On The Computational Complexity Of Bi-Clustering

by

Sharon Wulff

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In this work we formalize a new natural objective (or cost) function for bi-clustering - Monochromatic bi-clustering. Our objective function is suitable for detecting meaningful homogenous clusters based on categorical valued input matrices. Such problems have arisen recently in systems biology where researchers have inferred functional classifications of biological agents based on their pairwise interactions. We analyze the computational complexity of the resulting optimization problems. We show that finding optimal solutions is NP-hard and complement this result by introducing a polynomial time approximation algorithm for this bi-clustering task. This is the first positive approximation guarantee for bi-clustering algorithms. We also show that bi-clustering with our objective function can be viewed as a generalization of correlation clustering.

Acknowledgements

I would like to acknowledge the individuals who have encouraged and fostered me in the completion of my education and my masters degree. First and foremost I would like to thank my supervisor Shai Ben-David for his unsurpassed support and inspiration both mathematically and as a confidante and friend. I could not ask for a better supervisor. I would also like to thank the committee members, professors Forbes Burkowski and Ming Li for their time and remarks. I would very much like to thank my wonderful family, my beloved parents Uri and Batia, and sisters, Anat and Karin for supporting and putting up with me throughout all of my endeavors. Especially, I would like to thank my dad, without whom, I would not be in computer science today, and most of all for his unwavering love and support. I would like to thank my friend Eldar Kleiner for his great input. I would also like to thank my friend Peter Sudmant for his proofreading and support.

Contents

Chapter 1

Introduction

1.1 Background

Common clustering tasks take as input a data set and a similarity (or distance) function over the domain, with the aim of finding a partition of the data into groups of mutually similar elements. Bi-clustering is a variant of this general task, in which the input data comes from two domain sets, and instead of having a distance function over elements, the input contains some relation over the cartesian product of these sets. The Bi-clustering task is usually to partition each of the sets, such that the subsets from one domain, exhibit similar behavior across the subsets of the other domain.

Definition 1.1.1. *Given a matrix M of m rows and n columns, the* ***bi-clustering*** *task is to find subsets of the rows which exhibit similar behavior across subsets of the columns, or vice versa.*

Cheng and Church, [3], were the first to introduce bi-clustering for the purpose of gene expression analysis. Gene expression profiling has been established as a standard technique for measuring the activity (the expression) of thousands of

genes at once, under different biological conditions, to create a global picture of the cellular function. Microarrays or DNA chips, allow the measurement of mRNA levels simultaneously for thousands of genes. The set of measured gene expression levels under one condition are called the profile of that condition. A gene expression matrix, is a set of gene expression profiles , with rows corresponding to genes and columns corresponding to conditions. It is believed that the same activation patterns can be common to groups of genes across a collection of experimental conditions. This means that subsets of genes will be coexpressed under certain experimental conditions, but may behave almost independently under other conditions. Discovering such local expression patterns, the subsets of genes and conditions, may be the key to uncover many genetic pathways that are not apparent otherwise. Mathematically formulated, this is a bi-clustering task as applied to gene expression matrices.

E-commerce and online content distribution have greatly evolved during the last decade due to the world wide increase in internet and web based activities. Consumers are often required to identify content of interest from a potentially overwhelming set of choices. For example, online movie rentals, online book shopping, online audio broadcasting etc. Recommender systems have emerged as an important response to this so-called information overload problem. The goal of these systems is to give users an intelligent and proactive information service by making concrete product or service recommendations that are sympathetic to the learned preferences and needs of the customer.

One of the most dominant recommendation strategies is called Collaborative Filtering (CF). CF is motivated by the observation that people often look to their friends for recommendations. It relies on the availability of user profiles that capture the past rating histories of users. Recommendations are generated

for a target user by drawing on the rating histories of a set of suitable recommendation partners. These partners are generally chosen because they share similar or highly correlated rating histories with the target user. The CF task can be expressed as a bi-clustering problem with rows corresponding to users, columns to items, and the matrix entries describing a user-item preference, which is typically a score on some scale. Grouping the users and items into subsets which exhibit similar behavior can be used to predict missing user-item preferences as well as the behavior of new users or preferences regarding new products. A bi-clustering solution for collaborative filtering was suggested in [7] and in [5].

Bi-clustering, or co-clustering as it is sometimes referred to in the literature, is far from being a new framework. Bi-clustering algorithms have been applied to various fields, such as systems biology, information retrieval, text categorization and data mining. While bi-clustering tasks share the same general goal, namely, finding homogeneous patterns in a given data matrix, they vary in the mathematical formulation of the homogeneity, or similarity notion. The mathematical formulation of a bi-clustering task is usually given by the cost function it is aiming to optimize. The cost function conveys the similarity notion which the specific bi-clustering is trying to recover. Under a specific cost function, a score is assigned to each possible partition of the matrix, and the task becomes an optimization problem of finding the lowest (highest) cost partition. Naturally, there exist a host of bi-clustering algorithms, yet, in many cases there is little by way of a formal definition of the bi-clustering tasks. There is even less bi-clustering related work presenting an analysis of the complexity of the task. While some NP-hardness results exist for some formulations of bi-clustering, we are not aware of any positive approximation guarantees.

One active line of research is the development of bi-clustering algorithms without an explicit cost function. For example, the Coupled Two-Way Clustering (CTWC) introduced in [8], defines a generic scheme for transforming a one dimensional clustering algorithm into a bi-clustering algorithm. The "SAMBA" algorithm [16] uses probabilistic modeling of the data and graph theoretic techniques to identify subsets of genes that jointly respond across a subset of conditions. Spectral bi-clustering which employs linear algebraic techniques to identify meaningful structures, is another example of a class of cost function-less bi-clustering techniques. Such a technique was introduced in [13] and was applied to bi-clustering of a microarray data.

The first definition of bi-clustering as an optimization problem over some well defined objective function was introduced by Hartigan [12]. Hartigan considers real valued matrices and proposes several objective functions, including the sum of block's variances(in the blocks induced by the clusterings of rows and columns). Hartigan also proposes a heuristic for finding low-cost bi-clustering, however, with no performance guarantees either in terms of the quality of the optimization or in terms of its computational complexity.

Cheng and Church, in [3], introduced bi-clustering for the purpose of gene expression analysis. They formally define a cost function called the *low mean squared residue*, which can be viewed as a variant of Hartigan's minimum variance cost where the row and column averages of each entry are taken into account as follows: The residue of an element a_{ij} in the block A_{IJ} is, $r_{ij} = (a_{ij} - a_{Ij} - a_{iJ} + a_{IJ})$ where a_{Ij}, a_{iJ} and a_{IJ} are the row and column and block means. The mean squared residue of a block A_{IJ}, denoted as H is, $H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (r_{ij})^2$. The motivation behind this cost function is that

a "good" block should have a constant expression level plus possibly additive row and column specific effects. After removing row, column and block averages, the residual should be as small as possible. Cheng and Church propose an iterative search algorithm and as such converge to a local minima.

Other notable objective functions used in bioinformatics include the *loss in mutual information* of Dhillon, Mallela, and Modha [6], the *square residue* of Cho, Dhillon, Guan, and Sra [11]. In all of these papers, neither optimization quality guarantees nor computational complexity bounds are proved.

1.2 Monochromatic Bi-clustering

We formalize a new bi-clustering cost function, which we call monochromatic bi-clustering. Given an input matrix over some fixed finite domain of values we wish to partition the rows and the columns of the matrix such that the resulting matrix blocks are as homogeneous as possible.

The *monochromatic cost* of a partition of a given matrix, is the sum over the partition induced blocks, of the number of entries that are different than the majority in their block.
Formally, Given an input matrix M, the *monochromatic cost* of partitions $P^R = (R_1, \ldots, R_k)$ of the matrix rows and $P^C = (C_1, \ldots, C_\ell)$ of the matrix columns is the following sum $\frac{\sum_{1 \le s \le k,\, 1 \le t \le \ell} \phi(R_s \times C_t)}{|M|}$ where for a matrix block B,

$$\phi(B) = \min(|\{(\imath, \jmath) : (\imath, \jmath) \in B\,,\, M(\imath, \jmath) = 0\}|, |\{(i, j) : (\imath, \jmath) \in B\,,\, M(\imath, \jmath) = 1\}|)$$

namely, the number of B's entries that are not equal to the majority value in B. In the case of a tie, we arbitrarily declare one of the tied most frequent labels

as the majority label.

For the most part in this thesis we focus on binary matrices, however, our results can be easily extended to matrices over multiple values.

1.2.1 Motivation

The motivation for our formulation comes from applications of bi-clustering in systems biology. Recent research into various domains of systems biology addresses the analysis of networks of interactions. In studying networks of metabolic yeast genes Segre, DeLuna, Church and Kishony [4] have found that gene pairs can be grouped into "functional modules" that interact with each other "monochromatically". That is, the question of whether a pair of genes interact by buffering or aggravating each others individual effects is almost purely determined by their group (or module) memberships. Similarly, Yeh, Tscumi and Kishony [14] investigated multidrug effects by analyzing the network of pairwise interactions between antibiotics that effect the growth rate of Escherichia coli. They found that the drugs could be separated into classes such that antibiotics from two classes interacted either synergistically or antagonistically, based purely on the classes to which they belonged. It turns out that these classes correspond to the cellular functions effected by the drugs. These studies suggest that clustering of biological agents based on their pairwise interactions can be applied to discover the functional classifications of these agents.

1.2.2 Correlation Clustering Relation

While some of the bi-clustering problems model situations in which the input objects come from disparate domains (like movies and viewers or text documents and words), one can also consider a symmetric variant of the bi-clustering paradigm. This version models the case where the input matrix records pair-

wise relations among the elements of a single set. In the symmetric problem, the input matrix is a square symmetric matrix, and the output partitions of the rows and columns are required to be identical. The results of this research apply to both versions of the problem, however, for the sake of concreteness, our presentation is mostly in terms of the non-symmetric version.

It is interesting that a version of a clustering problem, as opposed to bi-clustering, *correlation clustering* as defined by Bansal,Blum and Chawla, [2], can be viewed as a special case of *symmetric monochromatic bi-clustering*. Correlation clustering is concerned with clustering sets that have a redundant version of a similarity relation over them. The similarity takes only one of two values "similar" or "dissimilar". Formally, the input to correlation clustering is a complete graph with binary labeled edges, $\{-1, +1\}$. The clustering goal is to find a partition of the graph vertices into groups (clusters) such that there are as few as possible -1 edges inside clusters and as few as possible $+1$ edges between vertices in different clusters. Giotis and Guruswami [9], analyze the *correlation clustering with a fixed number of clusters* version of the problem, in which the number of clusters is predetermined as part of the input. By viewing the input to the symmetric monochromatic bi-clustering task as a complete graph with labeled edges, one can readily model the correlation clustering with a fixed number of clusters (see section 1.3 for details). Note that our bi-clustering task is strictly more general. The correlation clustering checks only for the configuration that has one value along the matrix diagonal blocks and another value off the diagonal. For example, the systems biology applications do not conform to such a restriction.

One should note that while in correlation clustering the problem is meaningful

even when the number of clusters is not predetermined, the determination of the numbers of row and column clusters in our problem is inevitable, since otherwise the trivial singleton partition always achieves an optimal zero-cost.

1.2.3 The Contribution of This Research

Apart from introducing the monochromatic bi-clustering problem, we focus on analyzing the complexity of the resulting optimization task. We prove that on the one hand finding an optimal solution to monochromatic bi-clustering is NP-hard, on the other hand however, we present an approximation algorithm with performance guarantees. Namely, we introduce an algorithm for finding a close to optimal bi-clustering and prove that it is a polytime approximation scheme for the monochromatic bi-clustering task.

In spite of the existence of a wide variety of bi-clustering algorithms, there exists very little theoretical analysis of bi-clustering problems. In particular, we are not aware of any previous positive approximation guarantees for a bi-clustering algorithm.

Our approximation results, though stated with respect to binary matrices, can easily be extended to matrices over multiple values. This results in an increase in computational complexity which is exponential in the number of values (though this number is typically small), yet still polynomial in the size of the input matrix. The extension can be applied to problems where the domain consists of multiple categorical values.

The remainder of this thesis is organized as follows. After establishing some

notations and basic definitions in 1.3, we show that the monochromatic bi-clustering solution is NP-Hard in 2 and present an approximation algorithm for the monochromatic bi-clustering problem in 3. In section 4 we present additional research directions which were explored relating to the monochromatic bi-clustering task. Section 5 presents our conclusions.

1.3 Preliminaries

Let $M \in \{0,1\}^{m \times n}$ denote a binary input matrix and let R and C denote M's rows and columns respectively. Let $a = (\imath, \jmath)$ denote a single entry in the matrix. We denote by k and ℓ the number of rows and columns clusters. Let P^R, P^C denote partitions of the rows and columns of M $P^R = (R_1, \ldots, R_k)$ and $P^C = (C_1, \ldots, C_\ell)$ (i.e. $R = \bigcup R_\imath$ and for $\imath \neq \jmath, R_\imath \cap R_\jmath = \emptyset$ and similarly for C). We use $P = (P^R, P^C)$ to denote a partition for M into the $k \times \ell$ sub-matrices, $\{R_s \times C_t : s \leq k,\ t \leq \ell\}$.

Definition 1.3.1. *(Monochromatic cost) Given an input matrix M and a partition $P = (P^R, P^C)$, the* Monochromatic cost *of the partition reflects the amount of non-homogeneity of the entries in the sub matrices,*

$$Mon^D(M,P) = \frac{\sum_{(s,t):s\leq k, t\leq \ell} \phi(R_s \times C_t)}{|M|} \qquad \textit{for a matrix block } B,$$

$$\phi(B) = \min(|\{(\imath,\jmath) : (\imath,\jmath) \in B\,,\, M(\imath,\jmath) = 0\}|, |\{(i,j) : (\imath,\jmath) \in B\,,\, M(\imath,\jmath) = 1\}|)$$

namely, the number of B's entries that are not equal to the majority value in B. The superscript D denotes Disagreement.

Definition 1.3.2. *Given an input matrix M and a partition $P = (P^R, P^C)$, the* ***Monochromatic agreement*** *of the partition is*

$$Mon^A(M,P) = 1 - Mon^D(M,P)$$

In the maximization version of the problem we seek a partition of the input matrix which maximizes the Monochromatic agreement.

Claim 1.3.3. *For every matrix M, there exists a partition P, such that $Mon^A(M,P) \geq \frac{1}{2}$*

Proof. this is true since the cost of the partition that leaves all of the matrix's rows in the same cluster and all of the matrix's columns in the same cluster is equal to

$$\phi(M) = \frac{\min(|\{(\imath,\jmath) : M_{\imath,\jmath} = 0\}|, |\{(i,j) : M_{\imath,\jmath} = 1\}|)}{|M|} \geq \frac{1}{2}.$$

Definition 1.3.4. *a **label matrix** is a $k \times \ell$ binary matrix, $L \in \{0,1\}^{k \times \ell}$.*

We use label matrices to represent the majority values of the entries in the $k \times \ell$ blocks induced by a matrix partition. Such a matrix can be viewed as a compressed representation of the original matrix.

Definition 1.3.5. *Given a partition, $P = (P^R, P^C)$, of an input matrix M, and a label matrix L, the L-**monochromatic cost** of the partition is the sum over all pair $\{(\imath,\jmath) : \imath \leq \ell, \jmath \leq k\}$ of $\phi_L(R_\imath \times C_\jmath)$, where $\phi_L(R_\imath \times C_\jmath)$ for a matrix block, $R_\imath \times C_\jmath$, is the number of entries in the block whose values differ from $L_{\imath,\jmath}$. We will use L-**cost** and L-**monochromatic cost** interchangeably.*

The L-monochromatic cost of a partition is the sum over blocks induced by the partition, of the number of the block's entries which are different from the label of the block (instead of the majority value of the entries in the block).

Definition 1.3.6. *Given a set of rows R and a set of columns C we say that $R \sim C$ if $|R| = |C|$ and there is a bijection $h : R \to C$ such that $\forall r \in R$, $r = h(r)^T$. Namely, viewed as sets of vectors, R and C are identical.*

A feasible solution for a bi-clustering problem, where the input matrix consists of pairwise relations between the elements of a single set, must have an identical partition of the rows and columns. Both are in fact clusterings of the same set. Definition 1.3.6 is used to formulate this requirement.

Definition 1.3.7. *Given a square matrix M, a $k \times k$ bi-clustering of M, $((R_1, \ldots, R_k), (C_1, \ldots, C_k))$ is **symmetric** if there exist a permutation π of the set of indices, $\{1, \ldots, k\}$, such that $\forall i$ $R_i \sim C_{\pi(i)}$.*

Definition 1.3.8. *Given a matrix M, a $k \times k$ bi-clustering of M, $((R_1, \ldots, R_k), (C_1, \ldots, C_k))$ is **diagonal** if, for all $i \leq k$, the majority value of the entries in the block $R_i \times C_i$ is 1 where the majority in each off-diagonal block is 0.*

A label matrix representing a diagonal monochromatic solution has 1 along the diagonal and 0 in the off diagonal entries.

Definition 1.3.9. *The **correlation clustering** problem in a **graph notation** is given a complete graph on n nodes with each edge labeled either $+$ (similar) or $-$ (dissimilar), find a partitioning of the vertices into clusters that agrees as much as possible with the edge labels. The minimization version, aims to minimize the number of disagreements, namely the number of $-$ edges within clusters plus the number of $+$ edges between clusters.*

Definition 1.3.10. *The **correlation clustering** problem in a **matrix notation** is to find, for a binary symmetric matrix $S \in \{0,1\}^{n \times n}$ and an integer $k \in N$, a symmetric partition of S into k clusters, which minimizes the L-monochromatic cost, where L is a label matrix with 1 in the diagonal entries and 0 in the off diagonal entries.*

Claim 1.3.11. *Definitions 1.3.9 and 1.3.10 are equivalent formulations of the same optimization problem.*

Chapter 2

Hardness result

In this chapter we show that finding an optimal solution for the monochromatic bi-clustering problem is NP-hard. We first show its hardness for k=ℓ=2. We do so by showing that the correlation clustering problem for 2 clusters reduces to the 2×2-monochromatic bi-clustering problem. The hardness of this problem was shown by Giotis and Guruswami in [9]. We then show the hardness of $k \times \ell$-monochromatic bi-clustering for larger values of k and ℓ by reducing it to the 2×2 case.

Given an input Co, to the correlation clustering problem, we translate it to an input matrix $M = G(Co)$ such that the optimal 2×2-monochromatic bi-clustering solution for M is diagonal, symmetric, and induces an optimal solution for Co.

2.1 Construction of $M = G(Co)$

In order to force a diagonal solution, we build $G(Co)$ by padding Co with an extra $2N$ rows and $2N$ columns, for $N = 4n$, where n denotes the number of rows (and columns) in Co. $G(Co)$ consists of 9 blocks, $\{B_{\imath,\jmath}\}_{1 \leq \imath,\jmath \leq 3}$. The central

block, $B_{2,2}$ is the input matrix Co. The 4 corner blocks, $B_{1,1}$, $B_{1,3}$, $B_{3,1}$ $B_{3,3}$ will be square blocks of size $N \times N$ each. The remaining four blocks will therefore have size $n \times N$ each.

As for the entries of this matrix, M, all entries in $B_{1,1}$ and $B_{3,3}$ will have value 1. The entries of the central block, $B_{2,2}$ will stay as they are in Co, and all the other entries (in the remaining 6 blocks) assume value 0. We use $R(B_{\imath,*}), \imath \leq 3$ to denote the set of matrix rows in the $\imath$'s row block (on all column blocks), similarly we use $C(B_{*,\jmath}), \jmath \leq 3$ to denote the set of matrix columns in the $\jmath$'s column block, as shown in Figure 2.1. We claim that, having picked large enough N, an optimal 2×2-monochromatic clustering of M, denoted as $P^R = (R_1, R_2)$ and $P^C = (C_1, C_2)$, must have a diagonal structure. Namely, the bi-clusters $C_1 \times R_1$ and $C_2 \times R_2$ assume a majority value 1, while the other two bi-clusters have majority value of 0.

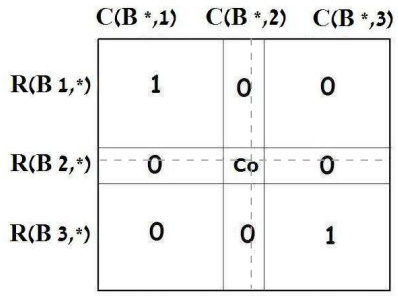

Figure 2.1: The reduction mapping $M = G(Co)$

2.2 Reduction Proof

Theorem 2.2.1. *The optimal 2×2-monochromatic bi-clustering solution for M is diagonal, symmetric, and induces an optimal solution for Co.*

Lemma 2.2.2. *There exists a* 2×2 *bi-clustering of* M *that has error at most* $n^2 + 2nN$.

Proof. Consider the partition defined by $C_1 = (c_1, ...c_{n/2+N})$. That is, without changing the order of either the rows or columns of M, we cut both the rows and the columns half way through M. It is not hard to see that for this partition the majority label for $C_1 \times R_1$ and for $C_2 \times R_2$ is 1, and the majority label for the other two blocks is 0. Furthermore, the four corner blocks, $B_{1,1}, B_{1,3}, B_{3,1}, B_{3,3}$ are perfectly labeled, so all the minority labeled entries belong to the non-corner $B_{\imath,\jmath}$'s, and they contain just $n^2 + 4nN$ entries. The cost is furthered reduced to $n^2 + 2nN$ since the non-corner blocks, have a value '0' and therefor are not errors in the blocks with majority value '0'. It is easy to verify that exactly half of their $4nN$ entries will be placed in these blocks.

Lemma 2.2.3. *If* $N \geq 4n$, *any optimal* 2×2*-monochromatic bi-clustering solution for* M, *denoted by OPT(M), is diagonal.*

Proof. It is not hard to see that any non-diagonal solution (that is, a solution that has two adjacent blocks of the partition assume the same majority value, is bound to have all the entries in one of the four corner $B_{\imath,\jmath}$'s misclassified (that is, their label does not agree with the majority label of their block). Consequently, any non-diagonal labeling of a solution 2×2 bi-clustering has, inevitably, at least $N^2 \geq n^2 + 2nN$ errors, and therefore is not optimal.

Lemma 2.2.4. *For a symmetric matrix* S *if a minimal monochromatic-cost* 2×2 *bi-clustering solution of* S, $OPT(S)$, *is diagonal, then it is a symmetric bi-clustering.*

Proof. Assume the contrary, namely that $OPT(S)$ is not symmetric. Then there exist at least one row and column pair which is placed in different clusters. We denote this pair $(\overline{r}, \overline{c})$. Note that because S is symmetric we can consider the

rows and columns as pairs ($\overline{r} = \overline{c}^T$). Denote the partitions P=(P^R,P^C) where $P^R = R_1$,R_2 and $P^C = C_1$,C_2. WLOG let us assume that the permutation which agrees the most with the rows and columns pairs, matches R_1 to C_1 and R_2 to C_2. Assume that $\overline{r} \in R_1$ and $\overline{c} \in C_2$.

In order to prove the lemma, we compare the number of entries of the row vector $\overline{r}$, that are different from their block's majority value in the partition P, with the number of entries of the column vector $\overline{c}$, that are different from their block's majority. This comparison leads to the conclusion that either $\overline{r}$ or $\overline{c}$ has a better placement. By moving either $\overline{r}$ to R_2 or $\overline{c}$ to C_1 we can reduce the monochromatic cost of the partition P and therefore P is not optimal.

We can divide the clusters R_1,R_2 and C_1,C_2 into two sets. Pairs on which the partition agrees will be in one set, and its compliment will be in the other set. Denote those sets by R_{1a}, R_{1d} and R_{2a}, R_{2d} and for the columns C_{1a}, C_{1d} and C_{2a}, C_{2d}. Note that the following observations hold

1. $R_{1a} \sim C_{1a}$ and $R_{2a} \sim C_{2a}$
2. $R_{1d} \sim C_{2d}$ and $R_{2d} \sim C_{1d}$
3. $\overline{r} \in R_{1d}$ and $\overline{c} \in C_{2d}$

Figure 2.2 illustrates the above partition.
Since the partition P is a diagonal the majorities in the induced sub matrices are as follows

	C_1	C_2
R_1	1	0
R_2	0	1

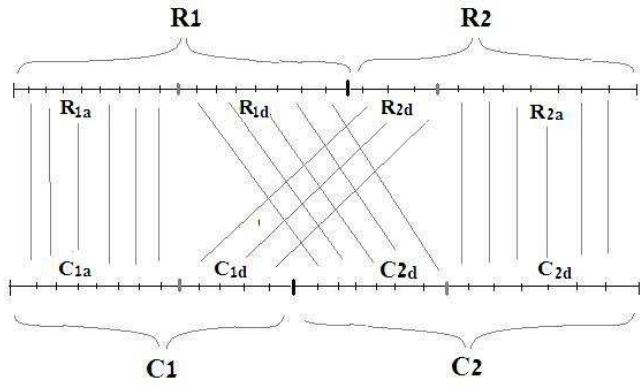

Figure 2.2: The symmetric matrix rows and columns partition.

We denote by $\{r \bigcap C\}$, where r is a row vector and C is a set of columns, the entries of r which intersect with the columns in C. We define $\{c \bigcap R\}$ in a similar manner. We call entries which are similar to the majority in their block "successes", we call them "mistakes" otherwise.

The entries of $\overline{r}$, can be divided into four groups according to their intersection with the four columns subsets we defined earlier, we get $\{\overline{r} \bigcap C_{1a}\}$, $\{\overline{r} \bigcap C_{1d}\}$,$\{\overline{r} \bigcap C_{2a}\}$ and $\{\overline{r} \bigcap C_{2d}\}$. In the same manner we can divide the entries of $\overline{c}$ into $\{\overline{c} \bigcap R_{1a}\}$,$\{\overline{c} \bigcap R_{1d}\}$, $\{\overline{c} \bigcap R_{2a}\}$ and $\{\overline{c} \bigcap R_{2d}\}$. Recall that $\overline{r}$ and $\overline{c}$ are essentially the same vector, and that $\overline{r} \in R_1$ and $\overline{c} \in C_2$. We perform an analysis of the L-cost derived by the above subsets in the partition P:

1. The subsets $\{\overline{r} \bigcap C_{1a}\} = \{\overline{c} \bigcap R_{1a}\}$ consists of the same entries values, because $R_{1a} \sim C_{1a}$. However, $\{\overline{r} \bigcap C_{1a}\}$ reside in the block $R_1 \times C_1$ where the majority of the entries is '1' while $\{\overline{c} \bigcap R_{1a}\}$ reside in $R_1 \times C_2$ where the majority of the entries is '0'. This means that for these subsets the number of "mistakes" derived by one is exactly the number of "successes" derived by the other and vice versa.

2. The subsets $\{\overline{r} \bigcap C_{2d}\} = \{\overline{c} \bigcap R_{1d}\}$ have the same entries values, both

$\{\overline{r} \bigcap C_{2d}\}$ and $\{\overline{c} \bigcap R_{1d}\}$ reside in the block $R_1 \times C_2$ where the majority is '0'. This means that these subsets exhibit the same number of "mistakes" and "successes" in P.

3. The subsets $\{\overline{r} \bigcap C_{1d}\} = \{\overline{c} \bigcap R_{2d}\}$ have the same entries values, $\{\overline{r} \bigcap C_{1d}\}$ reside in the block $R_1 \times C_1$ where the majority value is '1' , $\{\overline{c} \bigcap R_{2d}\}$ reside in $R_2 \times C_2$ where the majority value is '1' as well. This implies that these subsets exhibit the same number of "mistakes" and "successes" in P.

4. The subsets $\{\overline{r} \bigcap C_{2a}\} = \{\overline{c} \bigcap R_{2a}\}$ have the same entries values, $\{\overline{r} \bigcap C_{2a}\}$ reside in the block $R_1 \times C_2$ where the majority is '0' while $\{\overline{c} \bigcap R_{2a}\}$ reside in $R_2 \times C_2$ where the majority value is '1'. Like in 1, the number of "mistakes" derived by one is equal to the number of "successes" derived by the other and vice versa.

From the above comparison it is clear that either $\overline{c}$ or $\overline{r}$ has a better "score" of "mistakes" vs. "successes" in its current placement. We can then move the less successful one to the other cluster and thus achieve a reduction in the overall cost. This is a contradiction because we assumed that the solution is optimal.

Lemma 2.2.5. *For $N \geq 4n$ any optimal 2×2 bi-clustering for M partitions the rows in M such that the newly added row sets $R(B_{1,*})$ and $R(B_{3,*})$ reside in different row clusters and similarly $C(B_{*,1})$ and $C(B_{*,3})$ reside in different column clusters.*

Proof. By Lemma 2.2.3 any such optimal solution is diagonal. WLOG we consider the case that the blocks along the diagonal have majority value 1. The partition that places $R(B_{1,*})$ and $R(B_{3,*})$ in different clusters and $C(B_{*,1})$ and $C(B_{*,3})$ in different clusters, yields a perfect zero error cost for every entry in

each of the blocks $B_{1,1}$,$B_{1,3}$,$B_{3,1}$ and $B_{3,3}$ (with respect to the diagonal solution we consider). For each of the rows and columns in $R(B_{1,*})$, $R(B_{3,*})$ and $C(B_{*,1})$, $C(B_{*,3})$ the cost in this partition is at most n, whereas in any different placement, each displaced row or column incurs a cost of at least N.

Lemma 2.2.6. *Any optimal solution, $P = (P^C, P^R)$ for M induces a feasible solution for the correlation clustering problem on the matrix Co.*

Proof. Consider the partition of the rows (or columns) of Co induced by P (that is, the restriction of P to the sub-matrix, Co). This is a symmetric solution for Co since Co is a diagonal block in M and, by Lemma 2.2.4, P is a symmetric solution for M. In the correlation clustering setup the rows and columns represents the same objects. As a result any feasible solution for the correlation clustering problem must be symmetric (see definition 1.3.7). We have shown in Lemma 2.2.4 that an optimal bi-clustering solution for a symmetric matrix is symmetric. Since M is symmetric the partition of Co induced by P, an optimal bi-clustering solution, is symmetric and therefore a feasible solution for the correlation clustering problem for Co.

Lemma 2.2.7. *For any solution, $P = (P^C, P^R)$ for M, if its cost is no more than $n^2 + 2nN$ then its cost is exactly $2nN + Cost^P(Co)$, where $Cost^P(Co)$ is the correlation clustering cost of the partitioning that P induces on Co.*

Proof. By Lemma 2.2.5, every such solution makes no errors on the 4 corner blocks $(B_{1,1}, B_{1,3}, B_{3,1}, B_{3,3})$ and exactly $2nN$ errors on the other blocks except the central block, Co (namely the blocks $B_{1,2}, B_{2,1}, B_{3,2}, B_{2,3}$). This is easy to verify.

Lemma 2.2.8. *Given any correlation clustering solution, Q for Co (for $k = 2$), there exist a solution P for the matrix M whose monochromaticity cost on M is $2nN + Cost^Q(Co)$.*

Proof. Just extend Q to a solution for M by placing the additional blocks $B_{i,j}$ as described above.

Corollary 2.2.9. *An optimal (= minimal cost)* 2×2- *monochromatic bi-clustering for* M *induces an optimal correlation clustering solution for the input matrix,* Co.

2.2.1 Hardness For All Values

Lemma 2.2.10. *The* $k \times \ell$ *monochromatic bi-clustering problem is NP hard for all values* $k \geq 2, \ell \geq 2$.

Proof. The proof is by induction on k and ℓ. Given an input Co for the $k \times \ell$ bi-clustering problem, we pad it up to a matrix J in such a way that an optimal $(k+1) \times \ell$ optimal bi-clustering for J induces an optimal $k \times \ell$ bi-clustering of Co.

In the matrix padding we make use of another entry value s, which is not in 0,1. For the induction step with respect to the rows, we add $2n$ new rows, where n is the number of rows in the original matrix. The new entries all assume the value s. As a result, the matrix size is tripled and its majority is now s. Consequently, under any partition, at least one block would have a majority value s. Finally, note that it follows that any optimal partition must place all of the new rows in that block and all original rows in different blocks (otherwise they will be considered errors). This partition uses one row cluster for the new rows, leaving k and ℓ partitions to be made on the original matrix.

Chapter 3

Approximation Algorithm

3.1 Approximation Algorithm For The Monochromatic Bi-clustering

In this section we present a polynomial time approximation scheme (PTAS) for solving the monochromatic bi-clustering problem for every fixed k and ℓ. Namely, given any accuracy parameter, ϵ, for every input matrix, the algorithm runs in polynomial time and outputs a bi-clustering whose agreement score is within ϵ of the best possible bi-clustering for that matrix.

Theorem 3.1.1. *There exists an algorithm for the monochromatic bi-clustering problem, which given an input matrix M and an accuracy parameter ϵ, runs in time $2^{\frac{c}{\epsilon^2}}$, for some constant c and outputs a partition P of M, such that $Mon^A(M,P) \geq OPT - 4\epsilon$ with high probability.*

A pseudo-code of the algorithm is displayed in Algorithm1.

Algorithm 1 Monochromatic bi-clustering

Input: matrix M, label matrix L, ϵ, δ

Initialize $t = \frac{1}{2\epsilon^2} \log \frac{kl}{\delta\epsilon}$.

Choose uniformly at random t rows from R $r_1 ... r_t$,

and t columns from C $c_1 ... c_t$

for each Partition $R_1 ... R_k$ of $r_1 ... r_t$ and each Partition $C_1 ... C_l$ of $c_1 ... c_t$ **do**

  /* Compute the induced row and column partitions */

  for each of the rows $r \in R$ and columns $c \in C$ **do**

    compute $rowBlock = \min_{1 \le i \le k} \varphi_S(r, \overline{P^C}, L, i)$

    compute $columnBlock = \min_{1 \le i \le l} \varphi_S(c, \overline{P^R}, L, i)$

    put r in $rowBlock$, put c in $columnBlock$

  end for

  compute the monochromatic L-cost of the partition

end for

Return the partition with the minimal cost

3.1.1 Further Notation

We use lower case letters to denote entries of a matrix, a stands for an entry $M(\imath, \jmath)$ of a matrix M. $R(a) = R_\imath$ denotes the **row block** to which the entry a belongs, similarly $C(a) = C_\jmath$ denotes a **column block**. We use S to denote a sample, where S is a set of indices (can be used to denote either rows or the columns indices), we use t to denote the sample size therefor $t = |S|$. We use $\overline{P^R} = \overline{R_1}..\overline{R_k}$ and $\overline{P^C} = \overline{C_1}..\overline{C_\ell}$ to denote *sample* partition of the rows and columns respectively. Note that $\overline{R}(a)$ is defined only when $\imath \in S$ and similarly $\overline{C}(a)$ is defined only for $\jmath \in S$. We define the following two functions,

$$Err^R(a, L, \overline{P^R}, \imath) = \begin{cases} 1, & M(a) \neq L(\overline{P^R}(a), \imath) \\ 0, & else \end{cases}$$

$$\varphi_S(c, \overline{R_q}, L, \imath) = \frac{1}{t} \sum_{a \in c \times S} Err^R(a, L, \overline{P^R}, \imath)$$

The above function φ_S returns for a given column c the cost of placing it in the i'th column block, based on a sample of its entries (sample of the rows) and with respect to the Label matrix L. We define φ_S for the rows in a similar manner. Finally, we denote by OPT the optimal solution for the monochromatic problem.

3.1.2 Algorithm Overview

The algorithm uses ideas from the PTAS for Max k-CUT by Goldreich et al. in [10]. The input for our algorithm is a matrix $M \in 0, 1^{m \times n}$, ϵ and δ. The algorithm outputs a partition $P^R = (R_1, \ldots, R_k)$ and $P^C = (C_1, \ldots, C_\ell)$ of the rows and columns of M.

A basic step in the run of the algorithm, is a computation of a partition of the matrix with respect to a specific label matrix. This step is repeated for every possible label (up to isomorphism of label matrices) that the monochromatic solution can assume. Each such run yields a partition whose L-cost is close to optimal for the tested label. The partition with the minimal cost over the runs is returned.

After a label matrix has been fixed, the matrix partition, is done in the following manner. The algorithm chooses randomly a sample of the rows and a sample of the columns. Then the algorithm goes over all of the possible partitions of the sample into clusters (k clusters for the rows and ℓ clusters for the columns). For each partition of the **sample** of the **rows**, the partition of the **entire set** of the matrix **columns** is computed and for each partition of the **sample** of the **columns**, the partition of the **entire set** of the matrix **rows** is computed. The monochromatic cost of the partition of the rows and columns is then computed and the partition with the lowest cost is returned as the lowest partition for this label matrix.
We will show that given a partitioned sample of the rows and a label matrix, there is only one optimal partition of the matrix's columns, and vice versa and that this partition can be found in polynomial time.
The sample size of the rows and the columns depends on ϵ - the approximation accuracy one is aiming for (but is independent of the input size). The sample allows us to consider the set of all partitioning of the samples rather than considering all possible partitions of M.

We will show that taking a sample size of $\mathrm{O}(\frac{1}{\epsilon})$ suffices to approximate $\mathrm{OPT}(M)$ within an additive factor of 4ϵ with high probability. The basic step of our al-

gorithm, therefore, performs $\exp(O(\frac{1}{\epsilon}))$ operations.

The following claims form the rationale underlying our algorithm. The claims are stated with respect to the rows and any fixed label matrix L, but hold just the same for the columns.

1. Given any partition of the rows, we can find the optimal columns partition with respect to the L-cost (and the given row partition) in polynomial time. For each column, the number of mistakes it incurs in each potential column block (entries which are different than the label given by L) can be counted, for every column block placement (there are ℓ possible column blocks).

2. For any fixed partition of the rows, if we draw a large enough sample of the rows, then for most columns, their L-cost with respect to the partition of the sample of the rows, is close to their true L-cost w.r.t. the partition of the entire set of rows.

3. Given a sample S^R of the rows, and a partition $\overline{P^R}$ of this sample, we say that a partition, P^C, of the columns of M is *induced by* $\overline{P^R}$ and L, if P^C is an optimal column partition with respect to that partition of the sample rows and to the label matrix L.

4. Given a sample S^R or rows, we say that a partition, P^C, of the columns of M is *induced by* S^R, L, if *there exist* a partition, $\overline{P^R}$, of S^R (into k subsets) such that P^C is an optimal column partition with respect to that partition of the sample rows and the label matrix, L.

5. If one picks large enough samples of the rows and the columns, then the set of all the partitions of M induced by these samples, approximates the

set of all possible partitions of M. In particular, with high probability, one of these sample-induced partitions will have an L-cost that is close to the best possible L-cost of M.

3.1.3 Algorithm analysis

Theorem 3.1.2. *On input M, L, ϵ, δ, with probability at least 1 - δ the algorithm outputs k, ℓ partition of M such that the L-cost of the partition is not larger than OPT+ 4ϵ.*

Proof. We first consider the partition of the columns.
Let $W_1^R ... W_k^R$ denote the partition of M in OPT. We say that the sample $r_1 ... r_t \subset R$ is *good* with respect to $W_1^R ... W_k^R$ if there exists a partition $R_1 ... R_k$ of $r_1 ... r_t$ such that, for all columns $x \in C$, except for at most $\epsilon|C|$ columns, the following holds:

Equation 3.1.3. *For all $j \leq \ell$, $|\varphi_S(x, R_1 ... R_k, L, j) - \varphi_c(x, W_1^R ... W_k^R, j)| \leq \epsilon$*

Given a *good* sample $r_1 ... r_t$ and a *good* partition $R_1 ... R_k$, we know that for all but a fraction of ϵ of the columns in C 3.1.3 holds, therefore,

1. Assume that $x \in C$ satisfies 3.1.3. Placing x in a greedy manner with respect to $R_1 ... R_k$ and the labeling matrix L, which is what the algorithm does, yields a cost increase compared to the optimal which is bounded by $m\epsilon$. Where ϵ is the entry error percentage (see definition of φ_S) and m is the number of entries in x. Since the total number of columns is n, the cost increase in this case compared to the optimal solution is bounded by $mn\epsilon$.

2. Assume that $x \in C$ is a column which does not satisfy 3.1.3. The number of entries disagreeing with the majority in this column is bounded by m, which is the size of the column. Assuming that $r_1 ... r_t$ is a good sample

and that $R_1...R_k$ is a "good" partition of it the number of such columns is bounded by $n\epsilon$. This implies a cost increase compared to the optimal solution, of $mn\epsilon$

3. Altogether the number of errors for the above columns partition is $mn\epsilon$ + $mn\epsilon = 2mn\epsilon$

The same analysis can be applied for the partition of the rows assuming we have a *good* columns sample and a good partition of this sample. Therefore the overall increase compared to the optimal solution is $4mn\epsilon$.

Since we defined the monochromatic cost as a fraction of the number of mistakes made by the partition divided by the size of the matrix (mn), we can conclude that the algorithm yields a solution which is at most OPT+4ϵ.
Now it suffices to show the following.

Lemma 3.1.4. *With probability at least 1 - δ over the random sampling, the columns and rows, $c_1...c_t$ and $r_1...r_t$, picked by the algorithm are* good *w.r.t the partitions in the optimal solution.*

Proof. We inspect the rows sample $r_1...r_t$ which was chosen uniformly in R, let $R_1...R_k$ be the optimal partition of the sample w.r.t the partition of the rows in the optimal solution $W_1^r...W_k^r$, so that

$$R_\imath = W_\imath^r \bigcap \{r_1...r_t\} \ \forall \ 1 \leq \imath \leq k$$

We can focus our analysis on the closest to optimal partition of the sample. Since our algorithm goes over all possible partitions of the sample, this partition will be considered by the algorithm.

We consider a column $x \in C$, and a partition of the rows P^R, for each row $r_\imath \in S$ in the **sample** $1 \leq \imath \leq t$ and for each column block (column cluster)

$1 \leq \jmath \leq \ell$ we define a random 1/0 variable, $\xi_\jmath^\imath$,

$$\xi_\jmath^\imath = \begin{cases} 1 & \text{if } M(r_\imath, x) \neq L(P^R(r_\imath), \jmath) \\ 0 & \text{otherwise.} \end{cases}$$

The partition of the rows sample P^R, assigns a row cluster for each row in the sample, $P^R(r_\imath)$ is the index of the row cluster assigned to $r_\imath$. The index $\jmath$ $1 \leq \jmath \leq \ell$, is an index of a column cluster, a candidate to place the column x in. Therefore $L(P^R(r_\imath), \jmath)$ is the entry in the label matrix L that corresponds to the majority of the block $R_{P^R(r_\imath)} \times C_\jmath$ and the random variable $\xi_\jmath^\imath$ gets 1 if the entry $M(r_\imath, x)$ is different from this label, meaning that this entry will be counted as a "mistake".

By definition, for a column cluster $\jmath$, adding $\xi_\jmath^\imath$ over the entire rows sample yields:

$$\sum_{i=1}^{t} \xi_\jmath^i = t\, \varphi_S(x, R_1 ... R_k, \jmath)$$

This essentially is the **approximated** cost of placing the column x in the $\jmath$ column cluster, which is how we defined φ_S.

The **probability** that $\xi_\jmath^\imath = 1$ is equal to the number of entries of the column x under the optimal partition $W_1^r ... W_k^r$ which are different than the block's label, if we choose to place x in the $\jmath$'s column block. Recall that this probability (we divide it by m, the number of entries in x), is as defined before $\varphi_S(x, W_1^r ... W_k^r, L, \jmath)$.

Applying the Chernoff additive bound we get,

$$Pr_{r_1 ... r_t} \left[|\varphi_S(x, R_1 ... R_k, L, \jmath) - \varphi_S(x, W_1^r ... W_k^r, L, \jmath)| > \epsilon \right] < \exp(-2\epsilon^2 t)$$

By Markov's inequality (taking $t = \frac{1}{2\epsilon^2} \log \frac{k}{\delta\epsilon}$), for each column block $\jmath$, $1 \leq \jmath \leq \ell$

with probability $1 - \frac{\delta}{\ell}$ over the choice of $r_1 ... r_t$, for all but $\epsilon|C|$ of the columns *equation* 3.1.3 holds.

Therefore with probability of $1 - \delta$, the sample $r_1 ... r_t$ is *good* as required.

The same analysis can be applied for the partition of the rows.

Theorem 3.1.5. *For every every k, ℓ there exists is a polynomial time approximation scheme for the $k \times \ell$ Monochromatic bi-clustering maximization problem.*

Proof. According to claim 1.3.3 there is always a trivial solution to the monochromatic bi-clustering problem whose agreement is at least $|M|/2$. Therefore an additive $\epsilon/2$ approximation translates into a relative $(1 - \epsilon)OPT$ bound on the agreement of the solution. The theorem follows from 3.1.1.

Chapter 4

Additional Research Directions

4.1 The Regularity Lemma

The Regularity Lemma of Szemerdi is a very well known result in extremal graph theory. It was introduced by Szemerdi in 1975 as an auxiliary lemma in the proof of what is now known as Szemerdi's theorem. The lemma asserts that every dense graph can be partitioned into a constant number of regular pairs and a few leftover edges. Therefore each such graph can be well approximated by the union of a constant number of random like bi-partite graphs.

In its original proof, Szemerdi demonstrated the existence of a regular partition, but he did not provide a constructive proof to obtain such a partition. Later, Alon et al.[1] developed the first algorithm to create a regular partition of an arbitrary graph. They showed that a regular partition can be found in polynomial time complexity. Other polynomial-time algorithms for finding a regular

partition have been found since.

The Regularity Lemma has many applications in extremal graph theory, additive combinatorics, and computer science including computer vision and pattern recognition. The Regularity Lemma can be used as a general framework to prove certain approximate qualitative properties of arbitrary graphs. According to the lemma, it suffices to prove the desired statement for the finite set of regular pairs that approximate arbitrary graphs up to a certain level of accuracy. Once the dimensionality of the problem has been reduced to a finite, much smaller size, computer based search techniques can be used to solve the problem. Unfortunately, due to the fact that the number of the regular pairs given by the Regularity Lemma grow extremely fast, this approach is often impractical.

In order to formally describe the Regularity Lemma, we need several technical definitions, which follow the definitions in [1]

Let $G = (V, E)$ denote a graph, and let A, B denote two disjoint subsets of V, let $e(A, B)$ denote the number of edges of G with an end-point in A and an end-point in B. Define the density of the edges between A and B by $d(A, B) = \frac{e(A,B)}{|A||B|}$.

Definition 4.1.1. *For $\epsilon > 0$, the pair (A, B) is called* ϵ-regular *if for every $X \subset A$ and $Y \subset B$ for which $X \geq \epsilon|A|$ and $Y \geq \epsilon|B|$, the inequality:*

$$|d(A, B) - d(X, Y)| < \epsilon \qquad \textit{holds.}$$

A partition of V into pairwise disjoint classes $C_0, C_1, ..., C_k$ is said to be *equitable* if all the classes C_i $(1 \leq i \leq k)$ have the same cardinality. The exceptional set C_0 (which may be empty) has only a technical purpose: it makes it possible that all other classes have exactly the same number of vertices. An equitable partition $C_0, C_1, ..., C_k$ with C_0 being the exceptional set, is called ϵ-regular if

$|C_0| < \epsilon|V|$ and all but at most ϵk^2 of the pairs (C_i, C_j) are ϵ-regular.

Theorem 4.1.2 (Szemerdis Regularity Lemma [15])**.** *For every ϵ and every positive integer t there is an integer $T = T(\epsilon, t)$ such that every graph with $n > T$ vertices has an ϵ-regular partition into $k + 1$ classes, where $1 \leq k \leq T$.*

4.1.1 The Regularity Lemma and monochromatic bi-clustering

The symmetric case of the monochromatic bi-clustering problem can be expressed as a graph problem. The input matrix can be viewed as an adjacency matrix of a graph with the rows and columns corresponding to two identical sets of the graph vertices and the entries with the value '1' represent an edge while the entries with the value '0' represent a missing edge.
Consider the following graph problem: Given a graph $G = (V, E)$, find a partition of the graph vertices into k disjoint subsets $C_1, ..., C_k$ such that the edge density between every pair of subsets (C_i, C_j) is either close to one, or close to zero. Formulated as an optimization problem, we want to find a partition that minimizes the number of **non-edges** between pairs of subsets (C_i, C_j) which have an edge density closer to one, plus the number of **edges** between pairs of subsets having an edge density closer to zero.
Note that this graph optimization problem is equivalent to the symmetric case of the monochromatic bi-clustering.

This observation gives rise to the idea of applying the Regularity Lemma to monochromatic bi-clustering, formulated as a graph optimization problem. The partition guaranteed by the Regularity Lemma can be used to approximate the optimal partition of the matrix with respect to the monochromatic cost function. The intuition behind this idea, is that having the matrix partitioned into regular disjoint subsets is desirable in the monochromatic bi-clustering context. This is

due to the fact that every ϵ-regular pair, (C_i, C_j), derived by the lemma, must have a similar density between any subsets of its vertices. According to definition 4.1.1, for every $X \subset C_i$ and $Y \subset C_j$, such that $X \geq \epsilon|C_i|$ and $Y \geq \epsilon|C_j|$, the density difference is bounded $|d(C_i, C_j) - d(X, Y)| < \epsilon$. This implies that the density of (C_i, C_j), $d(C_i, C_j)$ must be either very high (close to one) or very low (close to zero). In any other case, we can choose the vertex subsets X and Y, such that the edge density between them, $d(X, Y)$ is either very high or very low, thus deviating from $d(C_i, C_j)$, and violating the regularity property.
The Regularity Lemma is stated with respect to regular graphs, but holds just the same for bi-partite graphs. This means that if we can use the Regularity Lemma to approximate monochromatic bi-clustering, the result will hold for the general case as well as the symmetric case.

Although this line of research seems like it could lead to an approximation scheme for the bi-clustering problem, in reality, the Regularity Lemma's preconditions, as well as very large bounds on the number of the regular pairs guaranteed by the lemma, make this approach impractical. For example, in the algorithm suggested by Alon et al. in [1], the lower bound on the number of vertices n, a graph should have in order to comply with the Regularity Lemma is given by the following calculation;
For any $t \in \mathbf{N}_{>0}$ and every $\epsilon > 0$, let b be the least positive integer such that

$$4^b > 600(\tfrac{\epsilon^4}{16})^{-5} \ , \ b > t$$

Let f be the integer valued function defined inductively as,

$$f(0) = b \ , \ f(i+1) = f(i)4^{f(i)}$$

and take

$$T = f(\lceil 10(\tfrac{\epsilon^4}{16})^{-5} \rceil) \text{ and } N = \max\{T4^{2T}, \tfrac{32T}{\epsilon^5}\}.$$

For any graph $G = (V, E)$, having $|V| = n \geq N$, a regular partition into $k + 1$ classes is guaranteed, where $t \leq k \leq T$. To get an intuition on how large these bounds are, take for example $\epsilon = 0.25$, T will then be $f(\lceil 10(\frac{(0.25)^4}{16})^{-5} \rceil) \approx f(10^{20})$. We don't need to determine b explicitly to see that $f(i) \gg i$, thus $T \gg 10^{20}$ and we also know that $N \gg T$.

Other algorithms have been suggested for finding a regular partition of a given graph, but their bounds on the number of vertices and the number of regular pairs, are pretty much along the same lines of [1].

In conclusion, although it seems like the Regularity Lemma has high correlation with the kind of structure we seek to uncover in monochromatic bi-clustering, it is an impractical approach. The typical size of bi-clustering problems is far less than the number required for the Regularity Lemma to hold. Also, usually the desirable partition of the matrix is into a much smaller number of clusters than the one guaranteed by the existing implementations of the lemma.

Chapter 5

Conclusions

5.1 Conclusions and directions for further research

In this work we formalized a specific bi-clustering task, monochromatic bi-clustering, that models problems of deducing group structure of vertices of a bi-partite graph, who's edges capture a pair-wise relation between the vertices. Such problems arise recently in systems biology in the study of epistasis networks where the application of a bi-clustering paradigm seems to lead to biologically meaningful data classification.

The main technical contribution of our work is the analysis of the computational complexity of the resulting optimization problem. We provide a polynomial time approximation scheme with arbitrarily good quality of approximation guarantees. We also show that finding the optimal monochromatic bi-clustering is NP-hard. Such understanding of the computational complexity of optimization is very rare in the domain of bi-clustering algorithms in general, and in particular for biologically relevant bi-clustering.

In spite of having achieved a polynomial time approximation scheme, we are well aware that from a practical point of view there is still quite a way to go before one has an algorithm that is fast enough to handle large data sets under realistic time constrains. We believe that with the growing realization of the relevance of bi-clustering tasks, further algorithmic research will follow and lead to the rigorous formalization of other relevant bi-clustering problems, and in the development of faster practical algorithms for such tasks.

Other interesting directions of future research include the investigation of sample-based bi-clustering and the intriguing issue of the relationship between the structure of the network of objects and other sources of similarity between objects.

Bibliography

[1] Noga Alon, Richard A. Duke, Hanno Lefmann, Vojtech Rodl, and Raphy Yuster. The algorithmic aspects of the regularity lemma. *J. Algorithms*, 16(1):80–109, 1994.

[2] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning, Special Issue on Clustering*, 56:89–113, 2004.

[3] Yizong Cheng and George M. Church. Biclustering of expression data. pages 93–103.

[4] G. M. Church D. Segr, A. DeLuna and R. Kishony. Modular epistasis in yeast metabolism. *Nature Genetics*, 37(1), 2005.

[5] Fabricio O. de Franca, Hamilton M. Ferreira, and Fernando J. Von Zuben. Applying biclustering to perform collaborative filtering. In *ISDA '07: Proceedings of the Seventh International Conference on Intelligent Systems Design and Applications*, pages 421–426, Washington, DC, USA, 2007. IEEE Computer Society.

[6] Inderjit S. Dhillon, Subramanyam Mallela, and Dharmendra S. Modha. Information-theoretic co-clustering. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 89–98, New York, NY, USA, 2003. ACM.

[7] Thomas George and Srujana Merugu. A scalable collaborative filtering framework based on co-clustering. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 625–628, Washington, DC, USA, 2005. IEEE Computer Society.

[8] G. Getz, E. Levine, and E. Domany. Coupled two-way clustering analysis of gene microarray data, 2000.

[9] Ioannis Giotis and Venkatesan Guruswami. Correlation clustering with a fixed number of clusters. *Theory of Computing*, 2(13):249–266, 2006.

[10] O. Goldreich, S. Goldwasser, and D. Ron. Property testing and its connection to learning and approximation. *Journal of the ACM*, 45(4):653–750, July 1998.

[11] Y. Guan H. Cho, I. S. Dhillon and S. Sra. Minimum sum-squared residue co-clustering of gene expression data. In *Proceedings of the Fourth SIAM International Conference on Data Mining*, pages 114–125. SIAM, 2004.

[12] J.A. Hartigan. Direct clustering of a data matrix. *Journal of the American Statistical Association*, 1972.

[13] Y. Kluger, R. Basri, J. T. Chang, and M. Gerstein. Spectral biclustering of microarray data: coclustering genes and conditions. *Genome Res*, 13(4):703–716, April 2003.

[14] A. Tschumi P. Yeh and R. Kishony. Functional classification of drugs by properties of their pair-wise interactions. *Nature Genetics*, 38(489), 2006.

[15] Endre Szemerédi. Regular partitions of graphs. In *Problèmes combinatoires et théorie des graphes (Colloq. Internat. CNRS, Univ. Orsay, Orsay, 1976)*, pages 399–401. CNRS, Paris, 1978.

[16] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data, 2002.