n_1, n_2\}$. Without loss of generality, we use $i = 1 \ldots n_1$ to denote members of $\mathcal{F}$ and $j = 1 \ldots n_2$ to denote members of $\mathcal{G}$.

We assume an underlying clustering in $\mathcal{F}$ such that there exists a partition of $\mathcal{F}$ into r_1 disjoint subsets $\{C_p : p = 1 \ldots r_1\}$. Similarly, $\mathcal{G}$ is partitioned into $\{D_q : q = 1 \ldots r_2\}$. These partitionings result in a total of $r_1 \times r_2$ biclusters (C_p, D_q). Let $K_p = |C_p|$ and $L_q = |D_q|$ be the respective cluster sizes of C_p and D_q, with $K = \min_p K_p$ and $L = \min_q L_q$.

We group all the biclusters into two classes. This is specified using an $r_1 \times r_2$ matrix B whose entries $B_{pq} \in \{b_0, b_1\}$ where b_0 and b_1 $(b_0 < b_1)$ are two arbitrarily defined real numbers, identifying the class of each bicluster. Member-pairs of each bicluster share the same class. This is specified using an $n_1 \times n_2$ matrix Y^*, where $Y^*_{ij} = B_{pq}$ if $i \in C_p$ and $j \in D_q$.

Associated with each class is a set of label-generating distributions. We assume that each observed label Λ_{ij} is generated independently from some distribution μ_{ij}. If $Y^*_{ij} = b_1$ then $\mu_{ij} \in \mathcal{M}$, otherwise if $Y^*_{ij} = b_0$ then $\mu_{ij} \in \mathcal{N}$. In the simplest case, there are only two label distributions μ and ν such that $\mathcal{M} = \{\mu\}$ and $\mathcal{N} = \{\nu\}$. In this case $\Lambda_{ij} \sim \mu$ if $Y^* = b_1$ and $\Lambda_{ij} \sim \nu$ if $Y^* = b_0$.

Let U_C be an $n_1 \times r_1$ matrix denoting the membership of each cluster C_p, where $(U_C)_{ip} = 1/\sqrt{K_p}$ if $i \in C_p$, otherwise $(U_C)_{ip} = 0$. Note that each row of U_C contains only one non-zero entry. Similarly, V_D is an $n_2 \times r_2$ membership matrix for clusters D_q. Let $\mathcal{K} = \text{diag}(\sqrt{K_1} \ldots \sqrt{K_{r_1}})$ be a $r_1 \times r_1$ diagonal matrix. Similarly let $\mathcal{L} = \text{diag}(\sqrt{L_1} \ldots \sqrt{L_{r_2}})$.

Y^* can therefore be related to B by:

$$Y^* = U_C \mathcal{K} B \mathcal{L} V_D^\top.$$

Let the reduced SVD of $\mathcal{K}B\mathcal{L}$ be $U_B S_B V_B^\top$. It is easy to see that a valid SVD of Y^* is given by $U = U_C U_B$, $V = V_D V_B$ and $S = S_B$ such that $Y^* = USV^\top$.

The difficulty of the biclustering task, especially when B is non-square, depends on the $r_1 \times r_2$ matrix $\mathcal{K}B\mathcal{L} = U_B S_B V_B^\top$. To capture this dependence, we introduce the notion of *coherence*, defined as

$$u_1 = \max_{p \in \{1 \ldots r_1\}} \|U_B U_B^\top e_p\|^2 = \max_{p \in \{1 \ldots r_1\}} \|U_B^p\|^2,$$

$$u_2 = \max_{q \in \{1 \ldots r_2\}} \|V_B V_B^\top e_q\|^2 = \max_{q \in \{1 \ldots r_2\}} \|V_B^q\|^2$$

where e_i is the i-th standard basis vector and we define $U_B^p = U_B^\top e_p$ and similarly $V_B^q = V_B^\top e_q$. Since both U_B and V_B have orthonormal columns, it is straightforward to see that $\frac{1}{r_1} \leq u_1 \leq 1$ and $\frac{1}{r_2} \leq u_2 \leq 1$. Our definition of coherence is based on similar notion in the literature of low-rank matrix estimation/approximation. In our biclustering setting, it characterizes how easy it is to infer the structure in the columns from that in the rows, and vice versa.

4. Algorithm and Main Results

The biclustering task is to find Y^* given the observations Λ. The algorithm consists of two steps. First, a weight function $w : \mathcal{L} \to \mathbb{R}$ is chosen to construct an $n_1 \times n_2$ weight matrix W, where $W_{ij} = w(\Lambda_{ij})$. The second step involves solving the following convex optimization problem for the $n_1 \times n_2$ matrix Y:

$$\begin{aligned} \max_{Y} \quad & \langle W, Y \rangle - \lambda \|Y\|_* \\ \text{s.t.} \quad & b_0 \leq Y_{ij} \leq b_1, \forall i,j \end{aligned} \tag{1}$$

where $\lambda = c\sqrt{n \log n}$ for some sufficiently large constant c and $\|Y\|_*$ denotes the nuclear norm of Y.

Our main theorem provides the sufficient condition for the exact recovery of the true clustering matrix Y^* using program (1) with high probability. The condition depends on the following population quantities:

$$E_1 := \min_{\mu \in \mathcal{M}} \mathbb{E}_\mu w, \qquad E_0 := \min_{\mu \in \mathcal{N}} \mathbb{E}_\mu[-w],$$

$$V := \max_{\mu \in \mathcal{M} \cup \mathcal{N}} \mathrm{Var}_\mu w,$$

where $\mathbb{E}_\mu$ and Var_μ denote the expectation and variance under the distribution μ. With this notation, we have the following:

Theorem 1. *Suppose that B is full rank and $|w(l)| \leq b, \forall l \in \mathcal{L}$. There exists a universal constant c such that if*

$$\min\{E_1, E_0\} > c\left(b\beta \log n + \sqrt{\beta V n \log n}\right) \max\left\{\frac{u_1}{K}, \frac{u_2}{L}\right\}$$

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

Note that the probability in Theorem 1 is with respect to the randomness of the observed labels, given a fixed B and Y^*.

Theorem 1 is general in the sense that it applies to any bounded weight function w and any choice of $b_0 < b_1$ [1]. The clustering structure in $\mathcal{F}$ and $\mathcal{G}$ is reflected by the dependence on u_1, u_2, K and L. The proof for Theorem 1 (given in the supplementary material) follows the idea as in the proof of Theorem 1 in (Lim et al., 2014), but with extra consideration for the above structural properties. For the case of standard clustering, where $\mathcal{F} = \mathcal{G}$, both B and Y^* are symmetric and we have $u_1 = u_2$ and $K = L$. In this case we recover almost all the results in (Lim et al., 2014).

It is important to note that any two clusters in $\mathcal{F}$ or $\mathcal{G}$ can only be distinguished if the corresponding rows or columns in B are unique. It is, however, unclear whether the full-rank requirement for B is strictly necessary for the purpose of recovering Y^*.

One sided clusters: For the case of biclustering, both K and L play a similar role in Theorem 1. If one uses the naive bound $u_1 \leq 1$ and $u_2 \leq 1$ then it suggests that sufficiently large clusters in *both* $\mathcal{F}$ and $\mathcal{G}$ are necessary for (1) to be successful. This renders the result particularly weak when, for example, L is small relative to K. Yet, the matrix B has the same low rank as long as K is large, regardless of L. In this case, one should still be able to recover the clusters in $\mathcal{F}$.

The following result shows that this is indeed the case, assuming that B has rather uniformly spread out ±1 entries:

Theorem 2. *Suppose that $r_1 \leq r_2$ and $r_1 \geq \frac{\beta \log r_1}{c}$ for some universal constant c. Let $\phi = \frac{K}{\max_p K_p}$, $\psi = \frac{L}{\max_q L_q}$ and b such that $|w(l)| \leq b, \forall l \in \mathcal{L}$. Suppose that B_{pq} for each (p,q) is independent ± 1 random variable with $\Pr(B_{pq} = +1) = \Pr(B_{pq} = -1) = \frac{1}{2}$. There exists a universal constant c' such that if*

$$\min\{E_1, E_0\} > \left(\frac{c' n_1}{\phi \psi^2 n_2}\right) \frac{b\beta \log n + \sqrt{\beta V n \log n}}{K}$$

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

Note that the probability in Theorem 2 is with respect to both the randomness in B as well as the observed labels, holding the clusterings in $\mathcal{F}$ and $\mathcal{G}$ fixed. The proof (again, given in the supplementary material) is via bounding $\max\{u_1/K, u_2/L\}$ by applying a bound on the singular values of B by Rudelson & Vershynin (2009).

Theorem 2 implies that the success of (1) is essentially independent of L when we are only interested in clustering

[1] For all practical purposes, the choice of either $(b_0, b_1) = (0, 1)$ or $(b_0, b_1) = (-1, 1)$ should suffice.

$\mathcal{F}$. In particular, we can think of each row in the observed matrix as the feature representation (with $\mathcal{G}$ the feature set) of the corresponding element in $\mathcal{F}$. As long as elements that belong to the same cluster in $\mathcal{F}$ have the same feature relationships, we can perform clustering through these features regardless of whether the features themselves show a clustering structure. This is illustrated in Figure 1. The example on the left shows clustering structure only in the rows (each column is unique) while the example on the right shows clustering structure in both the rows and the columns. According to Theorem 2 program (1) should be equally effective in both cases, with the same order-wise dependency on K.

It is interesting to note that in the unbalanced case where n_1 is fixed, the condition of success improves as n_2 grows, even when the cluster sizes K and L remain the same. This situation cannot occur in standard clustering, where the problem always gets harder as n grows if K stays the same. This phenomenon is not discussed in the previous work (Kolar et al., 2011; Ames, 2013) on biclustering.

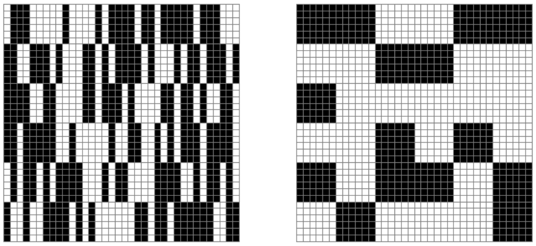


Figure 1. Two clustering/biclustering structures

4.1. Optimal Weights

We now consider the case where there are only two label distributions, i.e. $\mathcal{M} = \{\mu\}$ and $\mathcal{N} = \{\nu\}$.

Following previous works such as (Lim et al., 2014) and (Chen et al., 2012) we can view our algorithm as a convex relaxation of the maximum likelihood estimator, which is given by

$$\begin{aligned}
&\underset{Y \text{ a cluster matrix}}{\arg\max} \; \log \Pr(\Lambda | Y^* = Y) \\
= &\underset{Y \text{ a cluster matrix}}{\arg\max} \sum_{i,j} \log \mu(\Lambda_{ij})^{\frac{Y_{ij}-b_0}{b_1-b_0}} \nu(\Lambda_{ij})^{\frac{b_1-Y_{ij}}{b_1-b_0}} \\
= &\underset{Y \text{ a cluster matrix}}{\arg\max} \sum_{i,j} Y_{ij} \log \frac{\mu(\Lambda_{ij})}{\nu(\Lambda_{ij})}.
\end{aligned}$$

This therefore suggests the weight function

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

We have the following guarantee for the performance using this weight function.

Theorem 3. *Suppose $\mathcal{M} = \{\mu\}$ and $\mathcal{N} = \{\nu\}$. Suppose that B is full rank and $|\log \frac{\mu(l)}{\nu(l)}| \leq b, \forall l \in \mathcal{L}$. Let $\zeta = \max\{\frac{D(\mu||\nu)}{D(\nu||\mu)}, \frac{D(\nu||\mu)}{D(\mu||\nu)}\}$. There exists a universal constant c such that if*

$$\begin{aligned}
&\min\{D(\mu||\nu), D(\nu||\mu)\} \\
&> c\beta(\zeta+1)(b+2)(n \log n) \max\left\{\frac{u_1^2}{K^2}, \frac{u_2^2}{L^2}\right\}
\end{aligned}$$

or

$$\begin{aligned}
&\sum_{l \in \mathcal{L}} \frac{(\mu(l) - \nu(l))^2}{\mu(l) + \nu(l)} \\
&> c\beta(\zeta+1)(b+2)(n \log n) \max\left\{\frac{u_1^2}{K^2}, \frac{u_2^2}{L^2}\right\}
\end{aligned}$$

then with probability at least $1 - n^{-\beta}$ the solution of (1) with $w = W^{MLE}$ is unique and equals Y^* .

$D(\cdot\|\cdot)$ in the theorem denotes the KL-divergence.

The weight function w^{MLE} has been shown to be optimal up to a constant factor, at least for the case with two equal-size clusters (Lim et al., 2014).

4.2. Monotonicity

In the general case where we allow $\mathcal{M}$ and $\mathcal{N}$ to contain multiple distributions, it is unclear what weight function to use. The following property suggests a "conservative" weight function:

Theorem 4. *Suppose the weight function $w(l) = \log \frac{\bar{\mu}(l)}{\bar{\nu}(l)}, \forall l \in \mathcal{L}$ is used in program (1) where $\bar{\mu}$ and $\bar{\nu}$ are two distributions on $\mathcal{L}$ such that for any distributions $\mu \in \mathcal{M}$ and $\nu \in \mathcal{N}$, we have*

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

$\forall l \in \mathcal{L}$. If $\bar{\mu}$ and $\bar{\nu}$ satisfy the conditions of Theorem 3 then with probability at least $1 - n^{-\beta}$ the solution of (1) is unique and equals Y^*.

Theorem 4 says that if the distributions in $\mathcal{M}$ are label-wised well separated from the distributions in $\mathcal{N}$, at least as much as $\bar{\mu}$ from $\bar{\nu}$, then program (1) will perform no worse than if $\mathcal{M} = \{\bar{\mu}\}$ and $\mathcal{N} = \{\bar{\nu}\}$ with the associated w^{MLE} based on $\bar{\mu}$ and $\bar{\nu}$.

4.3. General Stochastic Block Model for Two Labels

We now discuss the results above by considering a special case of the general stochastic block model where there are only two labels, $\mathcal{L} = \{+, -\}$. Suppose that all member-pairs of the same bicluster (C_p, D_q) have the same label-distribution on $\mathcal{L}$. Let

$$\mu_{pq} = \Pr(\Lambda_{ij} = + | i \in C_p, j \in D_q), \forall p, q, i, j.$$

Suppose it is known (or estimated) that all μ_{pq} are in a set $\{\mu_1, \mu_2, \ldots, \mu_m\}$. Theorem 3 and 4 suggests a simple strategy to discover the clustering in $\mathcal{F}$ and $\mathcal{G}$:

1. Sort μ_i for $i = 1 \ldots m$ in ascending order and find the largest gap between any two consecutive μ_i.

2. Suppose the largest gap is between $\nu = \mu_{i_0}$ and $\mu = \mu_{i_1}$ where $\nu < \mu$, then set $\mathcal{N} = \{\mu_i : \mu_i \leq \nu\}$ and $\mathcal{M} = \{\mu_i : \mu_i \geq \mu\}$.

3. Solve program (1) with the weight function

$$w(+) = \log \frac{\mu}{\nu} \quad \text{and} \quad w(-) = \log \frac{1-\mu}{1-\nu}.$$

To illustrate, we use the example problem posed by Cai & Li (2014). This is a clustering problem ($\mathcal{F} = \mathcal{G}$) with 3 clusters, where the μ_{pq} for each block is given as follows:

$$\begin{bmatrix} 0.4 & 0.2 & 0.05 \\ 0.2 & 0.3 & 0.05 \\ 0.05 & 0.05 & 0.1 \end{bmatrix}$$

The traditional graph clustering approach where B is a diagonal matrix

$$\begin{bmatrix} b_1 & b_0 & b_0 \\ b_0 & b_1 & b_0 \\ b_0 & b_0 & b_1 \end{bmatrix}$$

could not solve this problem since the smallest diagonal μ_{pq} (0.1) is smaller than the largest off-diagonal μ_{pq} (0.2). Using our strategy one could come up with the following assignment where $\nu = 0.2$ and $\mu = 0.3$:

$$\begin{bmatrix} b_1 & b_0 & b_0 \\ b_0 & b_1 & b_0 \\ b_0 & b_0 & b_0 \end{bmatrix}$$

Suppose μ and ν satisfy the conditions in Theorem 3, then by Theorem 4 the 3 clusters can be identified with high probability since each row of B is unique.

4.4. General Multi-label Multi-distribution Case

In the general case where there are more than two labels, the strategy is to separate all possible distributions into two sets $\mathcal{M}$ and $\mathcal{N}$ such that they are as "far apart" as possible. The weight function can then be set with respect to a pair of distributions $\mu \in \mathcal{M}$ and $\nu \in \mathcal{N}$. Sometimes it may be preferable to merge two or more labels into one and use their marginal distributions instead. Ultimately, the performance of program (1) depends on E_0, E_1 and V in Theorem 1.

Program (1) may be run multiple times, each with a different weight function, to discover finer clustering or biclustering structure within the previous output. Full exploration of these possibilities is left to future work.

5. Implementation

Program (1) can be solved efficiently using the Alternating Direction Method of Multipliers (ADMM) (Boyd et al., 2011). We provide the pseudocode for a complete implementation in Algorithm 1, with explicit stopping condition.

The inputs and output of Algorithm 1 are the same terms used in program (1). We find that in practice, setting $\lambda = \sqrt{2n}$ works well. The threshold for convergence is specified by ϵ. We find that $\epsilon = 10^{-4}$ is a good tradeoff between the speed of convergence and the quality of the solution. All our experiment results, unless otherwise stated, were based on $\lambda = \sqrt{2n}$ and $\epsilon = 10^{-4}$.

An optional Step 7 for updating ρ may potentially improve the speed of convergence. This takes an additional parameter τ. If $\|X^{k+1} - Y^{k+1}\|_F > \tau\rho\|Y^{k+1} - Y^k\|_F$ then set $\rho := 2\rho$ and $Q^{k+1} := Q^{k+1}/2$. On the other hand, if $\tau\|X^{k+1} - Y^{k+1}\|_F < \rho\|Y^{k+1} - Y^k\|_F$ then set $\rho := \rho/2$ and $Q^{k+1} := 2Q^{k+1}$. Typically $\tau = 10$ is a stable choice. We refer the reader to Boyd et al. (2011) for further details.

Algorithm 1 ADMM solver for Program (1)

```
Input: $W \in \mathbb{R}^{n_1 \times n_2}$, $\lambda$, $b_0$, $b_1$, $\epsilon$
Output: $Y$
1. $\rho := 1, k := 0$
2. $Y^k := 0, Q^k := 0, (Y^k, Q^k \in \mathbb{R}^{n_1 \times n_2})$
3. $X^{k+1} := U \max\{\Sigma - \frac{\lambda}{\rho}, 0\} V^\top$ where $U \Sigma V^\top$ is an SVD of $(Y^k - Q^k)$.
4. $Y^{k+1} := \min\left\{\max\left\{X^{k+1} + Q^k + \frac{1}{\rho}W, b_0\right\}, b_1\right\}$
5. $Q^{k+1} := Q^k + X^{k+1} - Y^{k+1}$
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. $k := k + 1$, go to step 3.
```

In practice, due to finite precision and numerical errors, the output Y of Algorithm 1 will not have entries that are exactly b_0 or b_1, even if it is a successful recovery. A simple rounding, say, around $\frac{b_0+b_1}{2}$ would give the correct solution. The clusters in $\mathcal{F}$ (resp. $\mathcal{G}$) can then be obtained by sorting the rows (resp. columns) of Y. Even if the recovery is not 100%, a simple k-means algorithm can be applied to the rows/columns to obtain a desired number of clusters.

6. Experiments

6.1. Synthetic data

We first evaluate the performance of our algorithm on synthetic data, where the complete ground truth is available. We test a simple setup where there are only two labels and two label distributions. Three different sizes for n_1 (with $n_2 = n_1$) are used: 100, 500, and 1000. The number of clusters is 5 for $n_1 = 100$ and 10 for the others, all with equal sizes. The matrix B has independently generated ±1 entries with uniform probability. A noise level σ is defined such that each observed label Λ_{ij} equals Y^*_{ij} with probability $1 - \sigma$ and equals the opposite value with probability σ. This setup is very similar to that used in (Wulff et al., 2013), except that in our case, we report the actual "full recovery rate", i.e. the fraction of trials where the final output Y exactly equals Y^*, out of 100 independent trials. Error bars in all figures show 95% confidence intervals.

Figure 2 shows the results. The plot "conv" indicates results based on the output of Program (1) with each entry of Y rounded to the closest ±1. Since the noise is symmetric (for $+1$ and -1 observations), the weight for Program (1) can simply be set to $W_{ij} = \Lambda_{ij}$. Included in the figure are two additional plots "k-means" and "conv + k-means". The "k-means" results are based on running the k-means algorithm directly on the observed labels (since they are numeric), once on the rows and a separate run on the columns. After running k-means, all entries in the same bicluster ("block") are set to take the majority label in the block. The "conv + k-means" results are based on applying the same k-means process to the output of "conv". Note that Program (1) itself does not need to know the number of clusters while k-means requires the number of clusters as input. We choose k-means as reference since, despite its simplicity, it is still one of the top performing biclustering methods according to the report by Oghabian et al. (2014).

We can observe, from Fig. 2 that the simple k-means algorithm performs rather well but it is significantly outperformed by "conv" especially for larger n. We recommend running k-means on top of the output of program (1) especially when the number of clusters is known, since this produces the best recovery result. We can also validate both Theorem 1 and 3. The ratios $\frac{\sqrt{n}}{K}$ for the three sizes are 0.50, 0.45 and 0.32 respectively, and indeed the problem becomes "easier" as the ratio gets smaller.

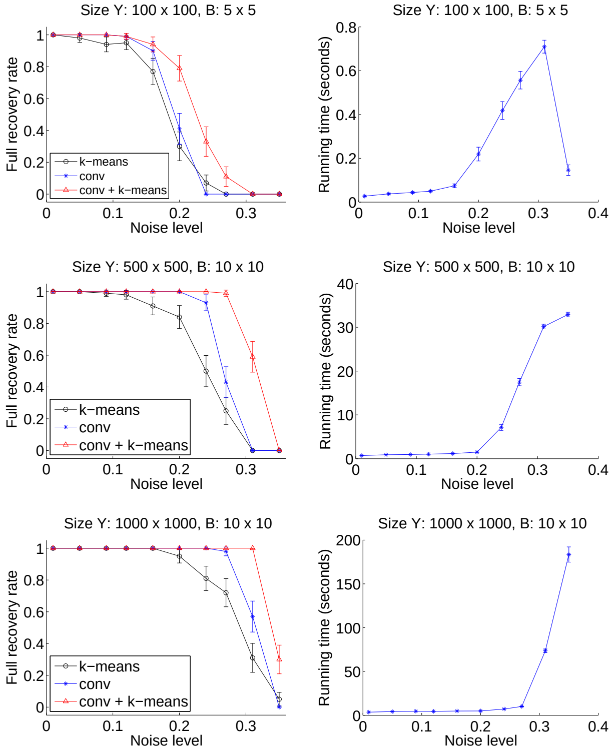


Figure 2. Full recovery rate and running time of program (1)

The average computational time needed to finish one instance of program (1) in Matlab on a Core i5 desktop machine is also reported in Fig. 2. It is interesting to note that convergence is typically very fast when the problem instance is either very "easy" (e.g. low noise) or too "hard", while the "borderline" instances usually require a much larger number of iterations to converge.

6.1.1. One-sided Biclustering

Our second set of experiments evaluates the prediction of Theorem 2. In particular, when large clusters exist in $\mathcal{F}$ (i.e. the rows), and B is more or less uniformly distributed, the recovery of Y^* depends mainly on K and not L. Figure 3 shows two sets of plots. Both experiments use $n_1 = 400$ and $n_2 = 800$. B has ±1 entries and the observations consist of two labels, 0 or 1. Entries with $Y^* = +1$ produce the label 1 with probability 0.7 while the rest produce the label 1 with probability 0.05. This is the case where the observation noise is non-symmetric.

In the left figure, B in each trial is generated with uniformly random ±1 entries. Equal-size clusters are generated in both the rows and the columns, where each row (resp. column) cluster has size K (resp. L). We test the biclustering performance in two cases: (1) fixed K and varying L, (2) varying K and L. Case 1 is shown in the blue plot while case 2 is in red. As predicted by Theorem 2, the full recovery rate remains high in case 1 where K remains fixed and large regardless of L. In case 2, the performance drops when both K and L become small, as expected.

In the right figure, we evaluate the biclustering performance in a scenario where the assumption of Theorem 2 is violated. In particular, we force B to be a 10×10 diagonal matrix – the number of clusters is fixed at 10 in both the rows and the columns. Note that in this case, u_1 and u_2 both equal 1. The first cluster in the rows (resp. columns) is chosen to have size equal to K (resp. L). The remaining clusters all have equal, but larger sizes. Here, we observe that the performance drops significantly in both cases, even though K is held large in the first case.

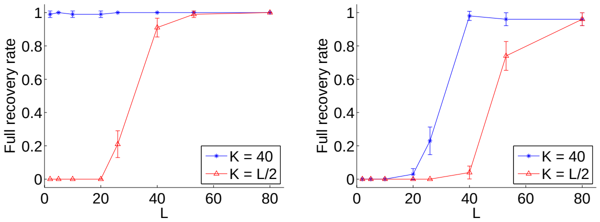


Figure 3. Full recovery rate for two different setups with varying cluster sizes. *Left*: B has uniform ± 1 entries. *Right*: B diagonal.

6.2. Real World Data

We further evaluate our algorithm on real world data, where the true data generating model is unknown.

6.2.1. Animal-Feature Dataset

Our first dataset is the animal-feature data originally published by Osherson et al. (1991) and has been used in (Wulff et al., 2013) and (Kemp et al., 2006) for biclustering. The dataset contains human-produced association between a set of 50 animals and 85 features. For each animal-feature pair, the degree of association–a real value between 0 and 100, is provided. The biclustering task is to simultaneously cluster the animals and features. Unfortunately no ground truth is available so evaluation is subjective. Nevertheless this allows a qualitative comparison with the results reported in, e.g. (Wulff et al., 2013) and (Kemp et al., 2006).

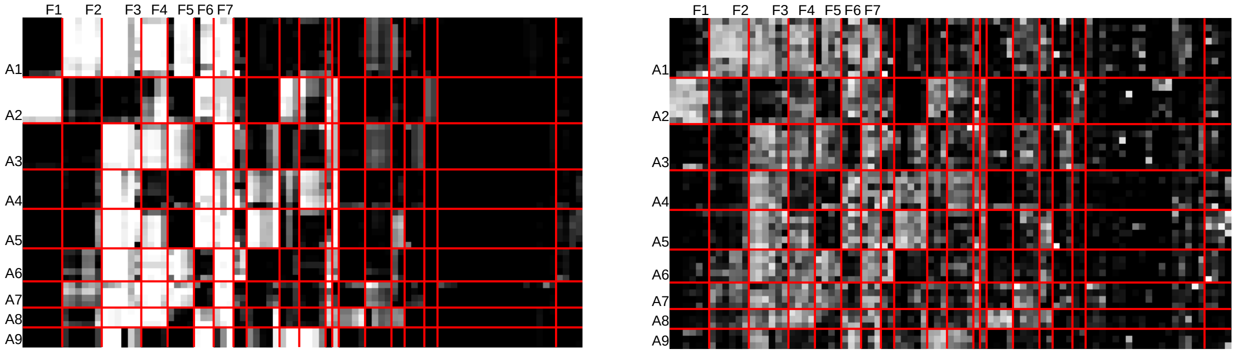
We use the raw data in two ways. First, we follow Kemp et al. (2006) and use the global mean value $t \approx 20.8$ as the cutoff point between a "yes" and "no". For the "yes" entries, we assume that the probability of error decreases linearly with the recorded degree of association, where we set 101 to be the point of 0 error probability while the probability of error at t is 0.5. Similarly for the "no" entries the point of 0-error is set at -1. We then employ the MLE weight and run program (1). To extract clusters (both the animals and the features) we run k-means on the rows and columns of the program output Y. Given that this is unlikely to be the true model, we expect a need to adjust the regularization parameter λ as well as the number of clusters. We tested a range of λ and cluster sizes, and choose a combination that produces clusters with the most uniform sizes. Figure 4 shows the result with $\lambda = \sqrt{n}$, for 9 animal clusters and 20 feature clusters. There is randomness involved in using k-means that depends on the initialization but for the chosen parameters the resulting clusters are more-or-less stable. We find the results reasonable and not unlike those obtained by Wulff et al. (2013) and Kemp et al. (2006).

6.2.2. Gene Expression Data

For our second real-world example, we use the gene expression data for leukemia, in the form provided by Monti et al. (2003). This is a filtered version of the original raw data published in (Golub et al., 1999), with mostly uninformative genes removed. These are DNA microarray data collected from bone marrow samples of acute leukemia patients, with 11 acute myeloid leukemia (AML) samples, 8 T-lineage acute lymphoblastic leukemia (ALL) samples, and 19 B-lineage ALL samples. The "features" are the recorded expression levels of 999 genes. The expression levels are real values with widely different range for each gene. We scale and shift the expression levels for each gene such that the mean is 0 and the standard deviation is 1. Since the actual model is unknown, we assume a Gaussian distribution where we use $\mathcal{N}(1,1)$ for μ and $\mathcal{N}(-1,1)$ for ν. The MLE weight is therefore $\log \frac{\mu(l)}{\nu(l)}$, where l is the observed expression level[2]. Note that the resulting weight is simply a linear function of l.

Since the type of each sample is known, we can evaluate the clustering performance for the 38 samples. Again, we ran k-means on the program output and tested a range of λ. Figure 5 shows the clustering performance with respect to λ, against the ground-truth clustering, in terms of the adjusted mutual information and the percent of sample-pairs that are correctly clustered together. The two performance measures give qualitatively similar results and the best performance is achieved around $\lambda = \sqrt{5n}$. In general, a small λ would produce a finer clustering structure while a larger λ produce a coarser structure. Note that using $\lambda = 0$ is equivalent to simply thresholding the raw value to the nearest ± 1 (right panel of Fig. 6). Running k-means directly on the raw data (middle panel of Fig. 6) results in the worst performance, with an adjusted mutual information of 0.4.

[2] Although in general the likelihood ratio is unbounded for Gaussian distributions, a standard truncation trick can be used to bound the weights such that our theoretical results still hold (see Lim et al., 2014).



A1: leopard, lion, wolf, bobcat, fox, german shepherd, tiger, grizzly bear, polar bear
A2: blue whale, humpback whale, walrus, seal, dolphin, killer whale, otter
A3: rabbit, mouse, hamster, squirrel, mole, beaver, skunk
A4: giant panda, sheep, cow, ox, pig, buffalo
A5: zebra, horse, antelope, deer, giraffe, moose
A6: chihuahua, persian cat, collie, siamese cat, dalmatian
A7: weasel, rat, raccoon, bat
A8: spider monkey, chimpanzee, gorilla
A9: elephant, hippopotamus, rhinoceros

F1: flippers, ocean, water, swims, fish, arctic
F2: fierce, hunter, stalker, meat, meatteeth, muscle
F3: walks, quadrapedal, ground, furry, chewteeth, brown
F4: smart, fast, active, agility
F5: claws, solitary, paws, small
F6: group, big, strong
F7: newworld, tail, oldworld
F8: white, domestic
F9: hooves, grazer, plains, fields, vegetation

Figure 4. Left: Program (1) output Y, *Right*: Raw data with the same rows/columns order

Figure 6 shows the program output Y (for $\lambda = \sqrt{5n}$), along with the raw data and its thresholded version. Each column corresponds to one sample and they have been arranged according to the ground truth clustering where columns 1-19 corresponds to B-ALL, 20-27 T-ALL and 28-38 AML. We observe that finer clustering seems to exist especially within the B-ALL group, and according to Monti et al. (2003) it is widely accepted that meaningful sub-classes of ALL do exist but the composition and nature of these sub-classes is not as well accepted.

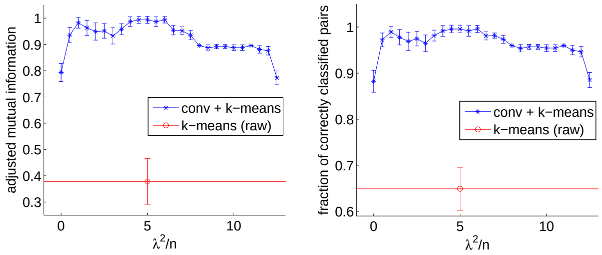


Figure 5. Clustering performance. *Left*: Adjusted mutual information, *Right*: Fraction of pairs correctly clustered together

Acknowledgments

S.H. Lim and H. Xu were supported by the Ministry of Education of Singapore through AcRF Tier Two grants R265000443112 and R265000519112, and A*STAR Public Sector Funding R265000540305. Y. Chen was supported by NSF grant CIF-31712-23800 and ONR MURI grant N00014-11-1-0688.

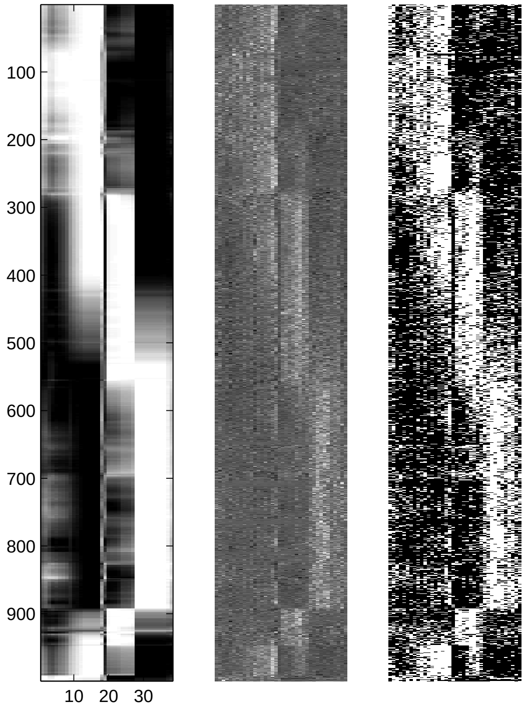


Figure 6. Left: Output Y from program (1), *Middle*: Raw data, *Right*: Raw data (binarized)

References

Ames, B. and Vavasis, S. Nuclear norm minimization for the planted clique and biclique problems. *Mathematical*

Programming, 129(1):69–89, 2011.

Ames, Brendan P.W. Guaranteed clustering and biclustering via semidefinite programming. *Mathematical Programming*, pp. 1–37, 2013. ISSN 0025-5610. doi: 10.1007/s10107-013-0729-x.

Anandkumar, Anima, Ge, Rong, Hsu, Daniel, and Kakade, Sham M. A tensor spectral approach to learning mixed membership community models. *arXiv preprint arXiv:1302.2684*, 2013.

Bansal, N., Blum, A., and Chawla, S. Correlation clustering. *Machine Learning*, 56(1), 2004.

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

Cai, T. Tony and Li, Xiaodong. Robust and computationally feasible community detection in the presence of arbitrary outlier nodes. *arXiv preprint arXiv:1404.6000*, 2014.

Chaudhuri, K., Chung, F., and Tsiatas, A. Spectral clustering of graphs with general degrees in the extended planted partition model. *COLT*, 2012.

Chen, Y., Sanghavi, S., and Xu, H. Clustering sparse graphs. In *NIPS 2012.*, 2012.

Chen, Y., Jalali, A., Sanghavi, S., and Xu, H. Clustering partially observed graphs via convex optimization. *Journal of Machine Learning Research*, 15:2213–2238, June 2014.

Cheng, Yizong and Church, George M. Biclustering of expression data. In *Proc. of the 8th ISMB*, pp. 93–103. AAAI Press, 2000.

Demaine, E. D., Immorlica, N., Emmanuel, D., and Fiat, A. Correlation clustering in general weighted graphs. *SIAM special issue on approximation and online algorithms*, 2005.

Eren, Kemal, Deveci, Mehmet, Küçüktunç, Onur, and Çatalyürek, Ümit V. A comparative analysis of biclustering algorithms for gene expression data. *Briefings in Bioinformatics*, 2012.

Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., and Bloomfield, C. D. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

Hartigan, J. A. Direct Clustering of a Data Matrix. *Journal of the American Statistical Association*, 67(337):123–129, 1972. ISSN 01621459. doi: 10.2307/2284710.

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

Kannan, R., Vempala, S., and Vetta, A. On clusterings - good, bad and spectral. In *IEEE Symposium on Foundations of Computer Science*, 2000.

Kemp, Charles, Tenenbaum, Joshua B., Griffiths, Thomas L., Yamada, Takeshi, and Ueda, Naonori. Learning systems of concepts with an infinite relational model. In *Proceedings of the 21st National Conference on Artificial Intelligence - Volume 1*, AAAI'06, pp. 381–388. AAAI Press, 2006.

Kolar, Mladen, Balakrishnan, Sivaraman, Rinaldo, Alessandro, and Singh, Aarti. Minimax localization of structural information in large noisy matrices. In *NIPS*, pp. 909–917, 2011.

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

Lim, S.H., Chen, Y., and Xu, H. Clustering from labels and time-varying graphs. In *NIPS 2014.*, 2014.

Mathieu, C. and Schudy, W. Correlation clustering with noisy input. In *SODA*, pp. 712, 2010.

McSherry, F. Spectral partitioning of random graphs. In *FOCS*, pp. 529–537, 2001.

Monti, Stefano, Tamayo, Pablo, Mesirov, Jill, and Golub, Todd. Consensus clustering: A resampling-based method for class discovery and visualization of gene expression microarray data. *Mach. Learn.*, 52(1-2):91–118, July 2003.

Oghabian, Ali, Kilpinen, Sami, Hautaniemi, Sampsa, and Czeizler, Elena. Biclustering methods: Biological relevance and application in gene expression analysis. *PLoS ONE*, 9(3), 2014.

Osherson, D.N., Stern, J., Wilkie, O., Stob, M., and Smith, E.E. Default probability. *Cognitive Science*, 15(2):251–269, 1991.

Puleo, G. J. and Milenkovic, O. Correlation Clustering with Constrained Cluster Sizes and Extended Weights Bounds. *ArXiv e-print 1411.0547*, 2014.

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

Rudelson, Mark and Vershynin, Roman. Smallest singular value of a random rectangular matrix. *Communications on Pure and Applied Mathematics*, 62(12):1707–1739, 2009.

Shamir, O. and Tishby, N. Spectral Clustering on a Budget. In *AISTATS*, 2011.

Swamy, C. Correlation clustering: maximizing agreements via semidefinite programming. In *Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms*, 2004.

Tanay, Amos, Sharan, Roded, and Shamir, Ron. Biclustering algorithms: A survey. In *In Handbook of Computational Molecular Biology*, 2005.

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

Wulff, S., Urner, R., and Ben-David, S. Monochromatic Bi-Clustering. *ICML 2013*, 2013.

Xu, Jiaming, Wu, Rui, Zhu, Kai, Hajek, Bruce, Srikant, R., and Ying, Lei. Jointly clustering rows and columns of binary matrices: Algorithms and trade-offs. *SIGMETRICS Perform. Eval. Rev.*, 42(1):29–41, June 2014.