

https://dspace.org/

Convex optimization for rank-sparsity decomposition, with application to the planted quasi-clique problem

Abdulsalaam, Sakirudeen A

2020

https://hdl.handle.net/10539/30982

Abdulsalaam, Sakirudeen Akinkunmi (2020) Convex optimization for rank-sparsity decomposition, with application for the planted quasi-clique problem, University of the Witwatersrand, Johannesburg, <http://hdl.handle.net/10539/30982>

Downloaded from a DSpace repository. https://dspace.org/

THESIS

CONVEX OPTIMIZATION FOR RANK-SPARSITY DECOMPOSITION, WITH APPLICATION TO THE PLANTED QUASI-CLIQUE PROBLEM



WITS
UNIVERSITY

Submitted by

Sakirudeen A. Abdulsalaam

School of Computer Science and Applied Mathematics

In fulfillment of the requirements

For the Degree of Doctor of Philosophy

University of the Witwatersrand,

Johannesburg, South Africa

June 2020

Supervisor: Professor Montaz Ali

Copyright by Sakirudeen A. Abdulsalaam 2020

All Rights Reserved

DECLARATION

I, the undersigned, declare that this thesis is my own original work, except where due references have been made. It is being submitted for the degree of Doctor of Philosophy at the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination at any other university.

Sakirudeen Akinkunmi Abdulsalaam

June 22, 2020.

ABSTRACT

CONVEX OPTIMIZATION FOR RANK-SPARSITY DECOMPOSITION, WITH APPLICATION TO THE PLANTED QUASI-CLIQUE PROBLEM

We consider the rank-sparsity decomposition problem with its application to the planted quasi-clique recovery in this thesis. Given a matrix which is a superposition of a low rank and a sparse matrix, the rank sparsity decomposition problem answers the question, "when is it possible to decompose the matrix into its low rank and sparse components?". The common convex formulation for this problem is to minimize a weighted combination of the nuclear norm and the l_1-norm. To prove optimality of solutions with this formulation, it is customary to derive a bound on the dual matrix which certifies the optimality of the solution. Among the methodological contributions of this thesis is the sharp theoretical bounds obtained for the dual matrix. We have improved the results on low rank matrix decomposition by deriving the bound on our dual matrix, using the matrix $l_{\infty,2}$ norm. Moreover, we established conditions under which recovery is achievable by deriving a dual matrix, certifying the optimality of our solution.

We adapt the convex formulation for the rank-sparstity decomposition to the planted quasi-clique problem. This problem is a generalization of the planted clique problem which is known to be NP-hard. This problem has applications in areas such as community detection, data mining, bioinformatics, and criminal network analysis. We have considered mathematical modelling, theoretical framework, and computational aspects of the problem. We showed that the planted quasi-clique can be recovered using convex programming. We have achieved this by adapting techniques from low rank matrix

decomposition to the planted quasi-clique problem. Our numerical results show that when the input graph contains the desired single large dense subgraph and a moderate number of diversionary vertices and edges, the relaxation is exact.

We have shown, numerically, the superiority of our formulation over the only existing Mixed Integer Programming (MIP) formulations. Further, we present a simplified proof to show that quasi-cliques also posses what is known as quasi-hereditary property. This property can be exploited to develop enumerative algorithm for the problem.

ACKNOWLEDGEMENTS

First and formost, I give all thanks to the Almighty Allah who spared my life till the end of this journey and His countless blessings on me. I say Alhamdulilah.

I would like to thank my advisor, Professor Montaz Ali. I wouldn't be able to complete this thesis without his guidance, encouragement and patience.

I am very grateful to my entire family, especially my parents, my wife; Halima, my kids; Muhammad and Muhsina, and my siblings. You are my pillars of support. I am thankful to my friends like brothers: Abdullateef Moshood, Khalid Sodiq, Ogungbayi Saheed and Abdulmajeed Adiat.

I would like to thank the Nigerian community at Wits University and all my friends around here. You guys made the journey a lot easier. I thank our father at Wits, Professor J. Pedro, who gives me fatherly advice when I am faced with challenges and Dr M.G Rasaq, who is my mentor and an informal co-supervisor. I take this opportunity to express my gratitude to Dr (Mrs) Y.O Aderinto, Dr K.O Babalola, Dr I.A Garba and Professor A. Sirajo.

Lastly, I thank Wits University, TETFund and CSIR for their financial support.

I would like to dedicate this thesis to my entire family.

TABLE OF CONTENTS

DECLARATION ii
ABSTRACT iii
ACKNOWLEDGEMENTS v
LIST OF TABLES ix
LIST OF FIGURES x

Chapter 1 Preamble 1
1.1 Introduction 1
1.2 Background and Motivation 1
1.3 Contributions of the Thesis 6
1.4 Organization of the Thesis 7

Chapter 2 Mathematical Fundamentals 8
2.1 Introduction 8
2.2 Basic Graph Theory 8
2.3 Vector and Matrix norms 10
2.4 Rank and Singular Value Decomposition 15
2.5 Orthogonal Projection 18
2.6 Convex set, hulls, cone and functions 20
2.7 Convex Programming 25
2.8 Semidefinite Programming 28

Chapter 3 Nuclear Norm Relaxation for Rank Minimization Problem 33
3.1 Introduction 33
3.2 The Rank Minimization Problem 33
3.3 Convex Hull of Matrix Rank 36
3.4 Optimality conditions for nuclear norm minimization problem . 40
3.5 Examples of rank minimization problem 41
3.5.1 The cardinality minimization problem 41
3.5.2 The low-rank matrix completion problem 45
3.5.3 The matrix decomposition problem 49
3.6 Conditions for guaranteed success of the nuclear norm heuristic 53
3.6.1 Restricted Isometries 54
3.6.2 Nearly Isometric Random Matrices 55
3.7 Algorithms for Nuclear Norm Minimization 57
3.7.1 Interior Point Method 57
3.7.2 Iterative thresholding 58
3.7.3 Proximal Gradient Algorithm 58
3.7.4 Alternating Direction Method of Multipliers 59

Chapter 4 Clique Relaxations . . . 62
4.1 Introduction . . . 62
4.2 Classes of clique relaxations . . . 62
4.3 Distance and Diameter Bases Relaxations . . . 63
4.4 Degree Based Relaxations . . . 72
4.5 Density Based Relaxations . . . 76

Chapter 5 Quasi-Clique . . . 79
5.1 Introduction . . . 79
5.2 Maximum Quasi-clique Problem . . . 79
5.3 Computational Complexity . . . 80
5.4 Quasi-inheritance . . . 81
5.5 Algorithms for maximum quasi-clique . . . 85
5.6 Analytical Upper Bound for $\omega_\gamma(G)$ . . . 86
5.7 Mathematical Formulations for Quasi-clique . . . 87
5.8 The Planted Quasi-clique Problem . . . 90

Chapter 6 Theoretical Guarantee for Planted Maximum Quasi-Clique Recovery . . . 95
6.1 Introduction . . . 95
6.2 Incoherence Property for Matrix $l_{\infty,2}$ Norm . . . 95
6.3 Background to the Proof . . . 97
6.3.1 The Bernoulli Model and Derandomization . . . 99
6.4 Subgradient Condition for Optimality . . . 100
6.5 Construction of Dual Certificate . . . 103
6.6 Key Lemmas . . . 106
6.7 Proof of Lemma 6.5.1 . . . 108

Chapter 7 Numerical Experiments . . . 119
7.1 Performance Comparison with the Existing Mixed Integer Programming Formulations . . . 119
7.2 Exact recovery from varying γ-clique size . . . 124
7.3 Recovery from varying edge density and random noise . . . 127

Chapter 8 Conclusion . . . 129

LIST OF TABLES

7.1 Relative Errors in edge density of the planted maximum γ-clique compared with recovered γ-clique for a graph with 50 nodes. Result for each γ is average of 10 runs. 121
7.2 Relative Errors in edge density of the planted maximum γ-clique compared with recovered γ-clique for a graph with 100 nodes. Result for each γ is average of 10 runs 122
7.3 Errors in the size of planted maximum γ-clique recovered using different methods for γ ranging from 0.6 to 1. n is the graph size while n_c is the size of the planted γ-clique. The first column under each method contains the average size of recovered quasi-clique using the method while the second contains the relative error of the method. 125

LIST OF FIGURES

1.1 The social network of the US September 11, 2001 terrorist. Source: www.orgnet.com 5

2.1 Reduced SVD of a matrix $M \in \mathbf{R}^{n_1 \times n_2}$ with rank r and $n_2 \leq n_1$ 16
2.2 Full SVD of a matrix $M \in \mathbf{R}^{n_1 \times n_2}$ with rank r and $n_2 \leq n_1$ 17
2.3 A non-orthogonal (Oblique) Projection 18
2.4 An orthogonal Projection 19
2.5 A Convex Cone 23

3.1 Illustration of convex hull of a function. $h(\mathbf{x})$ is the convex hull of $f(\mathbf{x})$ . . 36
3.2 Illustration of sparse solution via l_1-norm minimization 44
3.3 Schematic diagram of low-rank matrix completion. 48
3.4 Schematic diagram of low-rank plus sparse matrix decomposition. 51

4.1 A graph illustrating the difference between a 2-clique and a 2-club 65
4.2 A graph with 2-cliques but no 2-clan 66
4.3 A graph that illustrates k-plexes for $k = 1, 2$ and 4 73
4.4 A graph illustrating a $(k+1)$-plex that is not a k-defective clique 76

5.1 Illustration of lack of hereditary property of γ-clique. 83

7.1 Comparison of the CPU time for the MIP and NNM methods 123
7.2 Exact recovery of varying quasi-clique size from graphs of different sizes . 126
7.3 γ - clique recovery from varying edge density and random noise with $n = 100$ and $n_c = 85$ 128
7.4 γ-clique recovery from varying edge density and random noise with $n = 200$ and $n_c = 170$ 128

Chapter 1

Preamble

1.1 Introduction

In this chapater, we provide motivations for this research. Quasi-clique recovery is a relatively new research area with only a handful of research papers reported in the literature, despite its wide applicabilities. We open with some background to the problem and proceed to highlight our contributions to the body of knowledge. The last section of the chapter contains the organization of the remainder of the thesis.

1.2 Background and Motivation

Various real life problems can be modeled using graphs. For instance, the interaction between protein molecules (protein-protein networks) can be represented with a graph whereby the nodes represent proteins and the edges show interaction [5, 93]. Also, the world-wide web can be viewed as a graph [22, 126, 127, 140]. Each static HTML web page is a node and an edge between any two nodes means that a hyperlink exists betweeen the two pages. Furthermore, a group of people with certain similar traits (or interaction) can be described using graph. Each node represents a person and there is an edge between two people if they share similar trait or they communicate. A very large dense subgraph of any of these graphs can have different real-life implications. A highly connected subgraph of the protein-protein networks implies a protein complex [105]. In the same vein, a dense subgraph of a group of people can mean a group of allies or a group of people of the same origin [71].

A clique induces the densest subgraph of any undirected graph, $G = (V, E)$. $G[V']$, induced by $V' \subseteq V$, forms a clique if its nodes are pairwise adjacent [109]. In this

thesis, clique is used to refer to the subset of vertices or its induced subgraph interchangeably. The Maximum Clique Problem (MCP) is to find the clique with largest cardinality in a given graph [36, 135, 136]. The maximum clique (MC), *the maximum independent set* and the *minimum vertex cover* problem are computationally equivalent [132]. The size of the largest clique in a graph is known as the *clique number*. We denote it by $\omega(G)$. Although MCP is NP-hard [79], it has been well studied due to its wide applications. Verily, cliques possess the ideal properties for cohesiveness [139]. However, the requirement that every pair of nodes are adjacent is too restricting for some real-life applications. This necessitates the emergence of different clique relaxations. Some of these relaxation models, emanating from soical network analysis (SNA), are the k-clique, k-club, and k-plex, see for example, [19, 20]. A density based relaxation known as quasi-clique or γ -clique was introduced by Abello et al. [3, 4]. Although γ-clique is the most recent of clique relaxations, it is one of the most popular due to its suitability for a range of applications [165]. The following are some of the application domains where the quasi-clique model is applicable.

- Community Detection in Social Network Analysis: One of the major areas in which the quasi-clique model is widely applicable is in social networks [88, 139]. In this case, a set of vertices denote the actors in the network while the edges translate to ties among the actors [71]. The actors in the network are people while the relationship, interaction, connection, or association between them is referred to as ties. These ties can be in term of friendship, acquaintances, family or any kind of relationship. In another context, actors can be companies; while ties will represent business dealings between the companies. In this case, a community is characterized by a highly interconnected set of nodes. The density of the network is a measure of the strength of the relationships in the community. Although

cliques are sets of nodes with maximum density, the size of the community that is detected from a network using clique model can be very small.

- Data clustering and data mining: Quasi-clique model can be used for data clustering and data mining. Due to the advent of very high computing power, high volume of data are being generated and stored nowadays. The task of analysing these massive data is becoming challenging [143]. Data clustering can be defined as an unsupervised classification of a dataset into subgroups known as clusters. Each cluster contains objects that are similar to each other but different from members/elements of the other clusters (subgroups). The more intra group (inter group) similarity (dissimilarity) the better the clustering process. Quasi clique has been used for analyzing biological networks [19] and network clustering [166]. Data mining, on the other hand, entails automatic extraction of new information from a given large set of data. There are some other similar terms to data mining; e.g, knowledge mining, knowledge extraction, pattern/data analysis, and knowledge discovery [87]. Graph-based data mining [143] is applicable in finding frequent structure and graph matching in a large domain. Quasi-clique has been used in mining cross-graph in genomic data [142].

- Protein-protein network: Protein complexes are a collection of proteins which interact with one another at the same location and time. Their role in regulation of cellular processes and functions are essential. Protein-protein interaction networks (PPINs) are collections of protein-protein interactions. Important biological information at the cellular or molecular level of interacting proteins can be acquired from PPINs [68]. The protein-protein interaction network can be modelled using undirected graphs [31]. Mining these networks can provide important directions for the study of biological pathways and protein function [36].

It has been shown that quasi-clique is effective in mining protein complexes and other biological structures [31, 91, 114, 142].

- Criminal Network Analysis: Crime analysis involves collection and analysis of crime related data for prediction of crime occurrences based on distribution of existing data [96]. The study of terrorist network, money laundering, drug trafficking are all part of criminal network analysis [17]. Despite the fact that the law enforcement and intelligence agencies have access to large volume of raw data, criminal network analysis is currently neither efficient nor effective. This is because of lack of sophisticated (data mining) tools and techniques that will enhance effective and efficient utilization of the data [172]. Network analysis [71] and data mining/clustering [93, 172] are effective tools for crime analysis and intelligence [20]. Consider the $9/11$ terrorist attack of 2001 in the US for example. Figure 1.1 is the network of the suspected terrorists in the cruel attack. The network shows the links between the attackers. The data used to build this network were publicly available before the horrendous attack but were only collected after the attack. Although this network may neither be complete nor completely accurate, its analysis would have provided valuable insight into the network and activities of the terrorist group. The terrorist network is a dense graph but definitely not a clique. That means using a clique model to solve the problem will not give an acceptable result. However, using a clique relaxation model like γ-clique, where γ can be fine-tuned to the desired density, will give a better result.

For the practical applications enumerated above and others that we could not mention, a really random model may not give a good depiction of average case data. Indeed, a good representation of generic input data would be those containing a particular hidden structure, obscured by random noise [11]. In this thesis, we study the planted

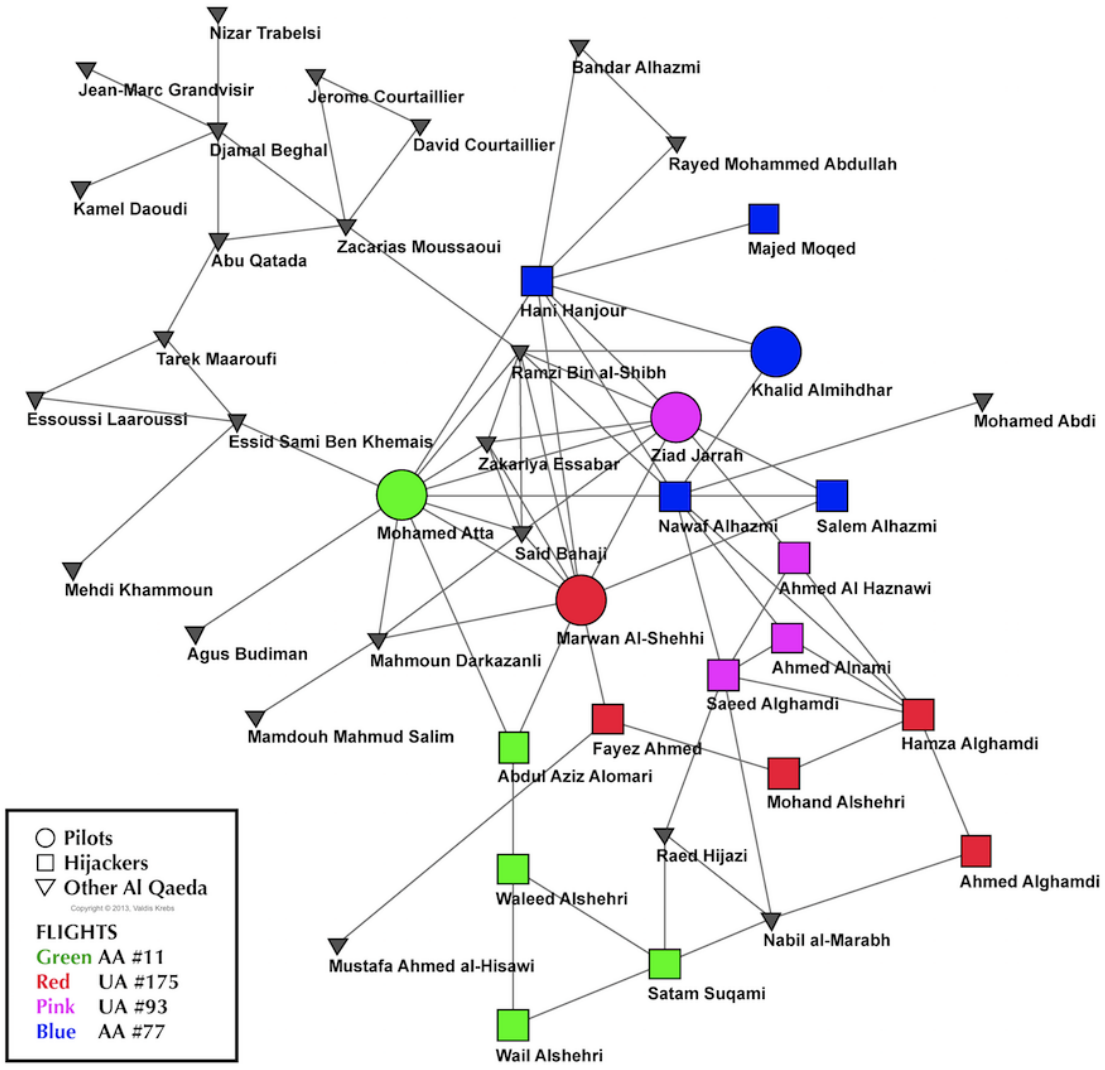


Figure 1.1: The social network of the US September 11, 2001 terrorist.
Source: www.orgnet.com

quasi-clique model. This model is more suitable in representing the real scenarios. The contributions of this thesis are contained in the next section.

1.3 Contributions of the Thesis

The main contribution of this thesis is our proposed mathematical model for maximum planted quasi-clique recovery. Of utmost importance is the theoretical framework for guaranteed recovery that we present. Our contributions can be summarised as follow:

- Clique has hereditary property. This implies that every induced subgraph of a clique is a clique. Many of the clique finding algorithms exploit this property. Unfortunately, this is not true for quasi-clique. Nevertheless, we present a simple and intuitive proof to show that quasi-clique is also endowed with what is known as *quasi-hereditary*.

- We develop a convex program for planted quasi-clique recovery based on matrix decompositon technique. Our convex program is an improvement and a generalization of the convex program in [12] that was proposed for the planted clique problem.

- We obtain a new result on the bound of the dual matrix, certifying the optimality of the solution of our convex program.

- We use this result to establish the condition under which exact recovery of the planted maximum quasi-clique is possible with our convex program. The result obtained from our computational experiments corroborate the theory.

1.4 Organization of the Thesis

The following is the outline of this thesis. We provide definitions and necessary background results on graph theory, linear algebra, convex programming, and probability theory in Chapter 2. A review of the nuclear norm heuristic for matrix rank minimization is contained in Chapter 3. Chapter 4 contains discussion on clique relaxations while Chapter 5 is dedicated to the study of quasi-clique and planted quasi-clique in particular. We present the theoretical framework for planted quasi-clique recovery in chapter 6. Chapter 7 contains the report of the numerical experiments we conducted to support our claim. We summarize the work done in this thesis and give our concluding remarks in Chapter 8.

Part of Chapter 3, 4, 5, 6 and 7 were used in preparing [1] and [2].

Chapter 2

Mathematical Fundamentals

2.1 Introduction

In this chapter, we summarise the classical results on convex and matrix analysis. We begin with some basic definitions from graph theory. We then proceed to the fundamental theories from linear algebra. These tools will be used in our analyses. We conclude the chapter with some established results from convex analysis.

2.2 Basic Graph Theory

A *graph* G is an ordered triple $(V(G), E(G), \phi(G))$ which comprises the vertex or node set (we use these terms interchangeably) $V(G)$, the edge set $E(G)$ and a relation $\phi(G)$. $\phi(G)$ associates two vertices, not necessarily distinct, with each edge. The vertices are known as endpoints. An edge which has identical endpoints is known as a *loop*, while an edge with distinct endpoints is called a *link*. Edges having the same pair of endpoints is called *multiple edges*. A graph without any loop or multiple edges are called *simple graph*. We specify a simple graph by its vertex and edge set only. In addition, we write V instead of $V(G)$ for the vertex set and E for the edge set $E(G)$. All the graphs considered in this thesis are simple graphs. A graph is finite if both its vertex and edge set are finite, otherwise, it is infinite. We also note here that our study only deals with finite graphs. For $i, j \in V$, a *path* between i and j is a sequence of vertices such that each vertex in the sequence is adjacent to the vertex next to it. A closed path is known as a cycle. An n-cycle is a cycle with n vertices. A cycle has a *chord* if it contains a pair of vertices that are adjacent, but not along the cycle. Two nodes are connected if there exists a path between them. G is a connected graph if

all of its nodes are pairwise connected, otherwise, it is disconnected. The distance, $d_G(i,j)$, between two vertices i and j in a graph is the number of edges in a shortest path connecting them. If no such path exists, then $d_G(i,j) = \infty$ [93]. It is possible to have more than one shortest path in one graph. The longest shortest path between any pair of nodes in G is known as the diameter of G and is denoted as $diam(G)$. Indeed, $diam(G) = \max_{i,j \in V}(d_G(i,j))$. i, j are said to be *adjacent* or *neighbours* if there exists an edge $(i,j) \in E$ and such an edge is *incident* to i and j. We denote the set of all neighbours of i in G as $N_G(i)$. The degree of i in G, denoted by $deg_G(i)$, is the number of nodes adjacent to i. In other words, $deg_G(i) = |N_G(i)|$. We denote the minimum degree in G as $\delta(G)$ while the maximum degree is $\Delta(G)$. $H = (V', E')$ is a subgraph of G if $V' \subseteq V$ and $E' \subseteq E$. H is a *spanning subgraph* if $V' = V$ and $E' \subseteq E$. $\rho(G)$ is the edge density of G. It is defined as the ratio of the total number of edges in G to the maximum number of possible edges. In other words, $\rho(G) = |E|/\binom{|V|}{2}$. The vertex connectivity of G, $\kappa(G)$, is the minimum number of vertices to be removed to make G become trivial or disconnected.

For every graph, G, with vertex set V of cardinality n, there is an associated matrix known as the *adjacency* matrix of G. The adjacency matrix A_G of G is an $n \times n$ symmetric matrix such that $A_G(i,j)$ is the number of edges which has $i, j \in V$ as endpoints. The adjacency matrix of a simple graph has entries $0's$ and $1's$ with the diagonal entries always being zeros. There exists, also, the *incidence* matrix M_G of G. M_G is an $n \times m$ matrix such that $M_G(i,j) = 1$ if node i is an endpoint of an edge (i,j), and 0 otherwise. m is the number of edges in G. The complement of G is a graph $\bar{G} = (V, \bar{E})$ such that $(i,j) \in \bar{E}$ if and only if $(i,j) \notin E$. A subset D of V is a *dominating set* if for all $i \in V$, either i is in D or i is adjacent to a vertex in D. A *clique* is a set of pairwise adjacent vertices while an *independent set* (or *stable set*) is a set of vertices that are pairwise non-adjacent.

Proposition 2.2.1. *Let $C \subseteq V$. $G[C]$ is a clique if and only if one of the following conditions hold:*

(a) $d_{G[C]}(i,j) = 1,\ \forall\, i,j \in C$

(b) $diam(G[C]) = 1$

(c) $\{i\}$ *is a dominating set of* $G[C]$*, for every* $i \in C$

(d) $\delta(G[C]) = |C| - 1$

(e) $\rho(G[C]) = 1$

(f) $\kappa(G[C]) = |C| - 1$.

The *chromatic* number, $\chi(G)$, of G is the minimum number of colours required to label the nodes of G such that adjacent nodes would be given different colours. G is k-partite if the nodes of G can be partitioned into k-disjoint sets such that no pair of vertices within the same set are adjacent. If k is equal to 2, we say G is bipartite. G is a *chordal graph* if each cycle with at least four nodes has a chord. For additional background details in graph theory, see [37, 170]

2.3 Vector and Matrix norms

Here and everywhere else, we denote vector spaces with bold capital letters, vectors with bold lower case letters, matrices with capital letters and scalars with lower case letters. Wherever this is not the case, it will be clearly stated. Let $\mathbf{V}$ be a vector space and $\mathbf{R}$ be the set of real numbers. A *vector norm* on $\mathbf{V}$ is a mapping $||.|| : \mathbf{V} \mapsto \mathbf{R}_+$ which satisfies the following conditions: for all $\mathbf{x}, \mathbf{y} \in \mathbf{V}$ and $\alpha \in \mathbf{R}$;

i. $||\mathbf{x}|| \geq 0$ and $||\mathbf{x}|| = 0$ if and only if $\mathbf{x} = 0$ (non-degeneracy property)

ii. $||\alpha\mathbf{x}|| = |\alpha|||\mathbf{x}||$ (linearity)

iii. $||\mathbf{x}+\mathbf{y}|| \leq ||\mathbf{x}|| + ||\mathbf{y}||$ (Triangle inequality)

The space $\mathbf{V}$ can be $\mathbf{R}^n$ or $\mathbf{C}^n$, i.e, real or complex. Examples of norms on $\mathbf{R}^n$ include the l_2-norm or the Euclidean norm denoted by $||.||_2$, defined by $\sqrt{\mathbf{x^T x}} = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle} = \sqrt{\sum_{i=1}^{n} |x_i|^2},\ \forall\, \mathbf{x} \in \mathbf{R}^n$; the l_1-norm $||.||_1$ defined by $\sum_{i=1}^{n} |x_i|,\ \forall\, \mathbf{x} \in \mathbf{R}^n$; the infinity norm or max norm $||.||_\infty$, defined by $\max_{1 \leq i \leq n} |x_i|$. $\mathbf{x}^T$ denotes the transpose of $\mathbf{x}$. Generally, an l_p-norm ($p \geq 1$) is denoted and defined as:

$$||\mathbf{x}||_p = \left(\sum_{i=1}^{n} |x_i|^p \right)^{1/p} \quad \forall \quad \mathbf{x} \in \mathbf{R}^n.$$

Theorem 2.3.1. *All the norms on $\mathbf{R}^n$ are equivalent, i.e, for each pair of norm $||.||_a$ and $||.||_b$ on $\mathbf{R}^n$, there are constants $0 < c_1 \leq c_2 < \infty$ such that:*

$$c_1 ||\boldsymbol{x}||_a \leq ||\boldsymbol{x}||_b \leq c_2 ||\boldsymbol{x}||_a \quad \forall\, \boldsymbol{x} \in \mathbf{R}^n. \tag{2.1}$$

For the l_2-norm, the l_1-norm, and the l_∞-norm the uniform equivalence relations are summarized by:

$$||\mathbf{x}||_2 \leq ||\mathbf{x}||_1 \leq \sqrt{n} ||\mathbf{x}||_2, \tag{2.2}$$

$$||\mathbf{x}||_\infty \leq ||\mathbf{x}||_2 \leq \sqrt{n} ||\mathbf{x}||_\infty, \tag{2.3}$$

$$||\mathbf{x}||_\infty \leq ||\mathbf{x}||_1 \leq n ||\mathbf{x}||_\infty. \tag{2.4}$$

Given any two vectors $\mathbf{x}, \mathbf{y} \in \mathbf{R}^n$, the following holds:

$$|\mathbf{x}^T \mathbf{y}| \leq ||\mathbf{x}||.||\mathbf{y}||. \tag{2.5}$$

The equality holds if and only if $\mathbf{x} = \alpha \mathbf{y}$ for some $\alpha \in \mathbf{R}$. (2.5) is referred to as the *Cauchy-Schwarz inequality*. If $\mathbf{x}$ and $\mathbf{y}$ in an inner product space are orthogonal, i.e

$\mathbf{x}.\mathbf{y} = 0$, then

$$||\mathbf{x} + \mathbf{y}||^2 = ||\mathbf{x}||^2 + ||\mathbf{y}||^2. \tag{2.6}$$

The matrix norm has analogous definition. Indeed, $||.|| : \mathbf{R}^{n_1 \times n_2} \mapsto \mathbf{R}_+$ is a matrix norm if:

(a.) $||X|| \geq 0, \ \forall \ X \in \mathbf{R}^{n_1 \times n_2}$,

(b.) $||\lambda X|| = |\lambda| ||X||, \ \forall \ \lambda \in \mathbf{R}$ and $X \in \mathbf{R}^{n_1 \times n_2}$,

(c.) $||X + Y|| \leq ||X|| + ||Y||, \ \forall \ X, Y \in \mathbf{R}^{n_1 \times n_2}$,

In addition, when X and Y are matrices with appropriate dimensions, some matrix norms satisfy the following additional property:

$$||XY|| \leq ||X|| ||Y||.$$

This property is known as *submultiplicity* property or *consistency condition*.

Common example of matrix norm is the Frobenius norm. The Frobenius norm of $X \in \mathbf{R}^{n_1 \times n_2}$ is defined as:

$$||X||_F = \left(\sum_{i=1}^{n_1} \sum_{j=1}^{n_2} X_{ij}^2 \right)^{1/2} = (tr(X^T X))^{1/2} = \left(\sum_{i=1}^{r} \sigma_i^2 \right)^{1/2}, \tag{2.7}$$

where $trace(A)$ is the trace of a matrix A, A^T denotes the transpose of A; and $\sigma_i, i = 1, \ldots, r$ are the singular values of X (to be discussed later in Section 2.4). For any $A, B \in \mathbf{R}^{n \times n}$,

$$trace(AB) = trace(BA) = trace(A^T B^T) = trace(B^T A^T).$$

The standard Inner Product, $\langle ., . \rangle$, defined on $\mathbf{R}^{n_1 \times n_2}$ is:

$$\langle X, Y \rangle = \sum_i \sum_j X_{ij} Y_{ij} = trace(X^T Y), \tag{2.8}$$

for $X, Y \in \mathbf{R}^{n_1 \times n_2}$.

Other examples of matrix norm include the *sum-absolute-value norm* known as entrywise matrix $||.||_1$ norm and defined as: $||X||_1 = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} |X_{ij}|$, and the *maximum-absolute-value norm* called entrywise matrix $||.||_\infty$ norm defined as:

$$||X||_\infty = \max\{|X_{ij}| : i = 1, \ldots, n_1; j = 1, \ldots n_2\}.$$

An important class of matrix norm is the operator norms or induced norms. Let $||.||_\beta$ be a norm on $\mathbf{R}^{n_2}$. The operator norm of $X \in \mathbf{R}^{n_1 \times n_2}$, induced by $||.||_\beta$ is defined as

$$||X||_\beta = \sup\{||X\mathbf{u}||_\beta : ||\mathbf{u}||_\beta \leq 1\}. \tag{2.9}$$

An alternative definition is given by

$$||X||_\beta = \sup_{\mathbf{u} \neq 0} \frac{||X\mathbf{u}||_\beta}{||\mathbf{u}||_\beta} = \sup_{||\mathbf{u}||_\beta \leq 1} ||X\mathbf{u}||_\beta = \sup_{||\mathbf{u}||_\beta = 1} ||X\mathbf{u}||_\beta. \tag{2.10}$$

If $||.||_\beta$ is the Euclidean norm, then the induced norm of X is the largest singular value. This norm is known as the *spectral norm*. It is denoted by $||X||_2$ and defined as

$$||X||_2 = \sigma_{\max}(X) = (\lambda_{\max}(X^T X))^{1/2}, \tag{2.11}$$

where $\lambda_{\max}$ is the largest eigenvalue of $X^T X$. Unless stated otherwise, we will denote the Euclidean norm by $||\mathbf{u}||$ and the spectral norm by $||X||$.

The norm induced by the l_∞-norm, denoted by $||X||_\infty$, is the maximum row sum of X. The norm is defined as

$$||X||_\infty = \sup\{||X\mathbf{u}||_\infty : ||\mathbf{u}||_\infty \leq 1\}$$
$$= \max_{1\leq i\leq n_1} \sum_{j=1}^{n_2} |X_{ij}|.$$

The last induced norm we are discussing here is the l_1 induced norm, $||X||_1$, defined as:

$$||X||_1 = \max_{1\leq j\leq n_2} \sum_{i=1}^{n_1} |X_{ij}|. \tag{2.12}$$

It is, basically, the maximum column sum of X. The Frobenius norm and induced norms discussed above satisfy the following inequalities, for $X \in \mathbf{R}^{n_1\times n_2}$ (see [83]);

$$||X|| \leq ||X||_F \leq \sqrt{n_2}||X||, \tag{2.13}$$
$$\frac{1}{\sqrt{n_2}}||X||_\infty \leq ||X|| \leq \sqrt{n_1}||X||_\infty, \tag{2.14}$$
$$\frac{1}{\sqrt{n_1}}||X||_1 \leq ||X|| \leq \sqrt{n_2}||X||_1, \tag{2.15}$$
$$||X|| \leq \sqrt{||X||_1||X||_\infty}. \tag{2.16}$$

Given any norm, $||.||$, there exists a dual norm, $||.||_d$, defined as

$$||X||_d = \sup_Y\{\langle X, Y\rangle : ||Y|| \leq 1\}. \tag{2.17}$$

In addition, the dual of a dual norm, $||(||.||_d)||_d$, is the original norm, $||.||$. For vectors in $\mathbf{R}^n$, the dual of l_p-norm, $p \in (1,\infty)$, is the l_q-norm, with $\frac{1}{p}+\frac{1}{q} = 1$. The Euclidean norm is self dual, the dual of l_∞-norm is l_1-norm while the dual of l_1-norm is l_∞-norm. This concept extends to the matrix norms defined earlier. Equivalently, the dual of Frobenius norm is Frobenius norm. However, the dual associated to the spectral norm is defined as

$$||X||_* = \sup_Y\{trace(X^TY) : ||Y|| \leq 1\}. \tag{2.18}$$

This norm is the sum of the singular values:

$$||X||_* = \sigma_1(X) + \sigma_2(X) + \ldots + \sigma_r(X) = trace\left((X^T X)^{1/2}\right),$$

where r is the rank of X. The norm is mainly called the Nuclear norm. Nevertheless, is known by some other names like Schatten 1-norm, the Ky Fan r-norm, and the trace class norm [147]. The induced norms are related with the following inequality

$$||X|| \leq ||X||_F \leq ||X||_* \leq \sqrt{r}||X||_F \leq r||X||. \tag{2.19}$$

2.4 Rank and Singular Value Decomposition

One of the important tools used in solving matrix rank problem is the Singular Value Decomposition (SVD). Let $M \in \mathbf{R}^{n_1 \times n_2}$ with $rank(M) = r$. Then, M can be factorized as

$$M = \hat{U}\hat{\Sigma}\hat{V}^T, \tag{2.20}$$

where $\hat{U} \in \mathbf{R}^{n_1 \times r}, \hat{V} \in \mathbf{R}^{n_2 \times r}, \hat{\Sigma}$ is a diagonal matrix with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r$ on the diagonal. The $\sigma_i, i = 1, \ldots, r$ are known as the singular values of M while the columns of $\hat{U}$ and $\hat{V}$ are the singular vectors. The matrices $\hat{U}$ and $\hat{V}$ satisfy:

$$\hat{U}^T\hat{U} = \hat{V}^T\hat{V} = I.$$

Any square matrix, A, such that $A^T A = AA^T = I$ is said to be Orthogonal. We will discuss more on orthogonality in the next section. The factorization of M into SVD can alternatively be written as

$$M = \sum_{i=1}^{r} \sigma_i u_i v_i^T, \tag{2.21}$$

where u_i, v_i is the $i-th$ column of $\hat{U}, \hat{V}$ respectively; and σ_i is the $i-th$ singular value. The columns of $\hat{U}$ and $\hat{V}$ are called the left and right singular vectors, respectively. Furthermore, for a symmetric matrix, the left and the right singular vectors are the same. The factorization in (2.20) and (2.21) is called *reduced singular value decomposition* (see Figure 2.1).

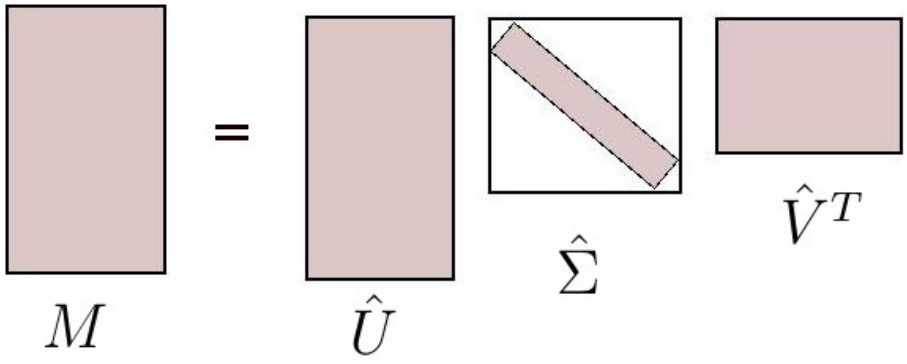


Figure 2.1: Reduced SVD of a matrix $M \in \mathbf{R}^{n_1 \times n_2}$ with rank r and $n_2 \leq n_1$

The SVD used in most applications is the reduced form. However, that is not the initial formulation of SVD. Observe that the columns of $\hat{U}$ are r orthonormal vectors in n_1-dimensional space. Unless $n_1 = r$, they do not form a basis of $\mathbf{R}^{n_1}$. For them to form a basis, we can adjoin an additional $n_1 - r$ orthonormal columns to $\hat{U}$ to get $U \in \mathbf{R}^{n_1 \times n_1}$. Similarly, $\hat{V}$ will be adjoined with $n_2 - r$ orthonormal vectors while the last $n_1 - r$ rows and $n_2 - r$ columns of Σ are padded with zeros. The decomposition hence becomes

$$M = U\Sigma U^T, \tag{2.22}$$

where $U \in \mathbf{R}^{n_1 \times n_1}, V \in \mathbf{R}^{n_2 \times n_2}$ and $\Sigma \in \mathbf{R}^{n_1 \times n_2}$. (2.22) is known as the *full singular value decomposition*. Figure 2.2 is a schematic diagram of full SVD. The dashed lines indicate the "silent" rows or columns of U, V and Σ that are discarded in (2.20).

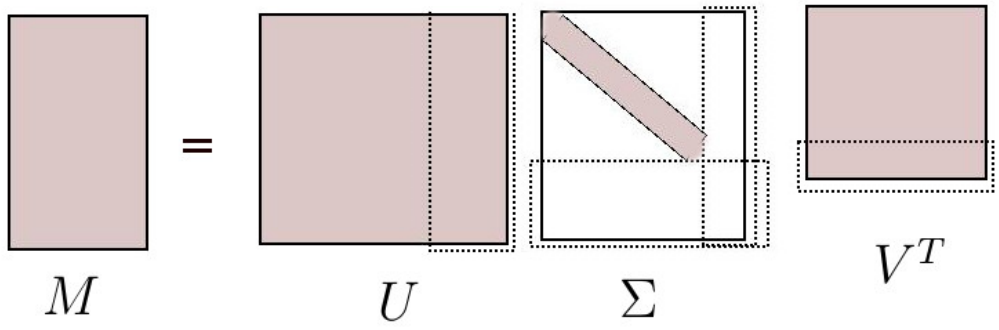


Figure 2.2: Full SVD of a matrix $M \in \mathbf{R}^{n_1 \times n_2}$ with rank r and $n_2 \leq n_1$

The rank of a matrix, M, denoted by $rank(M)$, is the number of its non-zero singular values. The rank is also equivalent to the number of linearly independent row(s) or column(s) of M. On the other hand, the null-space of M is the set of solutions to the homogeneous equation $M\mathbf{x} = \mathbf{0}$. Therefore,

$$nullspace(M) = \{\mathbf{x} \in \mathbf{R}^{n_2} : M\mathbf{x} = \mathbf{0}\}.$$

The dimension of null-space of M is known as the nullity of M and we denote it by $nullity(M)$. The following theorem states the relationship between the dimension, the rank and the nullity of M. In fact, it is known as the fundamental theorem of Linear Algebra.

Theorem 2.4.1. *For any matrix $M \in \mathbf{R}^{n_1 \times n_2}$ with $n_1 \geq n_2$,*

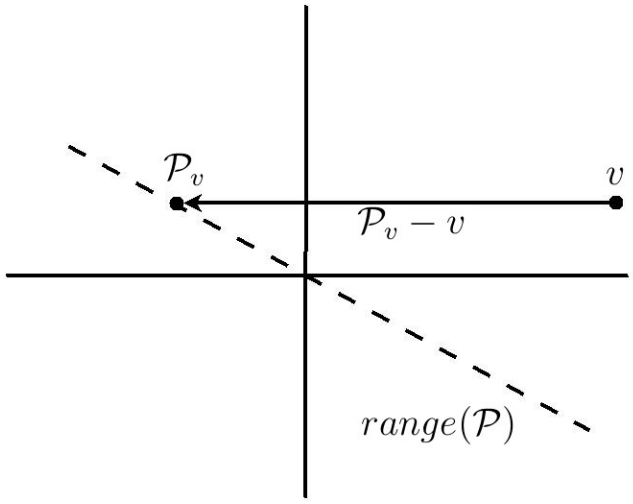


Figure 2.3: A non-orthogonal (Oblique) Projection

$$n_2 = rank(M) + nullity(M)$$

2.5 Orthogonal Projection

A projector, $\mathcal{P}$, is a square matrix which satisfies

$$\mathcal{P}^2 = \mathcal{P}. \tag{2.23}$$

In other words, $\mathcal{P}$ is *idempotent*. The projector can be orthogonal or non-orthogonal. A non-orthogonal projector can be called *oblique* projector [160]. One can see a projector as a situation where a light is shone on the subspace $range(\mathcal{P})$ from the right direction, then the shadow projected by a vector, $\mathbf{v}$, will be $\mathcal{P}\mathbf{v}$. One can easily observe that if $\mathbf{v}$ belongs to $range(\mathcal{P})$, then $\mathbf{v}$ will lie exactly on its own shadow. Mathematically, $\mathbf{v} \in range(\mathcal{P})$ implies that there exists some $\mathbf{u}$, such that $\mathbf{v} = \mathcal{P}\mathbf{u}$. Hence,

$$\mathcal{P}\mathbf{v} = \mathcal{P}(\mathcal{P}\mathbf{u}) = \mathcal{P}^2\mathbf{u} = \mathcal{P}\mathbf{u} = \mathbf{v}. \tag{2.24}$$

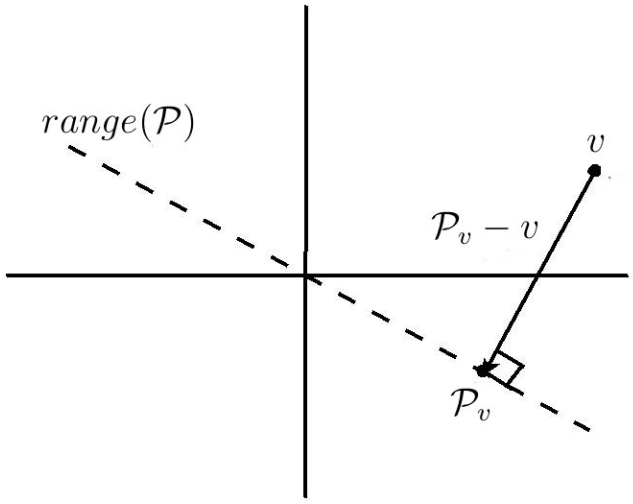


Figure 2.4: An orthogonal Projection

One can draw a line from $\mathbf{v}$ to $\mathcal{P}\mathbf{v}$ (see Figure 2.3). Projecting along the line, we have

$$\mathcal{P}(\mathcal{P}\mathbf{v} - \mathbf{v}) = \mathcal{P}^2\mathbf{v} - \mathcal{P}\mathbf{v} = 0. \tag{2.25}$$

This implies that $\mathcal{P}\mathbf{v} - \mathbf{v} \in null(\mathcal{P})$. If $\mathcal{P}$ is a projector, then $\mathcal{I} - \mathcal{P}$ is also a projector because

$$(\mathcal{I} - \mathcal{P})^2 = \mathcal{I}^2 - 2\mathcal{P}^2 + \mathcal{P}^2 = \mathcal{I} - \mathcal{P}. \tag{2.26}$$

This is called the *complementary* projector. $\mathcal{I} - \mathcal{P}$ projects onto to the nullspace of $\mathcal{P}$. For any projector, $\mathcal{P}$, the following holds:

$$range(\mathcal{I} - \mathcal{P}) = null(\mathcal{P}). \tag{2.27}$$

$$null(\mathcal{I} - \mathcal{P}) = range(\mathcal{P}). \tag{2.28}$$

$$range(\mathcal{P}) \cap null(\mathcal{P}) = \{0\}. \tag{2.29}$$

$\mathcal{P}$ is an orthogonal projector (Figure 2.4) if $\mathcal{P}$ satisfies (2.23) and $\mathcal{P} = \mathcal{P}^T$. We denote an orthogonal projector by $\mathcal{P}^{\perp}$. Let $V \subseteq \mathbf{R}^n$ be a subspace. For any $\mathbf{y} \in \mathbf{R}^n, \mathcal{P}\mathbf{y} \in V$ and $(\mathcal{I} - \mathcal{P})\mathbf{y} \in V^{\perp}$. If $\mathcal{P}_1$ and $\mathcal{P}_2$ are each orthogonal projectors, then for any $\mathbf{z} \in \mathbf{R}^n$ we have

$$||(\mathcal{P}_1 - \mathcal{P}_2)\mathbf{z}||^2 = (\mathcal{P}_1\mathbf{z})^T(\mathcal{I} - \mathcal{P}_2)\mathbf{z} + (\mathcal{P}_2\mathbf{z})^T(\mathcal{I} - \mathcal{P}_1)\mathbf{z}. \tag{2.30}$$

Furthermore, if $range(\mathcal{P}_1) = range(\mathcal{P}_2) = V$, then the right-hand side of Equation (2.30) is equal to zero. This proves the uniqueness of orthogonal projection of a subspace. Suppose the columns of $U = \{\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_k\}$ are an orthonormal basis for a subspace V, then $\mathcal{P} = UU^T$ is the unique orthogonal projection onto V.

2.6 Convex set, hulls, cone and functions

Let $F \subseteq \mathbf{R}^n$ be a set. F is an affine set if the line passing through any two different points in F lies in F, i.e, F is an affine set if for any $\mathbf{x}_1, \mathbf{x}_2 \in F$ and $\alpha \in \mathbf{R}$; $\alpha\mathbf{x}_1 + (1-\alpha)\mathbf{x}_2 \in F$. This means that F contains the linear combination of any pair of points in it, provided the coefficients of the linear combination sum to one. This idea can be extended to more than two points. A point of the form $\alpha_1\mathbf{x}_1 + \ldots + \alpha_k\mathbf{x}_k$, where $\alpha_1 + \ldots + \alpha_k = 1$, is an affine combination of the points $\mathbf{x}_1, \ldots, \mathbf{x}_k$. The set of all affine combination of points in F is called the *affine hull* of F, and it is denoted by $\mathbf{aff}\, F$. Mathematically,

$$\mathbf{aff}\, F = \{\alpha_1\mathbf{x}_1 + \ldots + \alpha_k\mathbf{x}_k : \mathbf{x}_1, \ldots, \mathbf{x}_k \in F, \alpha_1 + \ldots + \alpha_k = 1\}. \tag{2.31}$$

The affine hull is the smallest affine set containing F. Closely related to affine set, is the convex set. F is a convex set if

$$\alpha \mathbf{x}_1 + (1-\alpha)\mathbf{x}_2 \in F, \ \forall \, \mathbf{x}_1, \mathbf{x}_2 \in F, \forall \alpha \in [0,1];$$

that is the line segment between any two points in F lies in F. A point of the form $\alpha_1 \mathbf{x}_1 + \ldots + \alpha_k \mathbf{x}_k$, where the $\alpha_1 + \ldots + \alpha_k = 1$, and $\alpha_i \geq 0, i = 1, \ldots, k$ is known as the convex combination of the points $\mathbf{x}_1, \ldots, \mathbf{x}_k$. The *convex hull* of F is denoted by $\mathbf{conv}F$ and is defined as

$$\mathbf{conv}F = \left\{ \alpha_1 \mathbf{x}_1 + \ldots + \alpha_k \mathbf{x}_k : \mathbf{x}_i \in F, \sum_{i=1}^{k} \alpha_i = 1, \alpha_i \geq 0, i = 1, \ldots, k \right\}. \quad (2.32)$$

It is the set of all convex combination of points in F. Obviously, a convex hull is always convex. As it is the case for affine set, the convex hull is the smallest convex set that contains F. The following proposition states the operations that preserves convexity of a convex set.

Proposition 2.6.1 ([29], Proposition 1.1.1). *(a.) The intersection $\cap_{i \in I} F_i$ of any collection of convex sets $\{F_i : i \in I\}$ is convex.*

(b.) For any two convex sets F_1 and F_2, the vector sum $F_1 + F_2$ is convex.

(c.) For a convex set F and a scalar $\alpha, \alpha F$ is convex. In addition, if F is convex and α_1, α_2 are positive scalars then

$$(\alpha_1 + \alpha_2)F = \alpha_1 F + \alpha_2 F.$$

(d.) The interior and the closure of a convex set are convex. A point $x \in F$ is an interior point of F if there is a small ball centered at x that lies entirely in F. The set of all interior points of F is the interior of F. Also, the closure of a set F is the intersection of all the closed sets containing F.

(e.) The image and the inverse image of a convex set under an affine function is convex.

By convention, an empty set is a convex set. In addition, a singleton, the space $\mathbf{R}^n$, any subspace of $\mathbf{R}^n$ are common examples of convex set. We note here, however, that there are some special examples of convex sets. One of these sets, the *semidefinite cone*, is very important to our study. Hence, special attention is given to this set later in Section 2.8. We briefly mention the other sets in this category here. A *hyperplane* is a set of the form

$$\{\mathbf{y} : \mathbf{a}^T\mathbf{y} = b\},$$

where $\mathbf{a} \in \mathbf{R}^n$ is a nonzero vector and $b \in \mathbf{R}$. Analytically, it is the solution set of a nontrivial linear equation. A *closed halfspace* is a set of the form

$$\{\mathbf{y} : \mathbf{a}^T\mathbf{y} \leq b\},$$

where $\mathbf{a} \in \mathbf{R}^n$ is a nonzero vector and $b \in \mathbf{R}$. Simply put, it is the solution set of one nontrivial linear inequality. A hyperplane divides $\mathbf{R}^n$ into two halfspaces: open and closed halfspaces. The set $\{\mathbf{x} : \mathbf{a}^T\mathbf{x} < b\}$ is the interior of the halfspace and it is known as an open halfspace. The hyperplane $\{\mathbf{x} : \mathbf{a}^T\mathbf{x} = b\}$ is the boundary of the halfspace. We have the following theorem.

Theorem 2.6.1 ([150], Theorem 11.5)**.** *Every closed convex set is the intersection of the closed half-spaces which contain it.*

A *polyhedron* is the solution set of finite number of linear equalities and inequalities. That is

$$P = \{\mathbf{x} : \mathbf{a}_i^T\mathbf{x} \leq b_i, i = 1, \ldots, p; \mathbf{c}_i^T\mathbf{x} = d_i, i = 1, \ldots, q\}.$$

Thus, a polyhedron is an intersection of finite number of hyperplanes and halfspaces. A polyhedron that is bounded is sometimes called a *polytope*. Although, this is not a general convention. A set C is a cone if for every $\mathbf{x} \in C$ and $\alpha \geq 0, \alpha\mathbf{x} \in C$. C is a convex cone (see Figure 2.5) if it is a cone and convex. This implies that $\forall \mathbf{x}_1, \mathbf{x}_2 \in C$ and $\alpha_1, \alpha_2 \geq 0$,

$$\alpha_1\mathbf{x}_1 + \alpha_2\mathbf{x}_2 \in C.$$

Any point of the form $\alpha_1\mathbf{x}_1 + \ldots + \alpha_k\mathbf{x}_k$ with $\alpha_i \geq 0, i = 1, \ldots, k$ is called the conic combination of $\mathbf{x}_1, \ldots, \mathbf{x}_k$. C is a convex cone if and only if it contains the conic combinations of all it elements. The *conic hull* of C is the set of all the conic combinations of the entire points in C.

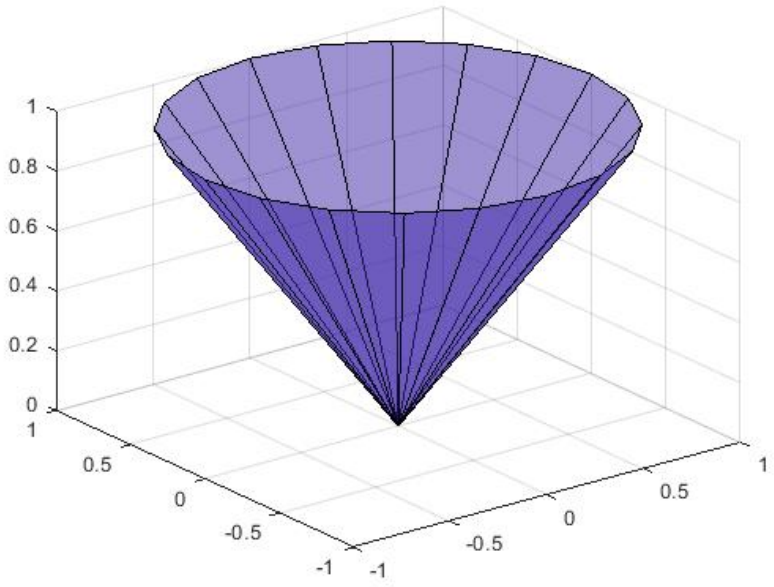


Figure 2.5: A Convex Cone

Convex functions have similar definition. Let F be a convex subset of $\mathbf{R}^n$. A real valued function $f : F \mapsto \mathbf{R}$ is said to be convex on F if for all $\mathbf{x}_1, \mathbf{x}_2 \in F$ and for any

scalar $\alpha \in [0,1]$, we have that

$$f(\alpha \mathbf{x}_1 + (1-\alpha)\mathbf{x}_2) \leq \alpha f(\mathbf{x}_1) + (1-\alpha) f(\mathbf{x}_2). \tag{2.33}$$

f is strictly convex if $\forall\, \mathbf{x}_1, \mathbf{x}_2 \in F, \mathbf{x}_1 \neq \mathbf{x}_2$ and $\alpha \in (0,1)$, the inequality in (2.33) is strict. f is concave if $(-f)$ is convex and strictly concave if $(-f)$ is strictly convex. An example of convex functions is a norm $||.||$, since by triangle inequality, we have

$$||\alpha \mathbf{x}_1 + (1-\alpha)\mathbf{x}_2|| \leq ||\alpha \mathbf{x}_1|| + ||(1-\alpha)\mathbf{x}_2||,$$

for any $\mathbf{x}_1, \mathbf{x}_2 \in \mathbf{R}^n$ and $\alpha \in [0,1]$. Given a scalar $\alpha \in \mathbf{R}$, the sublevel set of f is the set $\{\mathbf{x} \in F : f(\mathbf{x}) \leq \alpha\}$. For any value of α, the sublevel sets of a convex function is convex, but the converse is not true. That is, the sublevel sets of f can be convex whereas f is not convex. If f is concave, then the superlevel set is defined by $\{\mathbf{x} \in F : f(\mathbf{x}) \geq \alpha\}$. The superlevel set is also convex. The *epigraph* of $f : \mathbf{R}^n \mapsto \mathbf{R}$, denoted by $\mathbf{epi} f$, is a subset of $\mathbf{R}^{n+1}$ and is defined as

$$\mathbf{epi} f = \{(\mathbf{x}, w) : \mathbf{x} \in \mathbf{R}^n, w \in \mathbf{R}, f(\mathbf{x}) \leq w\}. \tag{2.34}$$

The epigraph is the link between convex sets and convex functions [41]. $f : \mathbf{R}^n \mapsto \mathbf{R}$ is convex if an only if $\mathbf{epi} f$ is convex. On the other hand, f is concave if and only if its *hypograph* is a convex set. The hypograph of f is denoted by $\mathbf{hypo} f$ and is defined as

$$\mathbf{hypo} f = \{(\mathbf{x}, w) : \mathbf{x} \in \mathbf{R}^n, w \in \mathbf{R}, f(\mathbf{x}) \geq w\}. \tag{2.35}$$

The convex *envelope* of f, denoted by $\mathbf{env} f$, is the supremum of all convex functions g such that $g(\mathbf{x}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in F$. For a convex function $f : F \mapsto \mathbf{R}$, $g \in \mathbf{R}^n$ is a

subgradient of f at $\mathbf{x}_0 \in F$ if

$$f(\mathbf{x}) \geq f(\mathbf{x}_0) + g^T(\mathbf{x} - \mathbf{x}_0), \quad \forall\, \mathbf{x} \in F. \tag{2.36}$$

The set of subgradients of f at $\mathbf{x}_0$ is known as the *subdifferential* of f at $\mathbf{x}_0$ and it is denoted by $\partial f(\mathbf{x}_0)$. Subgradients can be considered as an extension of the concept of gradient to non-smooth (non-differentiable) functions. In fact, if f is continuously differentiable, the subdifferential of f at $\mathbf{x}_0$ is exactly the gradient of f at $\mathbf{x}_0$, i.e, $\partial f(\mathbf{x}_0) = \{\nabla f(\mathbf{x}_0)\}$, where $\{\nabla f(\mathbf{x}_0)\}$ is the vector of all partial derivatives of f at $\mathbf{x}_0$.

2.7 Convex Programming

A convex optimization program is a problem of the form

$$\text{minimize} f(\mathbf{x}) \tag{2.37a}$$

$$\text{subject to } \mathbf{x} \in X, \tag{2.37b}$$

where the objective function $f : X \mapsto (-\infty, +\infty]$ is a convex function and $X \subseteq \mathbf{R}^n$ is a non-empty set. Convex programming requires minimizing a given convex function over a given convex set. When the convex set X is the intersection of a convex cone and an affine subspace, we have a conic program. One important example of conic programming is the *Semidefinite programming* (SDP). Detail discussion about SDP is given in Section 2.8. Any $\mathbf{x} \in X \cap dom(f)$ is called a feasible solution of (2.37). Problem (2.37) is feasible if there exists at least one feasible solution, i.e, $X \cap dom(f) \neq \emptyset$; otherwise, the problem is infeasible. $\hat{\mathbf{x}}$ is a minimum of f over X if $\hat{\mathbf{x}} \in X \cap dom(f)$ and $f(\hat{\mathbf{x}}) = \inf_{\mathbf{x} \in X} f(\mathbf{x})$. In this case, $\hat{\mathbf{x}}$ is called the minimizer of f over X. Alternatively, we say f attains a minimum over X at $\hat{\mathbf{x}}$ and write

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in X} f(\mathbf{x}). \tag{2.38}$$

If $\hat{\mathbf{x}}$ is the unique minimizer over X, then we write

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in X} f(\mathbf{x}). \tag{2.39}$$

The maximum point also has similar terminologies, i.e, $\hat{\mathbf{x}} \in X : f(\hat{\mathbf{x}}) = \sup_{\mathbf{x}\in X} f(\mathbf{x})$ is the maximum point of f over X if $\hat{\mathbf{x}}$ is a minimum of $(-f)$ over X. This is indicated by writing

$$\hat{\mathbf{x}} \in \arg\max_{\mathbf{x}\in X} f(\mathbf{x}). \tag{2.40}$$

And similarly, if $\hat{\mathbf{x}}$ is the unique maximum point over X, then

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in X} f(\mathbf{x}). \tag{2.41}$$

Suppose X is a subspace of $\mathbf{R}^n$, then X is closed and convex. Problem (2.37) can also be written in the following form (see [30, 41])

$$\text{minimize } f(\mathbf{x}) \tag{2.42a}$$

$$\text{subject to } g_i(\mathbf{x}) \leq 0, \quad i = 1, \ldots, p; \tag{2.42b}$$

$$h_j(\mathbf{x}) = 0, \quad j = 1, \ldots, q; \tag{2.42c}$$

where f and $g_i, i = 1, \ldots, p$, are convex functions and $h_j, j = 1, \ldots, q$ are affine. We define the Lagrangian $\mathcal{L} : F \times \mathbf{R}^q \times \mathbf{R}^q \mapsto \mathbf{R}$ associated with (2.42) as

$$\mathcal{L}(\mathbf{x}, \lambda, \nu) = f(\mathbf{x}) + \sum_{i=1}^{p} \lambda_i g_i(\mathbf{x}) + \sum_{j=1}^{q} \nu_j h_j(\mathbf{x}), \quad \forall \quad \mathbf{x} \in F,\ \lambda \in \mathbf{R}^p,\ \nu \in \mathbf{R}^q, \lambda \geq 0. \tag{2.43}$$

Hence, (2.42) can be written as the primal problem

$$prob_p = \inf_{\mathbf{x} \in F} \sup_{\lambda \in \mathbf{R}_+^p, \nu \in \mathbf{R}^q} \mathcal{L}(\mathbf{x}, \lambda, \nu), \tag{2.44}$$

where $\mathbf{R}_+$ is the set of positive real numbers. We switch the order of the supremum and infimum in (2.44) to get the dual problem

$$prob_d = \sup_{\lambda \in \mathbf{R}_+^p, \nu \in \mathbf{R}^q} \inf_{\mathbf{x} \in F} \mathcal{L}(\mathbf{x}, \lambda, \nu). \tag{2.45}$$

We denote the optimal value of the Lagrange primal and dual problem as $prob_{p^*}$ and $prob_{d^*}$ respectively. $prob_{d^*}$ is the best lower bound on $prob_{p^*}$ that can be obtained from the Lagrange dual function. Indeed, we have the following important inequality

$$prob_{d^*} \leq prob_{p^*}. \tag{2.46}$$

The inequality (2.46) holds even if the original function is non-convex [41]. The property stated in Equation (2.46) is known as the *weak duality*. It is possible that the dual value is strictly less than the primal value, i.e,

$$prob_{d^*} < prob_{p^*}. \tag{2.47}$$

In this case, we say duality gap exists between the primal and the dual problem. If the equality

$$prob_{d^*} = prob_{p^*} \tag{2.48}$$

holds, then there is no duality gap. Then, we say strong duality holds. The best bound which can be obtained from the dual function of the Lagrange is tight in this case. There are several results, establishing conditions under which strong duality holds.

The conditions are known as constraint qualification. A simple constraint qualification is the Slater's condition. The convex program (2.42) satisfies the Slater's constraint qualification if there exists $\mathbf{x}_0 \in dom(f)$ such that $g_i(\mathbf{x}_0) < 0, \forall i = 1, \ldots, p$, and $h_j(\mathbf{x}_0) = 0, \forall j = 1, \ldots, q$. We have the following theorem.

Theorem 2.7.1 ([38], Theorem 4.3.7)**.** *If the Slater's condition holds for the primal problem* (2.42)*, then the values of the primal and the dual are equal, and the dual value is attained if finite.*

We will now proceed to discuss an important class of convex programming known as semidefinite programming. This class of optimization problem is fundamental to this thesis.

2.8 Semidefinite Programming

In this section, we will introduce the standard formulation of a primal Semidefinite program (SDP) together with its dual. We review some important properties of the Semidefinite cone and program.

A matrix $M \in \mathbf{R}^{n \times n}$ is symmetric if $M = M^T$. Every symmetric matrix M can be factorized into

$$M = QDQ^T, \tag{2.49}$$

where $Q \in \mathbf{R}^{n \times n}$ is an orthogonal matrix and D is an $n \times n$ diagonal matrix of the eigenvalues of M arranged in non-increasing order. The columns of Q are the eigenvectors of M. The factorization in (2.49) is known as spectral decomposition of M. The set of eigenvalue of M is called its spectrum. Recall that λ is an eigenvalue of M, with corresponding eigenvector $\mathbf{v}$, if there exists a nonzero vector $\mathbf{v} \in \mathbf{R}^n$, such that $M\mathbf{v} = \lambda\mathbf{v}$. Alternatively, λ is an eigenvalue of M if it is a root of the characteristic polynomial $p(\lambda) = |M - \lambda I|$. A symmetric matrix M is said to be positive semidef-

inite (PSD) if for all $\mathbf{v} \in \mathbf{R}^n, \mathbf{v}^T M\mathbf{v} \geq 0$. If M is a symmetric positive semidefinite matrix, then all its eigenvalues are real and non-negative. We denote the space of $n \times n$ symmetric matrices as $\mathbf{S}^n$ and symmetric positive semidefinite matrices as $\mathbf{S}^n_+$. Note that, for symmetric matrices, the spectral decomposition is equivalent to the singular value decomposition with the singular values being the absolute value of the eigenvalues. Further, $M \in \mathbf{S}^n$ is known to be positive definite if for all nonzero vector, $\mathbf{v} \in \mathbf{R}^n, \mathbf{v}^T M\mathbf{v} > 0$. In other words, we say a symmetric matrix is positive definite if all of its eigenvalues are strictly positive. In like manner, we denote the space of $n \times n$ positive definite (PD) matrices as $\mathbf{S}^n_{++}$. Let $M_1, M_2 \in \mathbf{S}^n$. We denote symmetric PSD matrix M_1 by $M_1 \succeq 0$ and write $M_1 \succeq M_2$ if $M_1 - M_2 \succeq 0$. Symmetric positive definite matrix M_1 is denoted by $M_1 \succ 0$.

$\mathbf{S}^n_+ = \{M \in \mathbf{S}^n | M \succeq 0\}$ is closed under addition. Moreover, consider $M_1, M_2 \in \mathbf{S}^n_+$ and scalars $\alpha, \beta \geq 0$. For any $\mathbf{v} \in \mathbf{R}^n$, we have

$$\mathbf{v}^T(\alpha M_1 + \beta M_2)\mathbf{v} = \alpha \mathbf{v}^T M_1 \mathbf{v} + \beta \mathbf{v}^T M_2 \mathbf{v} \geq 0,$$

where $\alpha M_1 + \beta M_2 \in \mathbf{S}^n_+$. Hence, $\mathbf{S}^n_+$ is a close convex cone in $\mathbf{R}^{n \times n}$ with dimension $n \times (n+1)/2$. $\mathbf{S}^n_+$ is known as the semidefinite cone.

A semidefinite program (SDP) is an optimization problem defined as follows

$$\text{minimize } \langle C, X \rangle, \tag{2.50a}$$

$$\text{subject to } \langle A_i, X \rangle = \mathbf{b}_i, i = 1, \ldots, p, \tag{2.50b}$$

$$X \succeq 0, \tag{2.50c}$$

where the matrices C and $A_i, i = 1, \ldots, p$ are given symmetric matrices, the vector $\mathbf{b} \in \mathbf{R}^p$ is also given while X is the variable to be optimized. Let $\mathcal{A} : \mathbf{S}^n \mapsto \mathbf{R}^p$ be a given linear map, then (2.50) can be written as

$$\text{minimize } \langle C, X \rangle, \tag{2.51a}$$

$$\text{subject to } \mathcal{A}(X) = \mathbf{b}, \tag{2.51b}$$

$$X \succeq 0. \tag{2.51c}$$

Since the objective function $\langle C, . \rangle$ and the set $\{X | \mathcal{A}(X) = \mathbf{b}\}, \mathbf{S}^n$ are convex, problem (2.51) is a convex program. Therefore, we can define a dual problem and optimality condition for (2.51) as we did for (2.42). To define the dual of (2.51), we need the adjoint operator to $\mathcal{A}$. The adjoint of $\mathcal{A}$ is defined as $\mathcal{A}^* : \mathbf{R}^p \mapsto \mathbf{S}^n$ satisfying $\langle \mathcal{A}X, \mathbf{y} \rangle = \langle X, \mathcal{A}^* \mathbf{y} \rangle \; \forall \; X \in \mathbf{S}^n$. Observe that

$$\langle \mathcal{A}X, \mathbf{y} \rangle = \sum_{i=1}^{p} y_i \; trace(A_i X) = trace \left(X \sum_{i=1}^{p} y_i A_i \right) = \langle X, \mathcal{A}^* \mathbf{y} \rangle, \tag{2.52}$$

so that

$$\mathcal{A}^* \mathbf{y} = \sum_{i=1}^{p} y_i A_i.$$

We follow the Lagrangian approach used earlier in defining the dual program of (2.42). Let $\mathbf{y} \in \mathbf{R}^p$ be the Lagrange multiplier, the primal problem is

$$\inf_{X \succeq 0} \sup_{\mathbf{y} \in \mathbf{R}^p} \left(\langle C, X \rangle + \langle \mathbf{b} - \mathcal{A}X, \mathbf{y} \rangle \right). \tag{2.53}$$

The dual is also given by

$$\sup_{\mathbf{y}\in\mathbf{R}^p} \inf_{X\succeq 0} \left(\mathbf{b}^T\mathbf{y} + \langle X, C - \mathcal{A}^*\mathbf{y}\rangle\right). \tag{2.54}$$

The weak duality

$$\inf_{X\succeq 0} \sup_{\mathbf{y}\in\mathbf{R}^p} (\langle C, X\rangle + \langle \mathbf{b} - \mathcal{A}X, \mathbf{y}\rangle) \geq \sup_{\mathbf{y}\in\mathbf{R}^p} \inf_{X\succeq 0} (\mathbf{b}^T\mathbf{y} + \langle X, C - \mathcal{A}^*\mathbf{y}\rangle) \tag{2.55}$$

also holds for this primal-dual formulation. Introducing a slack variable, W, the following is the standard dual formulation for the SDP (2.51):

$$\text{maximize } \langle \mathbf{b}, \mathbf{y}\rangle \tag{2.56a}$$

$$\text{subject to } \mathcal{A}^*\mathbf{y} + W = C \tag{2.56b}$$

$$\mathbf{y} \in \mathbf{R}^p, W \succeq 0. \tag{2.56c}$$

The following Proposition and Theorem state the conditions under which strong duality holds and when the optimal value is attained.

Proposition 2.8.1. *Given a feasible solution $\hat{X}$ of (2.51) and a feasible solution $(\hat{\mathbf{y}}, \hat{W})$ of the dual program (2.56), the duality gap is $\langle C, \hat{X}\rangle - \langle \boldsymbol{b}, \hat{\mathbf{y}}\rangle = \langle \hat{W}, \hat{X}\rangle \geq 0$. If $\langle C, \hat{X}\rangle - \langle \boldsymbol{b}, \hat{\mathbf{y}}\rangle = 0$, then $\hat{X}$ and $(\hat{\mathbf{y}}, \hat{W})$ are each optimal solution of (2.51) and (2.56), respectively. Furthermore, $\hat{W}\hat{X} = 0$.*

Theorem 2.8.1. *Suppose that the Slater constraint qualification holds for (2.51) and (2.56). Then $\hat{X}$ is optimal for the primal problem and $(\hat{\mathbf{y}}, \hat{W})$ is optimal for the dual problem if and only if*

$$\mathcal{A}^*\hat{\mathbf{y}} + \hat{W} - C = 0 \quad \textit{dual feasibility}$$

$$\boldsymbol{b} - \mathcal{A}(\hat{X}) = 0 \quad \textit{primal feasiblity}$$

$$\hat{W}\hat{X} = 0 \quad \textit{complementary slackness}$$

$$\hat{X}, \hat{W} \succeq 0.$$

Semidefinite programming has wide applicabilities, especially, for NP-hard combinatorial optimization problems. Many of the problems in this category have convex relaxation that are SDP. In many cases, the SDP relaxations are very tight. The simplest and most successful SDP relaxation is the Max-Cut problem. Others are the Quadratic Assignment Problem (QAP), Maximum Clique Problem (MCP), graph partitioning, graph colouring, maximum satisfiability, etc (see [80], [90]). In this thesis, we are proposing a new SDP relaxation for another very difficult but useful optimization problem; the Maximum Quasi-clique Problem (MQCP). This problem has the maximum clique problem as a special case.

Essentially, there are two classes of algorithms that are known to solve SDP in polynomial time, namely; the ellipsoid method [86] and the interior point method [80, 90, 125]. Both of these algorithms have different variants. Several software packages have also been developed to solve SDP (e.g, see [63, 159, 176]).

Chapter 3

Nuclear Norm Relaxation for Rank Minimization Problem

3.1 Introduction

In this chapter, we introduce the Rank Minimization Problem (RMP), which is the bedrock of the works done in this thesis. RMP is known to be non-convex and NP-hard. Hence, we provide the convex hull of rank function and subsequently convex approximation of RMP. We give some important examples of RMP and state the optimality conditions. We conclude the chapter by listing some efficient algorithms for solving the convex program.

3.2 The Rank Minimization Problem

There are many real-life problems involving constraints on rank of matrices. Low order controller design, minimal realization theory, and model reduction are some examples from control theory [146]. Furthermore, applications of RMP in optimization community include inference with partial information [148] and embedding in Euclidean spaces [104]. Ames and Vavasis [12] recently applied rank minimization technique to solve the planted clique problem. Moreover, modelling using low-rank matrices is a popular technique in semidefinite relaxation of many combinatorial optimization problems, e.g maximum cut problem [82]. A prominent class of RMP is the low-rank matrix completion. This has wide applications in recommender system, system identification, collaborative filtering, remote sensing, global positioning and quantum state tomography [11, 48, 49, 50]. We are often interested in finding the low-

rank matrix satisfying some set of constraints. Let $X \in \mathbf{R}^{n_1 \times n_2}$ be the decision variable and suppose the set of feasible solutions for the constraints, C, is convex, then the rank minimization problem can be stated as follows

$$\begin{aligned} &\text{minimize } rank(X), \\ &\text{subject to } X \in C. \end{aligned} \tag{3.1}$$

If C is affine, then (3.1) is the affine rank minimization problem and can be written as

$$\begin{aligned} &\text{minimize } rank(X), \\ &\text{subject to } \mathcal{A}(X) = \mathbf{b}, \end{aligned} \tag{3.2}$$

where $\mathcal{A} : \mathbf{R}^{n_1 \times n_2} \mapsto \mathbf{R}^p$ is a linear map and $\mathbf{b} \in \mathbf{R}^p$. The rank minimization problem is NP-Hard, non-convex optimization problem in general. All the available algorithms have exponential running time. Various heuristic algorithms based on local optimization have been proposed. These include alternating projections and its variations [84, 130], alternating matrix inequalities [154], linearization [69], and augmented Lagrangian methods [70]. Nevertheless, in some cases with special structure, RMP can be reduced to solution of a linear system or solved by singular value decomposition [117, 137, 147]. When X is a diagonal matrix, the affine rank minimization becomes the cardinality minimization problem [147]. This problem is also known to be NP-hard. Projection pursuit [76, 112] and orthogonal matching pursuit [56, 61] are some of the the proposed algorithms for solving cardinality minimization problems.

Rank minimization problems are usually solved in the optimal controls community using the "trace heuristic". In this case, one minimizes the trace of a positive semidefinite decision variable in lieu of the rank (see, for example, [27], [117]). The trace heuristic formulation is the following problem

$$\begin{aligned} &\text{minimize } trace(X), \\ &\text{subject to } \mathcal{A}(X) = \mathbf{b}, \\ &X \succeq 0. \end{aligned} \tag{3.3}$$

(3.3) relies on the fact that positive semidefinite matrices have non-negative eigenvalues. Therefore, the trace minimization is equivalent to the l_1 norm minimization of the vector of eigenvalues in this case [72]. A generalization of this heuristic to non-symmetric matrices was introduced by Fazel et al. [72, 73] (see also [147]). In this case, problem (3.2) becomes

$$\begin{aligned} &\text{minimize } ||X||_*, \\ &\text{subject to } \mathcal{A}(X) = \mathbf{b}. \end{aligned} \tag{3.4}$$

In the next section, we will show that (3.4) is a convex optimization problem. Therefore, it can be solved in polynomial time by casting it as a semidefinite program. Several algorithms have been developed to solve (3.4), e.g [45, 106, 111, 158]. Although the affine rank minimization problem is intractable in general, it has been shown that the minimum rank solution can be obtained in polynomial time by solving (3.4) for many cases. Essentially, the nuclear norm relaxation provably recovers the low rank solutions with very high probability when the linear map $\mathcal{A}$ is restricted isometry [147]. Over time, nuclear norm has been observed to produce very low rank solutions in practice. However, the theoretical basis for when it does produce the minimum rank solution recently emerged [72, 145]. The connection between the nuclear norm and rank of a matrix can be shown using the conjugate function and the notion of convex hull from convex analysis. We explore these in the next section.

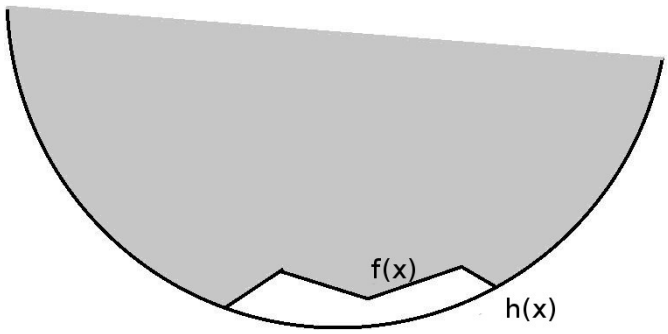


Figure 3.1: Illustration of convex hull of a function. $h(\mathbf{x})$ is the convex hull of $f(\mathbf{x})$

3.3 Convex Hull of Matrix Rank

The convex hull or convex envelope of a real-valued function $f : C \mapsto \mathbf{R}$ is the largest convex function h such that $h(x) \leq f(\mathbf{x})$ for all $\mathbf{x} \in C$ [41, 162]. This implies that among all the convex functions, h is the one that best approximates f. This is illustrated in Figure 3.1. In dealing with intractable problems like (3.2) where the objective function is non-convex, it is a common practice to use the convex hull as a tractable convex approximation. The convex hull of the rank function is given in the following theorem.

Theorem 3.3.1 ([72], Theorem 1)**.** *The convex hull of* $rank(X)$ *on the set* $\{X \in \mathbf{R}^{n_1 \times n_2} | \ ||X|| \leq 1\}$ *is the nuclear norm* $||X||_*$.

See [72], Section 5.1.5 for the proof of this Theorem 3.3.1. Summarily, the proof establishes the fact that the biconjugate of the $rank(X)$ on $\{X \in \mathbf{R}^{n_1 \times n_2} | ||X|| \leq 1\}$ is $||X||_*$ and concludes the arguement using the fact that the biconjugate of a rank function is equal to its convex hull under certain conditions (see [162], Theorem 1.3.5). We have the following corollary for the set $\{X \in \mathbf{R}^{n_1 \times n_2} | \ ||X|| \leq k\}, \forall\ k \geq 0$.

Corollary 3.3.1 ([11], Corollary 3.2.1)**.** *The convex hull of* $rank(X)$ *on the set* $\{X \in \mathbf{R}^{n_1 \times n_2} | \ ||X|| \leq k\}$ *is* $||X||_*/k \quad \forall\ k$.

Recall from the chain of inequalities (2.19) that $rank(X) \geq ||X||_*/||X||$, for all X. Hence, if $||X|| \leq 1, ||X||_* \leq rank(X)$ always holds. When the singular values are all equal to one, then $rank(X)$ is equal to the nuclear norm, i.e, $rank(X) = ||X||_*$. When the singular values are less than or equal to one, then the sum is less than the rank. Therefore, the nuclear norm is the tightest convex function that underestimates the rank function on the unit ball in the spectral norm [147]. Therefore, the nuclear norm heuristic provides bounds for solutions of affine rank minimization problem. That is, suppose X_0 is the minimum rank solution of $\mathcal{A}(X) = \mathbf{b}$, and X_0 has the spectral norm $||X_0|| = k$. The convex hull of the rank on the set $\mathcal{C} = \{X \in \mathbf{R}^{n_1 \times n_2} : ||X|| \leq k\}$ is $||X||_*/k$. If X_* is the minimum nuclear norm solution of $\mathcal{A}(X) = \mathbf{b}$, then we have:

$$||X_*||_*/k \leq rank(X_0) \leq rank(X_*).$$

Thus, when the nuclear norm solution is known, it provides both the upper and lower bound for minimum rank solution of (3.2). The nuclear norm is a convex function, hence it can be easily optimized. In fact, it is the best known approximation of the rank function over the convex set C.

As stated in Section 2.3, the nuclear norm is the dual of the spectral norm. Let $M \in \mathbf{R}^{n_1 \times n_2}$. Suppose $||M|| \leq \omega$, then we have $\omega^2 I - M^T M \succeq 0$. Then using Schur complement (see [41], Appendix A.5.5), we have

$$\begin{bmatrix} \omega I_{n_1} & M \\ M^T & \omega I_{n_2} \end{bmatrix} \succeq 0.$$

Consequently, the spectral norm can be characterised as the following semidefinte optimization problem

$$||M|| = \min \left\{ \omega \mid \begin{bmatrix} \omega I_{n_1} & M \\ M^T & \omega I_{n_2} \end{bmatrix} \succeq 0 \right\}. \tag{3.5}$$

Now, let $X \in \mathbf{R}^{n_1 \times n_2}$ and suppose $X = U\Sigma V^T$ is its SVD, where $U \in \mathbf{R}^{n_1 \times r}, V \in \mathbf{R}^{n_2 \times r}$ and $\Sigma \in \mathbf{R}^{r \times r}$ is a diagonal matrix. Define $Y := UV^T$. Since the columns of U and V are orthonormal vectors, $||Y|| = 1$ and $trace(X^T Y) = \sum_{i=1}^{r} \sigma_i(X) = ||X||_*$. Hence, the dual norm of $||.||$ is greater than or equal to the nuclear norm (see Equation (2.18)). To bound the dual norm above, we solve the following SDP

$$\begin{aligned} &\text{maximize } trace(X^T Y) \\ &\text{subject to } ||Y|| \leq 1. \end{aligned} \tag{3.6}$$

Using (3.5), problem (3.6) is equivalent to

$$\begin{aligned} &\text{maximize } trace(X^T Y) \\ &\text{subject to } \begin{bmatrix} I_{n_1} & Y \\ Y^T & I_{n_2} \end{bmatrix} \succeq 0. \end{aligned} \tag{3.7}$$

The dual of (3.7) is

$$\begin{aligned} &\text{minimize } \frac{1}{2}(trace(Z_1) + trace(Z_2)) \\ &\text{subject to } \begin{bmatrix} Z_1 & X \\ X^T & Z_2 \end{bmatrix} \succeq 0. \end{aligned} \tag{3.8}$$

If we set $Z_1 := U\Sigma U^T$ and $Z_2 := V\Sigma V^T$, then the triple (Z_1, Z_2, X) will be feasible for (3.8), since

$$\begin{bmatrix} Z_1 & X \\ X^T & Z_2 \end{bmatrix} = \begin{bmatrix} U \\ V \end{bmatrix} \Sigma \begin{bmatrix} U \\ V \end{bmatrix}^T \succeq 0.$$

In addition, we have $trace(Z_1) = trace(Z_2) = trace(\Sigma)$. Therefore, the objective function satisfies $\frac{1}{2}(2trace(\Sigma)) = trace(\Sigma) = ||X||_*$. Hence, the dual of the spectral norm is indeed the nuclear norm. This assertion can also be proved using the Slater's condition. Since there is no duality gap between (3.7) and (3.8), either of them can be used to compute the nuclear norm.

Substituting (3.8) into problem (3.4), we get the following semidefinite formulation

$$\begin{aligned} &\text{minimize } \frac{1}{2}(trace(Z_1) + trace(Z_2)) \\ &\text{subject to } \begin{bmatrix} Z_1 & X \\ X^T & Z_2 \end{bmatrix} \succeq 0, \\ &\mathcal{A}(X) = \mathbf{b}. \end{aligned} \tag{3.9}$$

Now, having established that problem (3.4) is the best known approximation of (3.2), we can go now ahead to write the dual problem to (3.4) as

$$\begin{aligned} &\text{maximize } \mathbf{b}^T\mathbf{z}, \\ &\text{subject to } ||\mathcal{A}^*(\mathbf{z})|| \leq 1, \end{aligned} \tag{3.10}$$

where $\mathcal{A}^* : \mathbf{R}^p \mapsto \mathbf{R}^{n_1 \times n_2}$ and $\mathbf{z} \in \mathbf{R}^p$. In like manner, using the SDP for characterization of the operator norm given in (3.5), then (3.10) becomes

$$\begin{aligned} &\text{maximize } \mathbf{b}^T\mathbf{z}, \\ &\text{subject to } \begin{bmatrix} I_{n_1} & \mathcal{A}^*(\mathbf{z}) \\ \mathcal{A}^*(\mathbf{z})^T & I_{n_2} \end{bmatrix} \succeq 0. \end{aligned} \tag{3.11}$$

3.4 Optimality conditions for nuclear norm minimization problem

The optimality condition of the nuclear norm heuristic is based on the notion of subdifferential defined in Section 2.6 in Chapter 2. Suppose $X \in \mathbf{R}^{n_1 \times n_2}$ and $X = U\Sigma V^T$ is the singular value decomposition of X, where U, V are $n_1 \times r, n_2 \times r$ matrices respectively, and Σ is an $r \times r$ diagonal matrix. Then the subdifferential of the nuclear norm at X is given by (see, [101, 168])

$$\partial||X||_* = \{UV^T + W \mid W^T U = 0, WV = 0, ||W|| \leq 1\}. \tag{3.12}$$

Consequently, we have the following compact optimality conditions for the convex problem (3.4).

Theorem 3.4.1 ([147], Equation (2.11), [11], Theorem 3.2.3)**.** *An $n_1 \times n_2$ matrix X is an optimal solution of* (3.4) *if there exists $\boldsymbol{z} \in \mathbf{R}^p$ such that*

$$\mathcal{A}(X) = \boldsymbol{b} \tag{3.13}$$

$$\mathcal{A}^*(\boldsymbol{z}) \in \partial||X||_*. \tag{3.14}$$

(3.13) ensures feasibility of the linear equations, while (3.14) certifies that there is no feasible direction for improvement. To see this, let W be any other matrix in the primal feasible set of (3.4). Since $\mathcal{A}^*(\mathbf{z}) \in \partial||X||_*$, we have

$$\begin{aligned} ||W||_* &\geq ||X||_* + \langle \mathcal{A}^*(\mathbf{z}), W - X \rangle \\ &= ||X||_* + \langle \mathbf{z}, \mathcal{A}(W - X) \rangle \\ &= ||X||_*. \end{aligned}$$

The last equality follows from the fact that W and X both feasible.

3.5 Examples of rank minimization problem

As stated at the beginning of this chapter, there are many problems involving rank constraints. However, we discuss three of them with direct link to our application.

3.5.1 The cardinality minimization problem

One of the important examples and a special case of rank minimization problem is the cardinality minimization problem. The Cardinality Minimization Problem (CMP) requires finding the sparsest vector which satisfies a given set of linear constraints. In other words, given $M \in \mathbf{R}^{n_1 \times n_2}$ and $\mathbf{b} \in \mathbf{R}^{n_1}$, we want to find the vector $\hat{\mathbf{x}} \in \mathbf{R}^{n_2}$ which solves $M\mathbf{x} = \mathbf{b}$ and have the minimum number of non-zero entries. The number of non-zero entries of a vector $\mathbf{x} \in \mathbf{R}^{n_2}$ is known as its cardinality and is denoted as $card(\mathbf{x})$. It is also sometimes called the weight, sparsity or $||.||_0$ norm of $\mathbf{x}$. We need to stress here that $||.||_0$ norm is not a norm in the real sense since it does not satisfy the linearity property, i.e, $||\lambda\mathbf{x}||_0 = ||\mathbf{x}||_0, \forall\ \lambda \neq 0$. A vector is sparse if it has few nonzero entries, i.e $\mathbf{x} \in \mathbf{R}^{n_2}$ and $card(\mathbf{x}) << n_2$.

To show that the CMP is a special case of the RMP, define the matrix X in (3.2) as a diagonal matrix, that is $X = diag(\mathbf{x})$ for $\mathbf{x} \in \mathbf{R}^n$. This way, the rank of X is equal to the number of nonzero entries in $\mathbf{x}$ and $rank(diag(\mathbf{x})) = ||\mathbf{x}||_0$. Consequently, (3.2) for CMP can be written as

$$\begin{aligned} &\text{minimize } rank(diag(\mathbf{x})) \\ &\text{subject to } \mathcal{A}(diag(\mathbf{x})) = \mathbf{b}, \end{aligned} \tag{3.15}$$

where the $n \times n$ diagonal matrix $diag(\mathbf{x})$ is the decision variable, $\mathcal{A} : \mathbf{R}^{n \times n} \mapsto \mathbf{R}^p$ is a given linear map and the vector $\mathbf{b} \in \mathbf{R}^p$ is also given. Since $\mathcal{A}$ acts only on the diagonal entries, there exists $M \in \mathbf{R}^{p \times n}$ such that $\mathcal{A}(diag(\mathbf{x})) = M\mathbf{x}$, $\forall\, \mathbf{x} \in \mathbf{R}^n$.

CMP is applicable in many domains, e.g, compressed sensing and single pixel camera design. Let $\hat{\mathbf{x}}$ represents a particular sparse signal. The problem, in this case, is how to take advantage of the sparsity of $\hat{\mathbf{x}}$ and encode it to reduce the amount of storage space required. This could be achieved by encoding $\hat{\mathbf{x}}$ as a linear combination of known signals. Hence, we store $\hat{\mathbf{x}}$ as $\mathbf{b} = M\hat{\mathbf{x}}$, for some M and $\mathbf{b}$. For this encoding to be useful, we must be able to reconstruct $\hat{\mathbf{x}}$ from $\mathbf{b}$. This process of reconstructing a sparse signal from a small number of measurements is known as *compressed* or *compressive* sensing. The term compressed sensing is from the fact that, in practice, the complete signal is not recorded. Rather, the measurement vector $\mathbf{b} = M\mathbf{x}$ is acquired by undersampling $\hat{\mathbf{x}}$ several time, independently. Therefore, only a "compressed" form of the original signal is "sensed". So, if M is selected such that $\hat{\mathbf{x}}$ is the unique solution of cardinality minimization of $M\mathbf{x} = \mathbf{b}$, then we can recover $\hat{\mathbf{x}}$ by solving

$$\begin{aligned} &\text{minimize } ||\mathbf{x}||_0, \\ &\text{subject to } M\mathbf{x} = \mathbf{b}. \end{aligned} \tag{3.16}$$

Problem (3.16) is called the l_0-norm minimization problem for the following reason. Recall that, for $p \geq 1$, the standard p-norm is given by

$$||\mathbf{x}||_p = \left(\sum_{k=1}^{n} |x_k|^p \right)^{1/p},$$

whereas the p-norm minimization problem is given by

$$\begin{aligned} &\text{minimize } \sum_{k=1}^{n} |x_k|^p, \\ &\text{subject to } \mathbf{x} \in \Psi, \end{aligned} \tag{3.17}$$

where Ψ is a convex set. So, as $p \to 0$, the objective function in (3.17) converges to

$$\lim_{p \to 0} \sum_{k=1}^{n} |x_k|^p = ||\mathbf{x}||_0.$$

The cardinality minimization problem (3.16) is non-convex and NP-hard [11]. A popular convex surrogate used instead of l_0 minimization is the l_1 minimization. Recall that, for $\mathbf{x} \in \mathbf{R}^n$, the l_1-norm is defined as

$$||\mathbf{x}||_1 = \sum_{k=1}^{n} |x_k|.$$

The l_1-norm relaxation for problem (3.16) is the following convex program:

$$\begin{aligned} &\text{minimize } ||\mathbf{x}||_1, \\ &\text{subject to } M\mathbf{x} = \mathbf{b}. \end{aligned} \tag{3.18}$$

The semidefinite formulation for (3.18) (see [54], Appendix A) is

$$\begin{aligned} &\text{minimize } \mathbf{1}_n^T Z \mathbf{1}_n, \\ &\text{subject to } -Z_{ij} \leq X_{ij} \leq Z_{ij} \quad \forall\, (i,j) \\ &\qquad\qquad MX = \mathbf{b}, \end{aligned} \tag{3.19}$$

where $\mathbf{1}_n \in \mathbf{R}^n$ is an n-dimensional vector with all entries equal to one, $Z \in \mathbf{R}^{n \times n}$ and $X = diag(\mathbf{x})$. Figure 3.2 gives an intuitive illustration of why l_1- norm minimization produces sparse solutions. (3.18) can be reformulated as a linear program by splitting $\mathbf{x}$ into its positive and negative components such that $\mathbf{x} = \mathbf{x}^+ + \mathbf{x}^-$ where $\mathbf{x}^+, \mathbf{x}^-$ are

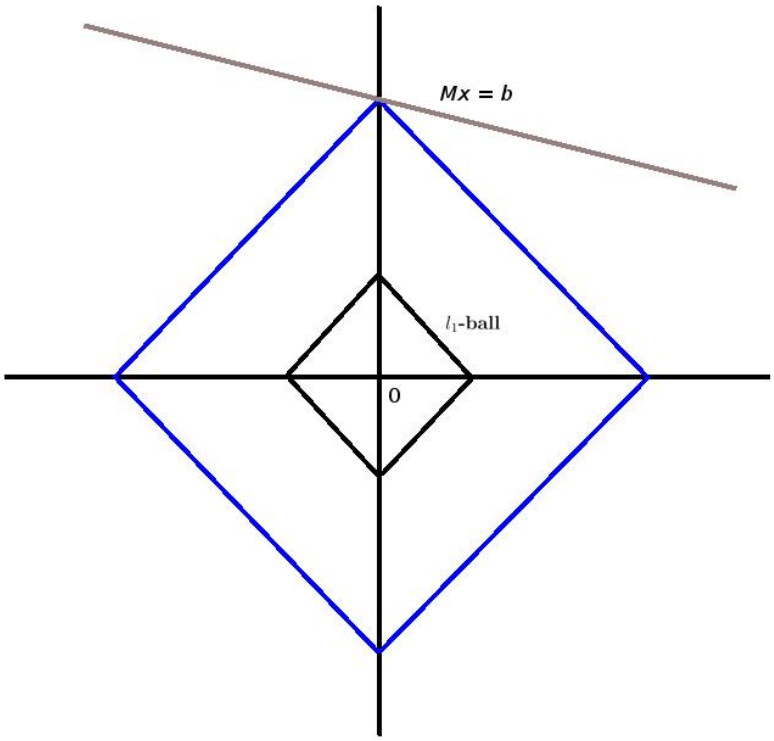


Figure 3.2: Illustration of sparse solution via l_1-norm minimization

the absolute values of the positive and negative entries of **x** respectively. So we have $x_k^+ = x_k$ for $x_k \geq 0$ and 0 otherwise. In like manner, $x_k^- = x_k$ for $x_k \leq 0$ and 0 otherwise. Consequently, we have the following linear program

$$\begin{aligned} &\text{minimize } \sum_{k=1}^{n}(x_k^+ + x_k^-) \\ &\text{subject to } x = x^+ + x^- \\ &M\mathbf{x} = \mathbf{b} \\ &x^+, x^- \geq 0. \end{aligned} \tag{3.20}$$

Therefore, one can obtain solution to (3.18) by solving (3.20). (3.20) can be solved using any linear programming method. In particular, simplex method and the interior point methods can be used [125]. In a more difficult case, (3.18) is equivalent to a

second order cone program (SOCP). This case can also still be solved using interior point methods. However, a specialized method is expected to outperform such existing standard methods. Orthogonal matching pursuit [62], iterative hard thresholding [34, 75], the homotopy method [131] are some of the efficient specialized algorithms for solving (3.18).

3.5.2 The low-rank matrix completion problem

A very important class of linearly constrained matrix rank minimization problem is the matrix completion problem. Low rank matrix recovery from partial sampling of its entries is applicable in collaborative filtering [148], dimensionality reduction [169] and multi-class learning [129]. Let $M \in \mathbf{R}^{n_1 \times n_2}$ be a rectangular matrix. Suppose only m entries of M are available with $m << n_1 n_2$. Matrix completion problem answers the question: can we recover M from its m partially observed entries? Obviously, this is not possible in general. In fact, if M is an $n \times n$ square matrix of unknown rank, r, then it can easily be observed that matrix completion is impossible unless $m \geq (2n - r)r$. This is because an $n \times n$ matrix of rank r depends on $2nr - r^2$ degree of freedom. The singular value decomposition (SVD) is helpful in revealing these degree of freedom. In many cases where the matrix we intend to recover is known to be low ranked, recovery may be possible. One of such scenario is when the columns of the matrix are independent and identically distributed random samples with low covariance. Furthermore, this problem can easily be ill-posed, even with prior knowledge that the unknown matrix has low rank. For example, suppose $\mathbf{y} \in \mathbf{R}^n$ is an n dimensional vector and $M \in \mathbf{R}^{n \times n}$ is a rank-one matrix where:

$$M = e_1 \mathbf{y}^T$$

and e_1 is the first element of the standard basis of the Euclidean vector space. In this case, we cannot recover M by sampling its entries. Even if we have up to 95% random sample of this matrix, we will probably miss the elements in the first row. Hence, we will not be able to recover $\mathbf{y}$, and by implication, M. This is analogous in compressed sensing to the fact that one cannot recover a signal supposed to be sparsed in time domain by sub-sampling in time domain [48]. Therefore, it is not possible to recover all low-rank matrices from a sample of its entries. To avoid such scenario, it has become a customary approach to presume that M has additional properties known as the incoherence [49, 50, 52, 54, 85]. Suppose $M \in \mathbf{R}^{n_1 \times n_2}$ has the singular value decomposition $M = U\Sigma U^T$. M satisfies the incoherence conditions, with parameter $\mu \in [1, \frac{\min(n_1,n_2)}{r}]$, if

$$\max_i ||U^T e_i||^2 \leq \frac{\mu r}{n_1}, \quad \max_i ||V^T e_i||^2 \leq \frac{\mu r}{n_2} \tag{3.21}$$

and

$$||UV^T||_\infty \leq \sqrt{\frac{\mu r}{n_1 n_2}}, \tag{3.22}$$

where $e_i's$ are the canonical basis vectors with suitable dimensions and $||.||_\infty$ is entry-wise l_∞ norm. Since the orthogonal projection onto the column space of U is given by $P_U = UU^T$, $\max_i ||U^T e_i||^2 \leq \frac{\mu r}{n_1}$ and $\max_i ||P_U e_i||^2 \leq \frac{\mu r}{n_1}$ are equivalent [52]. Similar argument goes for P_V. The incoherence condition asserts that for small values of μ, the singular vectors are reasonably spread out [49, 50, 85]. The notion of incoherence was studied in connection with recovery of sparse representation of vectors from "*over complete dictionary*" [65]. In addition, incoherence has been used in compressed sensing [54]. The main objective in compressed sensing is recovery of "low-dimensional" objects, e.g sparse vectors [47, 67] or low-rank matrices [49, 147]; given incomplete observations. Moreover, matrix completion, using nuclear norm minimization, was

first proposed in [49]. It was based on ideas and techniques from compressed sensing. Candés and Recht [49] showed that other structures, apart from sparse signals and images, can be recovered from a small set of observations. They introduced the incoherent model. More importantly, Candés and Recht proved that a random matrix $M \in \mathbf{R}^{n\times n}$ of rank r with $m < n^2$ observed entries can perfectly be recovered if

$$m \geq Crn^{6/5} \log n,$$

for some numerical constant $C > 0$ with high probability. On the other hand, from information theory point of view, $m = \mathcal{O}(nr)$ for recovery to be possible with any kind of method whatsoever. Since the emergence of this problem, several other works have followed [49], providing theoretical guarantee and efficient algorithms for successful recovery of low-rank matrices [48, 50, 95]. Although, other methods have been proposed, e.g [95], the method of choice remains the use of convex optimization. [48, 49, 50, 145] all proved the validity and effectiveness of this approach. Ames and Vavasis [12] adapted the idea of low-rank matrix recovery to solve the planted clique problem but with a different proof technique. [57] reduced the sample complexity for recovery of semidefinite matrix from $\mathcal{O}(nr^2 \log^2 n)$ to $\mathcal{O}(nr \log^2 n)$. The major step used to achieve this improvement is by deriving a new bound using the $l_{\infty,2} - norm$.

Let $M \in \mathbf{R}^{n_1\times n_2}$ is a rectangular matrix and that m entries of M are sampled such that $\{M_{ij} : (i,j) \in \Omega\}$ where Ω is a random subset of $\{1,\ldots,n_1\} \times \{1,\ldots,n_2\}$ with cardinality m. Candés and Recht [49] established that most matrices M of rank r can be perfectly recovered by solving the rank minimization problem

$$\begin{aligned} &\text{minimize } rank(X), \\ &\text{subject to } X_{ij} = M_{ij},\ \forall\ (i,j) \in \Omega, \end{aligned} \tag{3.23}$$

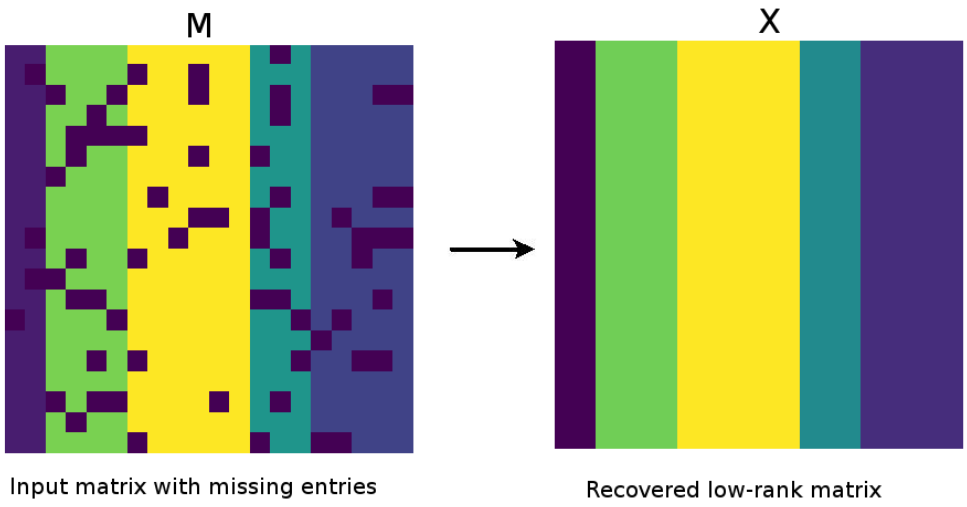


Figure 3.3: Schematic diagram of low-rank matrix completion.

where $X \in \mathbf{R}^{n_1 \times n_2}$ is the decision variable. If there was a unique low-rank matrix that fits the sampled data, then M can be recovered with high probability (see Figure 3.3 for illustration). This holds if sufficient entries of M is sampled and M is a random rank r matrix. However, since the rank function is known to be NP-hard, (3.23) is not so useful. [49] proposed following convex relaxation instead

$$\begin{aligned} &\text{minimize } ||X||_*, \\ &\text{subject to } X_{ij} = M_{ij} \; \forall \; (i,j) \in \Omega. \end{aligned} \tag{3.24}$$

Using (3.9), the SDP formulation for (3.24) is given as

$$\begin{aligned} &\text{minimize } \frac{1}{2}(trace(Z_1) + trace(Z_2)) \\ &\text{subject to } \begin{bmatrix} Z_1 & X \\ X^T & Z_2 \end{bmatrix} \succeq 0, \\ &\qquad X_{ij} = M_{ij} \; \forall \; (i,j) \in \Omega. \end{aligned} \tag{3.25}$$

The bound on minimum sample required for successful recovery in [49] has later been

improved in [50, 85, 95]. However, the sharpest bound is the work of Recht [145]. The result is contained in the following theorem.

Theorem 3.5.2.1 ([145], Theorem 2)**.** *Let $M \in \mathbf{R}^{n_1 \times n_2}$ be of rank r with SVD $U\Sigma V^T$. Without loss of generality, assume that $n_1 \leq n_2, \Sigma \in \mathbf{R}^{r\times r}, U \in \mathbf{R}^{n_1 \times r}$ and $V \in \mathbf{R}^{n_2 \times r}$. Assume that*

A0 The row and column spaces of M have coherences bounded above by some positive μ_0.

A1 The matrix UV^T has a maximum entry bounded, in absolute value, by $\mu_1\sqrt{r/(n_1n_2)}$ for some positive μ_1.

Suppose m entries of M are observed with locations sampled uniformly at random. If

$$m \geq 32 \max\{\mu_1^2, \mu_0\} r(n_1 + n_2)\beta \log^2(2n_2), \tag{3.26}$$

for some positive constant β, then the minimizer to problem (3.24) is unique and equal to M with probability at least $1 - 6\log(n_2)(n_1 + n_2)^{2-2\beta} - n_2^{2-2\beta^{1/2}}$.

3.5.3 The matrix decomposition problem

An extension of the matrix completion problem known as matrix decomposition problem was studied in [52, 54, 58]. Some application domains of this problem include face recognition, ranking and collaborative filtering, latent semantic indexing and video surveillance as enumerated in [52]. Others are graphical model learning, linear system identification and coherence optical systems discussed in [54]. In this setting, the low rank matrix is a submatrix of a larger matrix with other additional unwanted entries (sometimes called error or corruptions). Such a matrix is formed by adding an unknown low-rank matrix to an unknown sparse matrix. The goal is to recover the low-rank and

sparse components of the given matrix. A more difficult version of this is the case when the low-rank matrix has some missing entries. Chen et al. [58] calls this case recovery in presence of errors and erasures. Refer to Figure 3.3 for a pictorial description of this problem. Generally, the matrix decomposition problem is stated as follows. We are given a matrix $M \in \mathbf{R}^{n_1 \times n_2}$, which is a sum of a low rank matrix $B_0 \in \mathbf{R}^{n_1 \times n_2}$ and a sparse (errors) matrix $C_0 \in \mathbf{R}^{n_1 \times n_2}$. The cardinality and the values of the non-zero entries of C_0 are not known; likewise the locations of the non-zero entries are also not known. The goal is to recover B_0 and C_0 from M. The natural mathematical formulation for this problem is:

$$\begin{aligned} &\text{minimize } rank(B) + \lambda ||C||_0, \\ &\text{subject to } B + C = M. \end{aligned} \tag{3.27}$$

$B, C \in \mathbf{R}^{n_1 \times n_2}$ are the decision variables while λ is a positive constant. From the foregoing, and following from [52, 54, 171], the convex relaxation for (3.27) is

$$\begin{aligned} &\max \ ||B||_* + \lambda ||C||_1, \\ &\text{subject to } B + C = M, \end{aligned} \tag{3.28}$$

where $||C||_1 = \sum_{ij} |C_{ij}|$ is the entry-wise l_1 norm of C. By the foregoing, the SDP formulation for (3.28) is given as

$$\begin{aligned} &\text{minimize } \frac{1}{2}(trace(Z_1) + trace(Z_2)) + \lambda \mathbf{1}_n^T Z \mathbf{1}_n, \\ &\text{subject to } \begin{bmatrix} Z_1 & B \\ B^T & Z_2 \end{bmatrix} \succeq 0, \\ &\quad - Z_{ij} \leq C_{ij} \leq Z_{ij} \quad \forall \, (i,j) \\ &\quad B + C = M, \end{aligned} \tag{3.29}$$

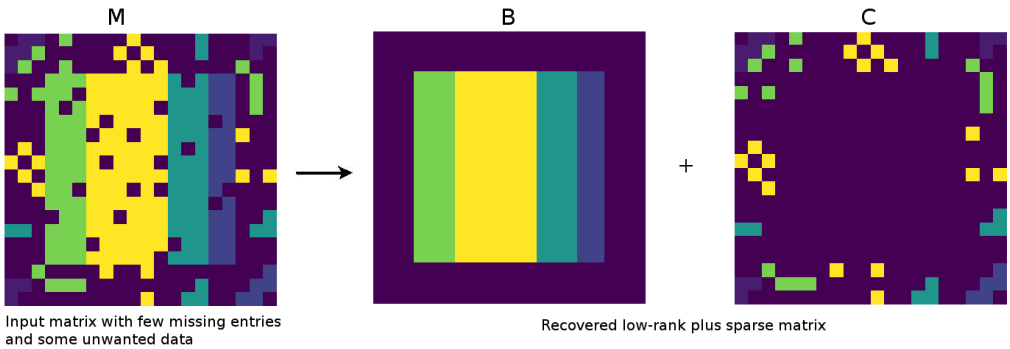


Figure 3.4: Schematic diagram of low-rank plus sparse matrix decomposition.

where $Z, Z_1, Z_2 \in \mathbf{R}^{n_1 \times n_2}$. A pertinent question will be; under what condition is decomposition possible? Wright et al. [171] showed that low-rank plus sparse matrices can successfully be recovered by solving (3.28), with high probability, if the rank of B_0 is $\mathcal{O}(\beta \frac{m}{\log m})$ for some $m \geq m_0$ where m is the number of known entries of the low rank matrix B_0, β and m_0 are positive constants. This holds with high probability as m increases. This is an instance of the so-called *blessing of dimensionality* [66]. Another important factor in the success of (3.28) is the choice of λ. [171] observed that the right scaling for λ is $\lambda = \mathcal{O}(m^{-1/2})$, however fixed $\lambda = \frac{1}{m^2}$. [54] presented a solution using a geometric method of analysis, whereby the tangent spaces of the algebraic varieties of the low-rank and sparse matrices play an important role. For the result of Chandrasekaran et al. [54] to be valid, λ must satisfy

$$\lambda \in \left(\frac{2\text{inc}(B_0)}{1 - 8\text{dg}_{\max}(C_0)\text{inc}(B_0)}, \frac{1 - 6\text{dg}_{\max}(C_0)\text{inc}(B_0)}{\text{dg}_{\max}(A^*)} \right), \tag{3.30}$$

where $\text{inc}(M) := \max\{\mu(\text{row space}(M)), \mu(\text{column space}(M))\}$ and $\text{dg}_{\max}(M)$ is the maximum number of non-zero entries in each column/row of a matrix M. Note that row space(M) and column space(M) refer to the row space and column space of a matrix, M, respectively. In addition, their conditions for guaranteed success of the convex program are based on the following two quantities. The first one is the maximum ratio

between the $||.||_\infty$ and the $||.||$, restricted to the subspace generated by matrices with row or column spaces agreeing with those of B_0. The second quantity is the maximum ratio between the $||.||$ and the $||.||_\infty$, restricted to the subspace of matrices that have support in C_0. Chandrasekaran et al. claim that if the product of the two quantites is small, the recovery is exact, provided λ satisfies (3.30). Candès et al. [52] provide a unique way of chosing λ which works for every problem instance. They found that when the input matrix satisfies the incoherence conditions with some other little assumptions, (3.27) perfectly recovers the low-rank and the sparse components. Let $n = \min\{n_1, n_2\}$. The main result is stated here.

Theorem 3.5.3.1 ([52], Theorem 1.1)**.** *Suppose $B_0 \in \mathbf{R}^{n\times n}$ obeys (3.21) - (3.22). Let $S \in \mathbf{R}^{n\times n}$ be any sign matrix. Suppose that the support set Ω of C_0 is uniformly distributed among all sets of cardinality m, and that $sgn([C_0]_{ij}) = S_{ij}$ for all $(i, j) \in \Omega$. Then, there is a numerical constant c such that with probability at least $1 - cn^{-10}$ (over the choice of support of C_0), (3.27) with $\lambda = 1/\sqrt{n}$ is exact provided that*

$$rank(B_0) \leq \frac{\rho_r n}{\mu}(\log n)^{-2} \text{ and } m \leq \rho_s n^2, \tag{3.31}$$

where $\rho_r, \rho_s \geq 0$ are numerical constants.

The proof techniques follow the path in [49] but also rely very much on the powerful Golfing Scheme introduced in [85]. Our work applies the matrix decomposition to a specific problem, recovery of planted maximum quasi-clique. We are inspired by the work of Ames and Vavasis [12] who applied the matrix completion technique to planted maximum clique recovery. However, our problem formulation and proof technique differ greatly from [12]. Indeed, we have the following theorem.

Theorem 3.5.3.2. *Suppose $B_0 \in \mathbf{R}^{n\times n}$ obeys the incoherence conditions. Let $\lambda = \frac{1}{\sqrt{n}}$ and $S \in \mathbf{R}^{n\times n}$ be the random sign matrix. Suppose further that the entries of C_0*

is such that $Sgn([C_0]_{ij}) = S_{ij}$*, where* $Sgn(.)$ *is a sign function. Denote the optimal pair of problem* (3.28) *as* (B^*, C^*)*. There exists universal positive constants* c *and* c_0*; independent of* n*, such that if*

$$c_0 \frac{\mu r \log n}{n} \leq p \leq 1,$$

where p *denotes the sampling probability of entries of* B_0*, then* (B^*, C^*) *is the unique optimal solution of* (3.28) *with probability at least* $1 - c_0 n^{-10}$.

In Chapter 5, we specialize Theorem 3.5.3.2 to our problem and proof the theoretical guarantee for maximum quasi-clique recovery using convex relaxation. Among other things, this thesis extends the concept in [57] to matrix decomposition problem.

3.6 Conditions for guaranteed success of the nuclear norm heuristic

At the beginning of Section 3.5.2, we addressed identifiability issues regarding the low-rank matrix to be recovered. We stated that for recovery to be possible at all, the matrix must satisfy the incoherence conditions (3.21) - (3.22) and certain number of entries must be known. Let $\mathcal{A} : \mathbf{R}^{n_1 \times n_2} \mapsto \mathbf{R}^d$ be a linear map and $b = \mathcal{A}(X_0)$, where X_0 is a rank r matrix. Define

$$\begin{aligned} X^* &:= \arg\min ||X||_* \\ &\text{subject to } \mathcal{A}(X) = b. \end{aligned} \tag{3.32}$$

In this section, we provide theoretical guarantees that ensure that the low-rank matrix X_0 is the same as the nuclear norm solution X^*, i.e, $X_0 = X^*$. The main conditions

will be defined by a sequence of parameters δ_r which measure the action of the linear map $\mathcal{A}$, when restricted to the set of rank r matrices.

3.6.1 Restricted Isometries

Recht et al. [147] extended the restricted isometry property (RIP) from vectors to matrices and gave the following definition. Let $\mathcal{A} : \mathbf{R}^{n_1 \times n_2} \mapsto \mathbf{R}^d$ be a linear map and assume $n_1 \leq n_2$. For every integer $r \in \{1, \ldots, n_1\}$. The r-restricted isometry constant is defined as the smallest number $\delta_r(\mathcal{A})$ such that

$$(1 - \delta_r(\mathcal{A}))||X||_F \leq ||\mathcal{A}(X)|| \leq (1 + \delta_r(\mathcal{A}))||X||_F \tag{3.33}$$

holds for every matrix X with rank less or equal to r. This definition is a generalization of the restricted isometry property (RIP) for sparse vectors developed by Candes and Tao [46] to low-rank matrices. By definition, $\delta_r(\mathcal{A}) \leq \delta_{r'}(\mathcal{A})$ for all r less than or equal to r' [147]. The following two theorems characterize when X_0 is equal to the minimizer of the nuclear norm heuristic, X^*, using the r-restricted isometry contstant $\delta_r(\mathcal{A})$. The theorems are a generalization of similar results in cardinality minimization to low-rank matrix recovery.

Theorem 3.6.1.1 ([147], Theorem 3.2)**.** *Suppose that $\delta_{2r} < 1$ for some integer $r \geq 1$. Then X_0 is the only matrix of rank at most r satisfying $\mathcal{A}(X) = b$.*

Theorem 3.6.1.1 is an extension of Lemma 1.2 of [46] to low rank matrix recovery. The next theorem gives the condition which guarantees $X^* = X_0$.

Theorem 3.6.1.2 ([147], Theorem 3.3)**.** *Suppose that $\delta_{5r} \leq 1/10$ for some integer $r \geq 1$. Then $X_0 = X^*$.*

The proof of Theorem 3.6.1.2 follows the approach in [51] (see [147] for details). Theorems 3.6.1.1 and 3.6.1.2 state the conditions under which the nuclear norm solu-

tion is equivalent to the rank r solution. However, nothing has yet been said about the linear mapping $\mathcal{A}$ for which $\delta_r < 1$ holds. In the next subsection, we will show that random linear maps, sampled from some distributions of matrices, have this property with high probability.

3.6.2 Nearly Isometric Random Matrices

Recht et al. [147] established that when the linear map, $\mathcal{A} : \mathbf{R}^{n_1 \times n_2} \mapsto \mathbf{R}^d$, is sampled from a class of probability distributions which obey some large deviation inequalities, then $\mathcal{A}$ will have small r-restricted isometry constant as d, n_1 and n_2 tend to ∞.

Definition 3.6.2.1 (Definition 4.1 of [147])**.** *Let $\mathcal{A}$ be a random variable that takes values in linear maps from $\mathbf{R}^{n_1 \times n_2}$ to $\mathbf{R}^d$. $\mathcal{A}$ is* nearly isometrically distributed *if for all $X \in \mathbf{R}^{n_1 \times n_2}$,*

$$\mathbf{E}[||\mathcal{A}(X)||^2] = ||X||_F^2 \tag{3.34}$$

and, for all $0 < \epsilon < 1$, we have

$$Pr(|||\mathcal{A}(X)||^2 - ||X||_F^2| \geq \epsilon ||X||_F^2) \leq 2 \exp\left(-\frac{d}{2}\left(\frac{\epsilon^2}{2} - \frac{\epsilon^3}{3}\right)\right), \tag{3.35}$$

and, for all $t > 0$, there exists some constant $\zeta > 0$, such that

$$Pr\left(||\mathcal{A}|| \geq 1 + \sqrt{\frac{n_1 n_2}{d}} + t\right) \leq \exp(-\zeta d t^2). \tag{3.36}$$

The two factors for a random linear map to be nearly isometric are the following [147]. First of all, the linear map must be isometric in expectation. In addition, the probability of large distortions of length has to be exponentially small. As an example, the family of random linear maps with matrix representations having independent and

identically distributed (i.i.d.) Gaussian entries, $M_{ij} \approx N(0, 1/d)$, is nearly isometric. Hence, the nuclear norm relaxation for rank minimization, subject to Gaussian constraints is exact with probability tending to 1 for large enough d [11]. Similarly, the ensemble of random linear operators with matrix representations

$$M_{ij} = \begin{cases} \sqrt{\frac{1}{d}} & \text{with probability } \frac{1}{2}, \\ -\sqrt{\frac{1}{d}} & \text{with probability } \frac{1}{2}, \end{cases} \tag{3.37}$$

and

$$M_{ij} = \begin{cases} \sqrt{\frac{3}{d}} & \text{with probability } \frac{1}{6}, \\ 0 & \text{with probability } \frac{2}{3}, \\ -\sqrt{\frac{3}{d}} & \text{with probability } \frac{1}{6}, \end{cases} \tag{3.38}$$

sampled from an i.i.d. symmetric Bernoulli distribution are nearly isometric.

The top singular value of M is concentrated around $1 + (n_1 n_2)/d$ for all the ensembles above [173]. Our last theorem for this section establishes the fact that if $\mathcal{A} : \mathbf{R}^{n_1 \times n_2} \mapsto \mathbf{R}^d$ is randomly sampled from a nearly isometric family of linear operators then $\delta_r(\mathcal{A})$ is small for sufficiently large d with probability tending to 1 as d tends to infinity.

Theorem 3.6.2.1 ([147], Theorem 4.2)**.** *Fix* $0 < \delta < 1$. *If* $\mathcal{A} : \mathbf{R}^{n_1 \times n_2} \mapsto \mathbf{R}^d$ *is a nearly isometric random variable, then, for every* $1 \leq r \leq \min\{n_1, n_2\}$*, there exist positive constants* c_0, c_1 *depending only on* δ *such that, with probability at least* $1 - \exp(-c_1 d)$*,* $\delta_r(\mathcal{A}) \leq \delta$ *whenever* $d \geq c_0 r(n_1 + n_2)\log(n_1 n_2)$.

The low-rank matrix recovery guarantees given by Theorems 3.6.1.1, 3.6.1.2 and 3.6.2.1 have been improved upon subsequently (see [47, 48, 49, 50, 145], for example).

3.7 Algorithms for Nuclear Norm Minimization

Several algorithms have been developed for effective minimization of the nuclear norm constrained by affine subspace of matrices. We highlight a few of them that have been known to be very successful in this section.

3.7.1 Interior Point Method

The interior point method for SDP is suitable for solving relatively small nuclear norm minimization problems for which a high degree of numerical precision is required. Although interior point methods give fast convergence and very accurate, the memory requirements and the number of computation per iteration are usually high. The primal SDP problem (3.9) has one $(n_1 + n_2) \times (n_1 + n_2)$ semidefinite constraint and p affine constraints while the dual problem (3.11) has one $(n_1 + n_2) \times (n_1 + n_2)$ semidefinite constraint and p scalar decision variables. Hence, the total number of decision variables for the primal-dual problem is $\binom{n_1+n_2+1}{2} + p$ [147]. The state-of-the-art interior point solvers for SDP generally use primal-dual methods. Update direction for the current solution are computed by solving a suitable Newton system.

If the matrix dimensions is less than 100×100, then any good interior point SDP solver, such as SeDuMi [155] or SDPT3 [159], will handle the problem with high accuracy [52, 107, 147]. However, when the dimensions of the matrix are larger than 100×100, the number of equations will be running into thousands. In this case, the corresponding Newton systems will be quite large, and without any specific additional structure, the memory requirements of such dense systems limit the size of problems that can be solved. This can be controlled to a reasonable extent by leveraging on the problem structure when solving the Newton system. The specilaized interior point method for nuclear norm of [107] is based on this idea.

3.7.2 Iterative thresholding

Cai et al. [45] developed an algorithm known as singular value thresholding (SVT), which performs nuclear norm minimization by repeatedly shrinking the singular values of an appropriate matrix. The iterative algorithm produces a sequence of matrices $\{X^k, Y^k\}$ and performs a soft-thresholding operation on the singular values of the matrix Y^k. Essentially, this reduces the complexity of each iteration to the cost of an SVD. Initializing $Y^0 = 0 \in \mathbf{R}^{n_1 \times n_2}$ and fixing $\tau > 0$ and a sequence of scalar step sizes $\{\delta_k\}_{k \geq 1}$, the algorithm inductively defines

$$\begin{cases} X^k = \text{shrink}(Y^{k-1}, \tau) \\ Y^k = Y^{k-1} + \delta_k \mathcal{P}_\Omega(M - X^k) \end{cases}$$

until a stopping criterion is satisfied. $\mathcal{P}_\Omega(.)$ is the projection onto Ω while the shrink(Y, τ) is a nonlinear operator which applies the soft-thresholding rule to the singular values of its input matrix (see [45], Section 2 for details). For the right values of τ, the sequence $\{X^k\}$ converges to a solution which approximately minimizes (3.24). For every $k \geq 0, Y^k$ vanishes outside of Ω, hence Y^k is sparse. The sparsity can be exploited to evaluate the shrink function quickly. Further, the X^k have low rank. Therefore, the algorithm has low storage requirements since only the principal factors are stored.

3.7.3 Proximal Gradient Algorithm

Consider the following relaxed version of problem (3.28)

$$\min \mu||B||_* + \mu\lambda||C||_1 + \frac{1}{2}||M - B - C||_F^2, \tag{3.39}$$

where $f(B, C) = \frac{1}{2}||M-B-C||_F^2$ is a penalty function for any violation of the equality constraint and the relaxation parameter μ is greater than zero. As $\mu \to 0$, any solution

of (3.39) tends to the solution set of (3.28). Since $f(B, C)$ is convex, smooth, and has Lipschitz continuous gradient, (3.39) is solvable by proximal gradient algorithms [161]. Iteratively, the algorithm forms separable quadratic approximations to the penalty term $f(B, C)$ at some chosen points $Y^k = (Y_B^k, Y_C^k)$. This is motivated by the fact that in some cases, the iterates (B^k, C^k) have simple or closed form expression. This property is exploited in [171] to develop an iterative thresholding algorithm for Robust Principal Component Analysis (RPCA). Nevertheless, the iterative thresholding algorithm proposed therein requires large number of iterations before it converges. Therefore, it has limited applicability. A significant improvements on the algorithm has been made in [158] for (3.24) by combining the technique for judicious choice of Y^k suggested in [26] with continuation techniques [171]. Ganesh et al. [78] propose the Accelerated Proximal Gradient (APG) algorithm for matrix decomposition problem with a far better performance by extending the soft-thresholding scalar operator

$$\mathcal{S}_\epsilon := \begin{cases} \text{sign}(x)(|x| - \epsilon) & \text{if } |x| > \epsilon \\ 0, & \text{otherwise,} \end{cases} \tag{3.40}$$

to vectors and matrices.

3.7.4 Alternating Direction Method of Multipliers

In general, Alternating Direction Method of Multipliers (ADMM) is a first order but improved Augmented Lagrangian method for solving convex programming with linear constraints. A specialised ADMM algorithm for problem (3.28) is developed in [175]. The Augmented Lagrangian function of (3.28) is given by

$$\mathcal{L}(B, C, Z) := ||B||_* + \lambda||C||_1 - \langle Z, B + C - M\rangle + \frac{\beta}{2}||B + C - M||^2, \tag{3.41}$$

where $Z \in \mathbf{R}^{n_1 \times n_2}$ is the multiplier while β is the penalty parameter. The classical ADMM iterative scheme (see [30], for instance) for this problem is

$$\begin{cases} B^{k+1} \in \arg\min_{B \in \mathbf{R}^{n_1 \times n_2}} \{\mathcal{L}(B, C^k, Z^k)\}, \\ C^{k+1} \in \arg\min_{C \in \mathbf{R}^{n_1 \times n_2}} \{\mathcal{L}(B^{k+1}, C, Z^k)\}, \\ Z^{k+1} = Z^k - \beta(B^{k+1} + C^{k+1} - M), \end{cases}$$

where the triple (B^k, C^k, Z^k) is a given iterate. Yuan and Yang [175] claim that this is equivalent to

$$\begin{cases} 0 \in \partial(||B^{k+1}||_*) - [Z^k - \beta(C^k + B^{k+1} - M)], \\ 0 \in \lambda\partial(||C^{k+1}||_1) - [Z^k - \beta(C^{k+1} + B^{k+1} - M)], \\ Z^{k+1} = Z^k - \beta(B^{k+1} + C^{k+1} - M). \end{cases} \tag{3.42}$$

Define $\mathcal{D}_\tau(X) = U\mathcal{S}_\tau(\Sigma)V^T$, where $X = U\Sigma V^T$ is any SVD of X and $\mathcal{S}_\tau$ is the soft thresholding operator defined in 3.40. The ADMM algorithm [103, 175] for low rank plus sparse matrix recovery is Algorithm 1.

Algorithm 1: Alternating Directions Method for Low Rank plus Sparse Matrix Decomposition

Result: B, C

initialize: $C^0 = Z^0 = 0, \beta > 0$;

while *not converged* **do**

compute $B^{k+1} = \mathcal{D}_{1/\beta}(M - C^k + \frac{1}{\beta}Y^k)$;

compute $C^{k+1} = \mathcal{S}_{\lambda/\beta}(M - B^{k+1} + \frac{1}{\beta}Y^k)$;

compute $Y^{k+1} = Y^k + \beta(M - B^{k+1} - C^{k+1})$;

end

The performance of Algorithm 1 on a wide range of problems is excellent because it requires relatively small numbers of iterations to converge with relatively good accuracy [52]. The dominant cost of each iteration is the computation of B^{k+1} by using singular value thresholding.

Chapter 4

Clique Relaxations

4.1 Introduction

In this chapter, we discuss the classes of clique relaxation models. We outline the properties of each of them and point out their advantages and drawbacks. We present existing solution approach to each of the models.

4.2 Classes of clique relaxations

Clique was introduced by Luce and Perry [109] to model the notion of *cohesive subgroup* in social network analysis. Since then, cliques together with its associated maximum clique problem have been extensively studied in graph theory [35], theoretical computer science [79, 99] and operation research [36, 44] from different perspectives. Cliques posses the ideal structures of a cohesive subgroup. They are endowed with three important properties expected of a cohesive subgroup, namely; familiarity (every members are neighbours and no strangers in the group), reachability (direct communication to and from every member) and robustness (trying to destroy the group by removing members is difficult). Despite the elegance of the clique model, it has been criticized for its high level of restriction [42, 140, 151]. The major drawback of the clique model is the fact that not every application problem requires that every pair of nodes be adjacent. In addition, in most cases, real-life networks are build based on experimental data that is either incomplete or contain errors [105]. Furthermore, since cliques contain paths of length one from each node to every other node, studying its internal structure is not interesting [151].

All the inadequacies enumerated above motivated researchers to develop clique relaxation models to eliminate the drawbacks of the clique model. Each of the relaxations weakens one of the properties of the clique or the other. The clique relaxation models include: the k-clique [108], k-clan [119, 120], k-club [120], k-plex [20, 153], k-core [157], k-defective clique [153] and γ-clique [3, 4, 140, 165]. These relaxations can be categorised, broadly, into three namely: distance/diameter based, degree based and density based. Furthermore, according to the taxonomy in [141], the relaxation can be absolute or relative, standard or weak, structural or statistical. For standard relaxation, the relaxed clique-defining property is required to hold in the induced subgraph, whereas the weak relaxation requires the property to hold in the original graph. Also, in absolute relaxation, the relaxation parameter is an absolute bound on the relaxation while the relative version is defined with respect to the cardinality of the node set and a parameter $\zeta \in (0, 1)$. In addition, in structural relaxation, the relaxation property is satisfied by each node, whereas, in statistical relaxation the desired property is satisfied on average over-all members of the group. The general idea is to characterize a *clique-like* structure with a well defined mathematical definitions and known graph theoretic properties. Over the years, several works have been published on two of these three categories, i.e the diameter/distance based and the degree based relaxations. Clique relaxations based on edge density is relatively new. We discuss the first two classes in this chapter. Density based relaxation forms the main concept in this thesis and is dealt with in the next chapter.

4.3 Distance and Diameter Bases Relaxations

The first form of clique relaxations is the k-clique. It was introduced by Luce [108]. k-clique relaxes the requirement of having an edge between every pair of vertices in the group by allowing them to be of distance at most k apart. A subgraph $H = (V', E')$ of

G is a k-clique if for all $i, j \in V' \subseteq V, d_G(i,j) \leq k$. V' is maximal by inclusion [17]. This implies that no vertex outside of V' is of distance k or less from every vertex in V'. Since H is maximal, any node $w \in V \setminus V'$, there exists $i \in V'$ such that $d_G(w,i) > k$. k-clique is a weak, absolute, structural clique relaxation.

Finding the largest k-clique in a graph, G, is known as the maximum k-clique problem. Suppose the cardinality of the largest k-clique in the graph is $\omega_k(G)$ and is called k-clique number; the following binary integer linear program recovers the maximum k-clique in G [17]:

$$\omega_k = \max \sum_{i \in V} x_i \tag{4.1a}$$

subject to:

$$x_i + x_j \leq 1, \quad \forall\, i,j \in V \ : i < j, d_G(i,j) > k \tag{4.1b}$$

$$x_i \in \{0,1\}, \forall\, i \in V, \tag{4.1c}$$

where $d_G(i,j)$ is the pairwise distance between vertex i and j in V. The constraint (4.1b) makes sure that either i or j is included in the k-clique whenever the distance condition is satisfied. Hence, the feasible solutions will be incidence vectors of k-cliques in G. x_i corresponds to the nodes in V and $x_i = 1$ if the node belongs to the k-clique and 0 otherwise. Finding maximum k-clique in G is equivalent to finding maximum clique in G^k [17]. G^k is the kth power of G. $G^k = (V, E^k)$ where $(i,j) \in E^k \,\forall\, i,j \in V$ if $d_G(i,j) \leq k$. Thus, the solution approaches available for maximum clique problem [36] can similarly be used to solve the maximum k-clique problem.

Recall that a k-clique is defined with respect to the given graph G and not the induced subgraph H. Therefore, if two vertices $i, j \in V$ belong to a k-clique H, then $d_G(i,j) \leq k$. However, the diameter of the induced subgraph $G[H]$ is not necessarily k. In other words, $d_{G[H]}(i,j) \leq k$ does not necessarily hold. Consider Figure 4.1 for

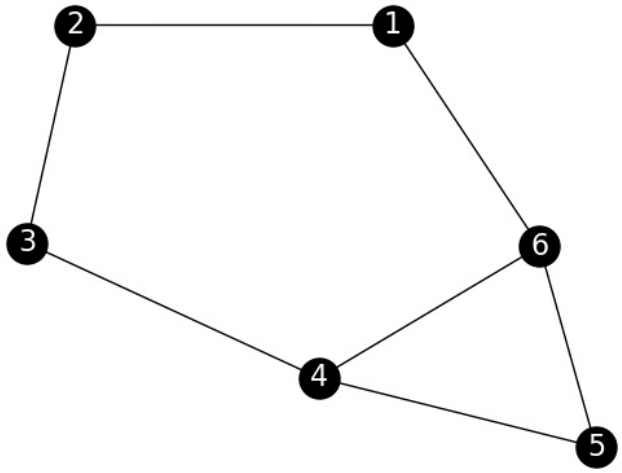


Figure 4.1: A graph illustrating the difference between a 2-clique and a 2-club

example. The vertex set $H_1 = \{1, 3, 4, 5, 6\}$ form a 2-clique, although the diameter of H_1 is 3 which is greater than 2. In fact, a disconnected subgraph can form a k-clique [119]. Hence, k-clique lacks the tightness and connectedness property which is a defining feature of a cluster. This deficiency motivates [6] to come up with the concept of *Sociometric clique* [139]. This is later known as k-clan [119]. H is a k-clan if it is a k-clique such that every pair of nodes $i, j \in H$ are of distance at most k in H; i.e, $d_H(i, j) \leq k$. In other words, a k-clan is a k-clique with diameter k. It is possible for a graph to contain k-cliques but no k-clan. An example is given in Figure 4.2. Figure 4.2 contains a graph with 2-cliques: $H_2 = \{1, 3, 4, 6, 7, 8\}, H_3 = \{2, 3, 4, 5, 6, 7, 8\}$ but neither of them is a 2-clan.

A k-clan, H, that is a maximal subgraph of G is a k-club. The fact that H is a k-club means that $\forall\, w \in G \setminus H$, there exists $u \in H$ such that $d_H(u, w) > k$. The subset of a k-club may not necessarily be a k-club. In other words, k-club lacks *heredity* [141]. A graph property, ϕ, is hereditary on induced subgraphs, if the induced

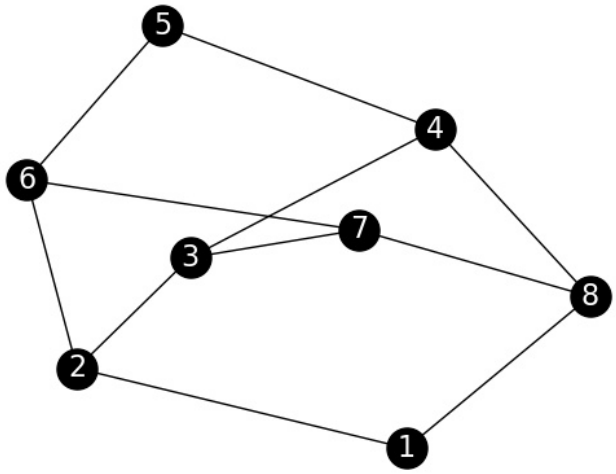


Figure 4.2: A graph with 2-cliques but no 2-clan

subgraphs obtained by deleting any subset of vertices does not violate ϕ. k-club lacks heredity of any type. Due to the fact that k-clubs do not possess any kind of heredity, it is not possible to easily adapt the maximum clique algorithms to the maximum k-club problem. In fact, Pajouh and Balasundaram [134] have shown recently that, for $k \geq 2$, checking for maximality of a k-club is NP-hard. In addition, they gave conditions under which a connected 2-clique will be a 2-club based on the concept of *partitionable* cycle. G is a partitionable cycle, if it contains a spanning cycle, S, and a pair of nonadjacent nodes i and j such that every edge in $E \backslash E(S)$ has one endpoint in $V_S(i,j)$ and the other endpoint in $V_S(j,i)$, where $V_S(i,j)$ and $V_S(j,i)$ are the internal nodes on the two paths between i and j in S. A partitionable cycle is asymmetric if $|V_S(i,j)| \neq |V_S(j,i)|$. If no subset of vertices ν, $5 \leq \nu \leq 2k+1, \forall\, k \geq 2$, induces an asymmetric partitionable cycle in G, then every connected k-clique is a k-club [134]. As a corollary, in a bipartite graph, every connected 2-clique is a 2-club [152]. In this case, checking maximality of a k-club reduces to checking maximality of a connected k-clique.

The maximum k-club problem seeks the k-club with maximum cardinality in a given graph. We first present the general integer programming model for maximum k-club problem and then describe special cases for $k = 2, 3$ that are of highest practical interest and have received much attention in the literature. Let C_{ij}^k be the set of all chains (paths) linking $i, j \in V$, with lenght at most k. Suppose V_p is the vertex set of a chain p. Suppose, further, that $x_i \in \{0, 1\}\ \forall i \in V$ is equal to one if and only if the vertex i belongs to the k-club. Define an auxiliary binary variable $y_p, \forall p \in C = \cup_{ij \in V} C_{ij}^k$. Then, the chain formulation [9, 40] for the k-club problem is the following:

$$\max \sum_{i \in V} x_i, \tag{4.2a}$$

subject to

$$x_i + x_j \leq \sum_{p \in C_{ij}^k} y_p + 1, \quad \forall\, (i, j) \notin E, C_{ij}^k \neq \emptyset, \tag{4.2b}$$

$$x_i + x_j \leq 1,\ \forall\, (i, j) \notin E, C_{ij}^k = \emptyset, \tag{4.2c}$$

$$y_p \leq x_q,\ \forall\, p \in C,\ \forall\, q \in V_p, \tag{4.2d}$$

$$x_i, y_p \in \{0, 1\},\ \forall\, i \in V, p \in C. \tag{4.2e}$$

Constraint (4.2b) guarantees that any non-adjacent pair of nodes i, j in the k-club are linked by at least a chain. Constraint (4.2c) ensures that nodes i, j are not both in the k-club if $d(i, j) > k$. Constraint (4.2d) makes sure that all the nodes of a selected chain belong to the k-club. The last constraint defines x_i and y_p as a binary variable. For $k = 2$ the above chain formulation becomes:

$$\max \sum_{i \in V} x_i, \tag{4.3a}$$

subject to

$$x_i + x_j \leq \sum_{l \in N(i) \cap N(j)} x_l + 1, \quad \forall\, (i,j) \notin E, N(i) \cap N(j) \neq \emptyset \tag{4.3b}$$

$$x_i + x_j \leq 1, \quad \forall\, (i,j) \notin E, N(i) \cap N(j) = \emptyset \tag{4.3c}$$

$$x_i \in \{0,1\},\ \forall\, i \in V. \tag{4.3d}$$

Due to the number of possible distinct paths of length at most k between every pair of nodes, the chain formulation may have a very huge number of variables when k >2. Generally, we have $|C_{ij}^k| = \mathcal{O}(n^{k-1})$ for every pair of nodes. Hence, $|C| = \mathcal{O}(n^{k+1})$ [152]. Therefore, this model does not scale well as k increases. In fact, to solve a small instance of problem with $k \geq 3$ is challenging. To reduce the number of variables in the mathematical formulation for the maximum 3-club, the *neighborhood* formulation was proposed in [8]. This formulation has $|V| + |E|$ number of variables. Observe that a pair of nonadjacent nodes $i, j \in V$ belongs to a 3-club V' if there is a node $t \in V'$ such that $t \in N_i \cap N_j$ or there are two adjacent nodes $u, v \in V'$ such that u is a neighbour of i, and v is a neighbour of j. The first condition holds if and only if $d_{G[V']}(i,j) = 2$. If the first condition does not hold and the second condition holds then $u \in \{N(i) \setminus N(j)\}$ and $v \in \{N(j) \setminus N(i)\}$. We denote the set of edges that connect such intermediate vertices for i and j as E_{ij} and define

$$E_{ij} = \{(u,v) \in E \mid u \in (N(i) \setminus N(j)), v \in (N(j) \setminus N(i))\}, \tag{4.4}$$

for all i, j such that $d_G(i,j) = 3$. Define binary variables x_i and z_{ij} for each node $i \in V$ and edge $(i,j) \in E$, respectively. Then the maximum 3-club problem can be formulated as the following binary program:

$$\max \sum_{i \in V} x_i, \tag{4.5a}$$

subject to

$$x_i + x_j \leq \sum_{l \in N_i \cap N_j} x_l + \sum_{(u,v) \in E_{ij}} z_{uv} + 1, \quad \forall\, (i,j) \notin E, \tag{4.5b}$$

$$z_{ij} \leq x_i, z_{ij} \leq x_j, x_i + x_j \leq z_{ij} + 1, \quad \forall\, (i,j) \in E, \tag{4.5c}$$

$$x_i, z_{ij} \in \{0,1\},\ \forall\, i \in V,\ \forall\, (i,j) \in E. \tag{4.5d}$$

Constraint (4.5b) ensures that two nonadjacent vertices i and j are not both in a 3-club unless a common neighbour is in the 3-club or a pair of their neighbours, u and v, linked by an edge, are in the solution set. Constraint (4.5c) guarantees that (i,j) is used if and only if both its end nodes belong to the solution. [8] also proposed another formulation known as *node cut set* formulation for the maximum 3-club problem. It has order $|V|$ variable but the growth in the number of constraints is exponential [152].

Veremyev and Boginski [163] presented an alternative integer programming formulation for the k-club problem known as the recursive formulation. Consider a set of vertices $V' \subseteq V$ and its characteristic vector x. Define a binary variable $v_{ij}^{(l)}, (i,j = 1,\ldots,n; l = 2,\ldots,k)$ such that $v_{ij}^{(l)} = 1$ if there is at least a path of length l from i to j in $G[V']$ and 0 otherwise. For $l = 2$, we have

$$v_{ij}^{(2)} = \min\{x_i x_j \sum_{s=1}^{n} A_{is} A_{sj} x_s, 1\}, \tag{4.6}$$

where A_{ij} are entries of the adjacency matrix of the input matrix. The linearized version of (4.6) is

$$\begin{cases} v_{ij}^{(2)} \leq x_i, \quad v_{ij}^{(2)} \leq x_j \\ v_{ij}^{(2)} \leq \sum_{s=1}^{n} A_{is}A_{sj}x_s, \quad v_{ij}^{(2)} \geq \frac{1}{n}\left(\sum_{s=1}^{n} A_{is}A_{sj}x_s\right) + (x_i + x_j - 2). \end{cases}$$

$v_{ij}^{(l)}$ for $3 \leq l \leq k$ can recursively be found using

$$v_{ij}^{(l)} = \min\{x_i \sum_{s=1}^{n} v_{sj}^{(l-1)} A_{is}, 1\}, \tag{4.7}$$

with the following linearization

$$\begin{cases} v_{ij}^{(l)} \leq x_i \quad v_{ij}^{(l)} \leq \sum_{s=1}^{n} A_{is} v_{sj}^{(l-1)}, \\ v_{ij}^{(l)} \geq \frac{1}{n}\left(\sum_{s=1}^{n} A_{is} v_{sj}^{(l-1)}\right) + (x_i - 1). \end{cases}$$

Consequently, the maximum k-club can be formulated as the following integer program [163]:

$$\max \sum_{i \in V} x_i, \tag{4.8a}$$

subject to

$$\sum_{l=2}^{k} v_{ij}^{(l)} \geq x_i + x_j - 1, \quad \forall \notin E \tag{4.8b}$$

$$v_{ij}^{(2)} \leq x_i, \quad v_{ij}^{(2)} \leq x_j, \quad \forall\, i, j \in V, i < j \tag{4.8c}$$

$$v_{ij}^{(2)} \leq \sum_{s=1}^{n} A_{is}A_{sj}x_s, \quad \forall\, i, j \in V, i < j \tag{4.8d}$$

$$v_{ij}^{(2)} \geq \frac{1}{n}\left(\sum_{s=1}^{n} A_{is}A_{sj}x_s\right) + (x_i + x_j - 2), \quad \forall\, i, j \in V, i < j \tag{4.8e}$$

$$v_{ij}^{(l)} \leq x_i \quad v_{ij}^{(l)} \leq \sum_{s=1}^{n} A_{is} v_{sj}^{(l-1)}, \quad \forall\, i, j \in V, i < j, 3 \leq l \leq k \tag{4.8f}$$

$$v_{ij}^{(l)} \geq \frac{1}{n}\left(\sum_{s=1}^{n} A_{is} v_{sj}^{(l-1)}\right) + (x_i - 1), \quad \forall\, i, j \in V, i < j, 3 \leq l \leq k, \quad \text{(4.8g)}$$

$$x_i, v_{ij}^{(l)} \in \{0, 1\},\ \forall\, i, j \in V, i < j, l = 2, \ldots, k. \quad \text{(4.8h)}$$

(4.8) has $\mathcal{O}(kn^2)$ variables and constraints. A simpler and more efficient variant of formulation (4.8) is developed in [164]. Moreover, other MIP-based techniques for finding k-clubs have recently been considered in [43, 121]. Among the drawbacks of the 2-club model is that it produces star-like hub-and-spoke structures as maximum-cardinality solutions. An improved model that finds connected 2-clubs has recently been studied in the literature [7, 97].

Due to the correspondence between the maximum clique and the maximum k-clique problem, the heuristic and exact algorithms for the maximum clique problem can be applied to the k^{th} power of the graph to solve the maximum k-clique problem. This is one of the major reasons why maximum k-clique has not been studied extensively. Till date, there is no computational results for the problem [152]. In a similar vein, due to computational intractability, very few exact methods exist for the maximum k-clique problem. However, a number of heuristic algorithms exist for solving the maximum k-club. Three algorithms (DROP, CONSTELLATION and k-CLIQUE & DROP) were proposed in [39]. Among the three, DROP, with order $\mathcal{O}(|V|^3|E|)$ time complexity, was reported to be the most effective for dense graphs. CONSTELLATION, with order $(k(|V| + |E|))$, performs well in finding the maximum k-club in sparse graphs according to the report of the numerical experiments in [39]. The solutions found by k-CLIQUE-DROP are dominated by either DROP or CONSTELLATION in most cases. In addition, since finding the size of maximum k-clique is NP-hard for any fixed k [39, 116], the running time of k-CLIQUE & DROP is not polynomial unless $P = NP$. Another simple heuristic called Iterative DROP (or IDROP for short) was recently pro-

posed in [55]. IDROP is a modified DROP and also has polynomial time complexity. With all the instances of problems considered in [55], the solutions found by IDROP are always as good as those found by DROP and CONSTELLATION or better. However, this is at the cost of more CPU time. Bourjolly et al. [40] present the first exact algorithm for maximum k-club. Their branch-and-bound (B&B) algorithm also uses DROP heuristic to direct its branching process. [55] showed that the time complexity this algorithm is $\mathcal{O}(1.62^n)$.

The k-club improves the reachability among the members of a cohesive subgroups, however, they perform poorly in terms of cohesiveness properties [139]. For example, a star graph[1] posses the structure of a 2-club but has low familiarity and is susceptible to hub disconnection.

4.4 Degree Based Relaxations

One of the popular degree-based clique relaxation models is called k-plex. It was introduced by Seidman and Foster [151]. According to [151], a set of vertices $V' \subseteq V$ forms a k-plex if the minimum degree in the induced subgraph $G[V']$ is at least $|V'|-k$, where $k \geq 1$. The case $k = 1$ is equivalent to clique while $k > 1$ is a relaxation. We illustrate this concept with the aid of Figure 4.3. The set of nodes $\{2, 3, 4, 5\}$ is a 1-plex (a clique), $\{1, 2, 3, 4, 5\}$ is a 2-plex while the entire graph is a 4-plex. A k-plex is maximal if it is not contained in a larger k-plex.

The maximum k-plex problem is to find the largest k-plex in a given graph. Finding a k-plex for any positive integer constant, k, is proved to be NP-complete [20]. [17] proposed a complementary problem to k-plex and called it a co-k-plex. A subgraph H is a co-k-plex if the maximum degree in H is at most $k - 1$. Simply put, H is a

[1] A star graph is a graph with a vertex at the centre, hub, with other vertices linked to the centre but no link with any other vertex

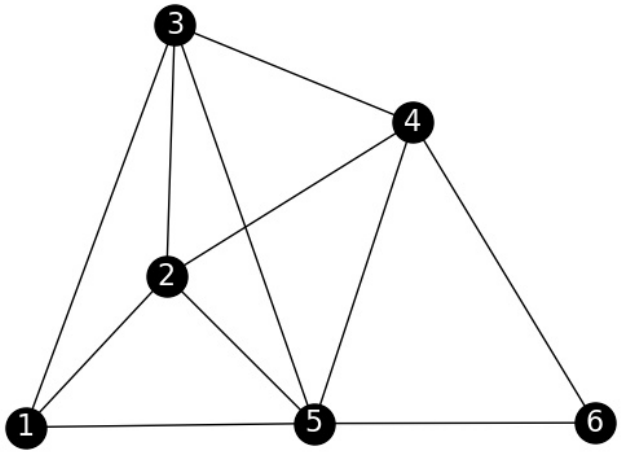


Figure 4.3: A graph that illustrates k-plexes for $k = 1, 2$ and 4

co-k-plex if $\forall\ v \in V', |N(v) \cap V'| \leq k - 1$. A maximal co-$k$-plex is a co-$k$-plex that is not contained in a larger co-k-plex in G. We remark here that H is a co-k-plex in G if and only if H is a k-plex in $\bar{G}$. As a consequence, $H \subseteq G$ is a clique in G if an only if H is an independent set in $\bar{G}$, since a clique is a 1-plex while an independent set is a 1-co-plex. [151] established the following graph theoretic properties of a k-plex. Suppose H is a k-plex, then

1. any subgraph of H is a k-plex
2. if $k < \frac{n+2}{2}$, where n is the number of nodes in H, then $diam(H) \leq 2$
3. $\kappa(H) \geq n - 2k + 2$.

With reference to the taxonomy of clique relaxations defined earlier at the beginning of this chapter, a k-plex is a standard, absolute, structural clique relaxation. We also remark here that k-plexes are endowed with hereditary property. In addition, every k-plex is a $(k + 1)$-plex.

The integer programming formulation of Pattillo et al. [17, 139] is the only known deterministic method for maximum k-plex problem. Let $z_i = deg_{\bar{G}} = |V \setminus N_G(i)|$ be the degree of node i in $\bar{G}$. The following binary program finds the largest k-plex in G

$$\max \sum_{i \in V} x_i, \tag{4.9a}$$

subject to

$$\sum_{j \in V \setminus N_G(i)} x_j \leq (k-1)x_i + z_i(1 - x_i), \quad \forall\, i \in V, \tag{4.9b}$$

$$x_i \in \{0, 1\}, z_i \in \mathbb{N}, \ \forall\, i \in V. \tag{4.9c}$$

Constraint (4.9b) makes sure that node i belonging to the k-plex has at most $k-1$ non-neighbors inside the k-plex.

For any fixed $k \in \mathbb{N}$, the maximum k-plex problem is NP-hard [20]. Nonetheless, heuristics have been developed either to solve the problem or as a subroutine for the exact branch-and-bound method. McClosky and Hicks [115] extended a simple clique finding heuristic to find maximal k-plexes, which was then used as a lower bound for branch-and-bound. If $G = (V, E)$ is a k-plex and $|V| \geq 2k - 2$, then $diam(G) = 2$ [139]. This condition is useful in the development of a branch and bound method for this problem. Most of the exact methods available for solving the maximum k-plex problem adapt either branch and bound or branch and cut. Balasundaram et al. [20] developed a branch and cut algorithm for k-plex using *valid inequalities* based on maximum independent set of size at least k. The authors successfully used the algorithm to solve large-scale real life problem in social networks known as the Erdös graphs. The algorithm was also able to find 2-plex in a dense graph of moderate size. [122] compute k-plexes with maximum cardinality using the duality between maximum

k-plex and d-bounded-degree (d-BDD for short) vertex deletion. The complement of the input graph is computed and d-BDD is solved on the complement graph for $d := k - 1$. The minimum d-BDD-set is then translated back into a maximum k-plex in the input graph. [28] present an efficient algorithm that enumerates all the maximal k-plex in a given graph. Conte et al. [59] recently proposed the idea of *coreness* and *cliqueness* filtering criteria as a mean to recover large k-plexes in a network.

A subgraph induced by $V' \subseteq V$ is a k-core if $\forall\, i, j \in V', deg_{G[V']}(i) = |N(i) \cap V'| \geq k$. That is $G[V']$ is a k-core if the minimum degree of every of its vertices is k. Finding the largest k-core in a graph is known as the maximum k-core problem. This problem is solvable in polynomial time [139]. Indeed, the following simple greedy algorithm can generate optimal solution for this problem. First, the vertex i with the minimum degree, $\delta(G)$, is picked. Check if $\delta(G) \geq k$. If yes, the whole graph, G, is a k-core. If not, delete i, update $G := G - i$ and continue recursively until a maximum k-core or $\emptyset$ is found. Finding the maximum k-core is mostly used as a pre-processing step for solving maximum clique or some other clique relaxation problems. The reason is because some of these structures are guaranteed to be a part of the largest k-core, for a particular k. For instance, we know that a k-plex of size p cannot contain a vertex with degree less than $p - k$. To find such a maximum k-plex, one can solve for maximum $(p - k)$ core first since a k-plex of size p will always be a subset of the largest $(p - k)$-core in a given graph. Using the largest k-core as a preprocessing (or scale-reduction step) for solving another problem is known as *peeling* [139]. Peeling has been successfully applied for solving the maximum clique problem in [3] and the maximum k-plex problem in [20].

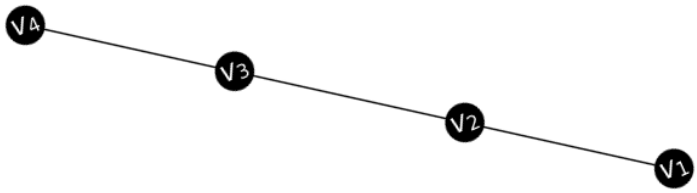


Figure 4.4: A graph illustrating a $(k+1)$-plex that is not a k-defective clique

4.5 Density Based Relaxations

An alternative method for clique relaxation is the density based relaxation. The k-defective clique, the k-densest sugraph and the maximum quasi-clique are density based relaxations.

$H = (V', E')$ is a k-defective clique if $|E'| \geq \binom{V'}{2} - k$, where k is a non-negative integer. In other words, H is a k-defective clique if it differs from a clique by at most k missing edges. A clique is equivalent to a 0-defective clique. This model was introduced by Yu et al. [174]. A maximal k-defective clique is a k-defective clique that is not contained in a larger k-defective clique. k-defective clique can be classified as a standard, absolute structural clique relaxation model. The relationship between a k-defective clique and a k-plex is stated in the following proposition

Proposition 4.5.1. *If $H = (V', E')$ is a k-defective clique in G, then H is a $(k+1)$-plex, but the converse may not hold.*

Proof. From definition, H is a k-defective clique implies that $\delta(H) \geq |V'| - (k+1)$ and the result follows. On the contrary, consider the graph H^* (see Figure 4.4) with vertices $\{v_1, v_2, v_3, v_4\}$ and edges $\{(v_1, v_2), (v_2, v_3), (v_3, v_4)\}$. H^* is a 3-plex since $\delta(H^*) = 1$. However, H^* is not a 2-defective clique. □

The following gives the summary of the analytical properties of a k-defective clique.

Corollary 4.5.1. *Let H, with the vertex set V', be a k-defective clique in G. Then*

1. *Any $k+1$ vertices in H form a dominating set*

2. *Any induced subgraph of H is a k-defective clique*

3. *If $k < \frac{|V'|}{2}$, then $diam(H) \leq 2$*

Item 2 in the corollary above states the hereditary property of the k-defective clique. Since k-defective has hereditary property, developing an enumerative algorithm for the problem becomes easy.

Observe that for small values of k, k-defective clique will be very close to clique and hence can still be too restrictive to be practical. On the other hand, for large values of k, it may lack structure, since it is possible that all missing edges are incident to a very small subset of vertices, which may result in isolated vertices. A k-plex, however, has a bound on the number of non-neighbors for each vertex. Thus when k is small, k-plex provides a very good relaxation because it is less restrictive. In addition, it retains some properties of a clique, e.g, low diameter and high connectivity; for small values of k. Unfortunately, as k increases, this relaxation becomes less useful.

The k-densest subgraph problem can be defined as follows. Given $G = (V, E)$ and a nonnegative integer $k \leq |V|$, the k-densest subgraph problem is to find a subgraph of cardinality k in G with the maximum edge density. This problem is also known as the k-cluster problem [60], the heaviest (unweighted) k-subgraph problem [98] and the maximum edge subgraph problem [15]. This problem has been shown to be NP-hard on bipartite, perfect graphs [60] and planar graphs [94]. Assuming NP does not have subexponential time algorithms, no polynomial-time algorithm exists for k-densest subgraph problem in general. However, it is polynomially solvable if $(|E| = \Omega(|V|^2))$ and $k = \Omega(|V|)$ [14]. Other approximation algorithms also exist (see [18]). Billionnet and Roupin [33] present a deterministic approach for solving the k-densest subgraph. They apply rounding technique to the optimal solution of the linear

programming relaxation of the $0-1$ quadratic program:

$$\text{maximize} \sum_{(i,j)\in E} x_i x_j, \tag{4.10a}$$

subject to

$$\sum_{i\in V} x_i = k, \tag{4.10b}$$

$$x_i \in \{0,1\}, \quad \forall\, i \in V, \tag{4.10c}$$

where $x_i = 1$ if and only if vertex i belongs to the densest subgraph. In addition, various mixed integer programming formulations for this problem are presented in [32].

The second density based clique relaxation model, i.e the quasi-clique, is central to this thesis, hence the next chapter is devoted to it.

Chapter 5

Quasi-Clique

5.1 Introduction

This chapter deals with the maximum quasi-clique recovery. We discuss the computational complexity of the problem and then state, the easy to compute, upper bound for the quasi-clique number. We state our proposition for the quasi-hereditary property of quasi-clique and present an intuitive proof for the proposition. We highlight the existing Mixed Integer Programming (MIP) formulations for quasi-clique recovery and then present our convex relaxation model for the planted quasi-clique problem.

5.2 Maximum Quasi-clique Problem

The most recent but popular among the clique relaxation models is the edge based model called quasi-clique. It is also called γ-quasi-clique or simply γ-clique. This model was proposed by Abello et al. [3] and has been attracting attention since it was proposed (see [138] and the references therein). Let $H \subseteq G$ with the vertex set V' and the edge set E'. H is a γ-clique if $|V' \times V' \cap E| / \binom{|V'|}{2} \geq \gamma$, where $\gamma \in (0, 1]$. Alternatively, H is a γ-clique if $|E'| \geq \frac{\gamma |V'|(|V'|-1)}{2}$, for $\gamma \in (0, 1]$. A 2 parameter variant to this definition is contained in [42]. Quasi-clique is a relative, standard, statistical clique relaxation.

Q_γ is a maximal γ-clique in G if it is a γ-clique and there is no γ-clique $Q'_\gamma \subseteq G$ such that $Q'_\gamma \supseteq Q_\gamma$. The maximum γ-clique problem is to find the largest maximal γ-clique in a given graph. In other words, given an edge density threshold, find the subset of vertices with the largest cardinality such that its induced subgraph satisfies the edge density requirement. We use quasi-clique (γ-clique) to refer to the subgraph

and the subset of vertices, interchangeably. The cardinality of the largest quasi-clique in a given graph is known as γ-*clique number* and we denote it with ω_γ. The size of the largest quasi-clique in binomial random graphs has been proved to be concentrated around some two integers and these integers have been explicitly derived in [21].

5.3 Computational Complexity

In this section, we present the results on the computational complexity for maximum γ-clique problem. To simplify the analysis, γ is replaced with a rational number $\frac{a}{b}$, for given positive integers a, b with $a < b$. Hence, the analysis will be for $\frac{a}{b}$-clique model. Following the approach in [79], the decision version of the problem can be stated as follows: Given a graph $G = (V, E)$ and positive integers a, b and k, does G contain a $\frac{a}{b}$-clique of size at least k?

Proposition 5.3.1 ([140], Proposition 1)**.** *The $\frac{a}{b}$-clique problem is NP-complete for any $a, b > 0, a < b$.*

Proof. Observe that $\frac{a}{b}$-clique belongs to the class NP since it is a generalization of the classical clique. The main idea is to construct an auxiliary graph $G' = (V', E')$, for a given k and $\frac{a}{b}$, and to prove that G has a clique of size k if and only if $G \cup G'$ has a $\frac{a}{b}$-clique of size $|V'| + k$ [2]. The construction is done in the following way. A set of vertices V' with the cardinality $|V'| = 4(|V|^2 + k^2)b - k$ is built and the edges were also constructed to obtain a $2|V|$-regular graph.[3] Edges are arbitrarily placed such that we have $\frac{a}{b}\binom{|V'|+k}{2} - \binom{k}{2}$ number of edges between the $|V'|$ vertices. The expression $\frac{a}{b}\binom{|V'|+k}{2} - \binom{k}{2}$ always returns an integer value since $|V'| + k$ is a multiple of $2b$. $|V'|$

[2] Detailed proof of this proposition is contained in [140]. We have included a brief summary here for completeness.

[3] Constructing such graphs with even regularity can always be achieved by placing all the vertices in a circle and connecting each nodes to its immediate $|V|$ neighbours on each side of the circle.

is sufficiently large to ensure that the inequalities

$$\binom{|V'|}{2} \geq \frac{a}{b}\binom{|V'|+k}{2} - \binom{k}{2} \geq |V||V'| \tag{5.1}$$

hold. The first inequality makes sure that the required number of edges for a $\frac{a}{b}$-clique of size $|V'|+k$ fit in $G \cup G'$, with k vertices coming from a clique in G. The second inequality makes it possible to build a $2|V|$ regular graph on $|V'|$ nodes with desired number of edges.

The proof is completed by showing that G has a clique of size k if and only if $G \cup G'$ contains a $\frac{a}{b}$-clique of size $|V'|+k$. Let a clique C be contained in G. The combined vertices of $G[C]$ and G' will be $|V'|+k$ and the total number of edges will be

$$\frac{a}{b}\binom{|V'|+k}{2} - \binom{k}{2} + \binom{k}{2} = \frac{a}{b}\binom{|V'|+k}{2}, \tag{5.2}$$

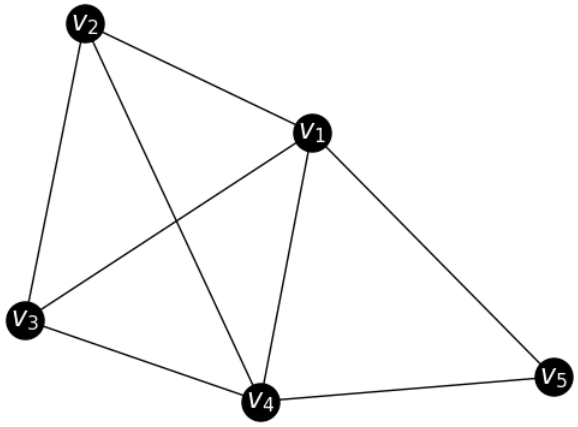
which shows that the vertex set form a $\frac{a}{b}$-clique by definition. On the contrary, suppose $G \cup G'$ contains a $\frac{a}{b}$-clique with $|V'|+k$ vertices. Note that there exists a $\frac{a}{b}$-clique Q' of size $|V'|+k$ in $G \cup G'$. All the vertices of G' are in Q'. Therefore, exactly k vertices come from G contributing $\binom{k}{2}$ edges. If the number of edges from the k vertices from G is not $\binom{k}{2}$, then the set of vertices $|V'|+k$ that form Q' cannot have density $\frac{a}{b}$. □

Corollary 5.3.1 ([140], Corollary 1)**.** *The γ-clique problem is NP-complete, for $\gamma \in (0, 1]$.*

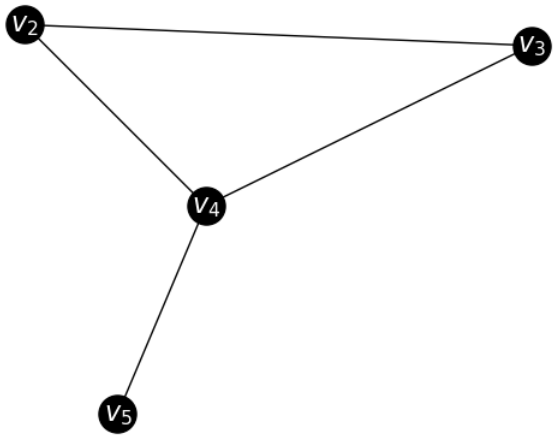
5.4 Quasi-inheritance

The clique has a property that many of the clique finding algorithms exploit. The property is known as downward closure [42] or hereditary property [140]. This means that, if G is a clique, then every induced subgraph of G must also be a clique. Unfortu-

nately, this property is not possessed by quasi-clique. A subgraph of a γ-quasi-clique may not necessarily be a γ-quasi-clique. Consider, for example, the graphs in Figure 5.1. G_1 is a 0.8-quasi-clique but an induced subgraph of G_1; G_1' is not. However, γ-quasi-clique has a property called quasi-hereditary instead of downward closure [140]. That is, given a γ-quasi-clique, the subgraph induced by removing the vertex with the least degree in the vertex set will also be a γ-quasi-clique. In fact, we have the following proposition.



(a) G_1 is a 0.8-clique.



(b) $G_1' = G_1(V - \{v_1\})$ is an induced subgraph of G_1 but not a 0.8-clique

Figure 5.1: Illustration of lack of hereditary property of γ-clique.

Proposition 5.4.1. *Let $Q_\gamma = (V, E)$ be a γ-clique. If $v_{min} \in V$ is the vertex with the minimum degree in Q_γ, then the subgraph $Q_\gamma[V \setminus \{v_{min}\}]$ is a γ-clique.*

Proof. Recall that Q_γ is a γ-clique implies that its edge density $|E_{Q_\gamma}| \geq \gamma\binom{|V|}{2}$.

If a subgraph of Q_γ, induced by removing an arbitrary vertex, v, and all its incidence edges, forms a γ-clique, then:

$$|E_{Q_\gamma[V\setminus\{v\}]}| \geq \gamma\binom{|V|-1}{2} = \left\lceil \frac{\gamma(|V|-1)(|V|-2)}{2} \right\rceil, \tag{5.3}$$

where the function $\lceil . \rceil$ is defined by $\lceil x \rceil = \min\{z \in \mathbb{Z} : z \geq x, \forall\, x \in \mathbb{R}\}$. Note that the average degree of $v_i \in V$ is given by $deg_{Q_\gamma}(v_i) = \lceil \gamma(|V|-1) \rceil$. Therefore, $v_{\min}$ (the vertex with the minimum degree) has at most $\lceil \gamma(|V|-1) \rceil$ number of edges connected to it. This means

$$deg_{Q_\gamma}(v_{\min}) \leq \lceil \gamma(|V|-1) \rceil . \tag{5.4}$$

Removing the edges attached to $v_{\min}$ from Q_γ, we have:

$$|E_{Q_\gamma[V\setminus\{v_{\min}\}]}| \geq \left\lceil \frac{\gamma(|V|(|V|-1))}{2} \right\rceil - \lceil \gamma(|V|-1) \rceil \tag{5.5}$$

$$\geq \left\lceil \frac{\gamma(|V|(|V|-1))}{2} - \frac{2\gamma(|V|-1)}{2} \right\rceil \tag{5.6}$$

$$= \left\lceil \frac{\gamma(|V|-1)(|V|-2)}{2} \right\rceil = |E_{Q_\gamma[V-1]}|. \tag{5.7}$$

The inequality (5.5) follows from Equation (5.4) while (5.6) holds from the the property of the $\lceil . \rceil$ operator. The equality in (5.7) follows from Equation (5.3). This completes the proof. □

5.5 Algorithms for maximum quasi-clique

With the quasi-hereditary property, γ-clique can still be recovered using enumerative algorithms [140]. Majority of the existing works on γ-clique focused on developing heuristics for detection of large quasi-clique. Abello et al. [3] were the first to publish on the maximum quasi-clique problem. They proposed a greedy randomized adaptive search procedure (GRASP) for finding large quasi-clique in graphs generated from communication data. The GRASP for quasi-clique first constructs a clique to serve as a seed. It then grows the quasi-clique using a modified local search procedure. Suppose $\mathcal{Q}$ with cardinality q is the seed. The local search procedure searches for vertices (u, v, w) such that $(u, v) \in E$, $w \in \mathcal{Q}$ and u and v are adjacent to at least $\gamma(q-1)$ vertices of $\mathcal{Q} \setminus \{w\}$. If such vertices are found, then w is removed from $\mathcal{Q}$ and u, v are added. The procedure continues iteratively. A similar approach using a semi-external memory algorithms that handles massive graph together with GRASP so as to be able to deal with graphs with millions of nodes is presented in [4]. A heuristic solution approach to quasi-clique problem by extending *Reactive Local Search* (RLS) and *Dynamic Local Search* (DLS) for MCP to γ-clique problem can be found in [42].

Reactive Local Search [24, 25] works by keeping the current and modifying it with two essential moves namely; node addition and node removal. The search mechanism of RLS is based on Tabu Search [81]. Whenever a node is added/removed from the current clique, it becomes *prohibited* (i.e, cannot be considered for removal/addition) for the next t moves. This search heuristic is complemented by a memory based reactive scheme that automatically adjust the parameter t to suit the problem instance.

Instead of node addition and removal, Dynamic Local Search, on the other hand, modifies the current clique using node addition and plateau moves. Its basic search mechanism is based on penalty values associated with each node. At the initial stage of the search process, a single node is uniformly selected at random. This is set to a

current clique and every other node's penalty is set to zero. The algorithm then proceeds by alternating between the following two search phases: the expansion phase and the plateau phase. In the expansion phase, a node with the minimum penalty from the set of nodes that could be added is selected (ties are broken randomly) and added to the current clique. In the plateau phase however, the nodes in the current clique not connected to the selected node with the minimum penalty are removed.

5.6 Analytical Upper Bound for $\omega_\gamma(G)$

There exists a constant-time computable upper bound on the clique number, $\omega(G)$, for a given graph, proposed by Amin and Hakimi [13]. The following proposition is a generalization of this upper bound to quasi-clique.

Proposition 5.6.1 ([140], Proposition 3)**.** *Let $G = (V, E)$ be a graph with γ-clique number $\omega_\gamma(G)$, then the following holds:*

$$\omega_\gamma(G) \leq \frac{\gamma + \sqrt{\gamma(\gamma + 8|E|)}}{2\gamma}. \tag{5.8}$$

Furthermore, if G is a connected graph, then

$$\omega_\gamma(G) \leq \frac{\gamma + 2 + \sqrt{(\gamma+2)^2 + 8\gamma(|E| - |V|)}}{2\gamma}, \tag{5.9}$$

where $|V|$ and $|E|$ are the cardinalities of the vertex set, and the edge set, respectively.

The following lower bound for clique number can be derived using Motzkin and Straus [123] formulation for maximum clique problem:

$$\omega(G) \geq \frac{1}{1 - \Upsilon}, \tag{5.10}$$

where $\Upsilon = \frac{2|E|}{|V|^2}$. From (5.10), the following relationship exists between $\omega(G)$ and $\omega_\gamma(G)$ [140]:

$$\frac{\omega(G)-1}{\omega(G)} \leq \frac{\omega_\gamma(G)-1}{\omega_\gamma(G)} \leq \frac{\omega(G)-1}{\gamma\omega(G)}. \tag{5.11}$$

5.7 Mathematical Formulations for Quasi-clique

The existing deterministic algorithm for γ-clique recovery is based on MIP formulations, which we now briefly present. The first study on quasi-clique from mathematical perspective was carried out in [140], where γ-clique problem was proved to be NP-complete. Also, an upper bound was derived for maximum γ-clique. For $i \in V$, define $x_i \in \{0,1\}$ such that $x_i = 1$ if and only if $i \in V'$ and 0 otherwise, where V' is the vertex set of the maximum quasi-clique. The following linear MIP formulation was proposed in [140]:

$$\omega_\gamma = \max \sum_{i \in V} x_i, \tag{5.12a}$$

subject to:

$$\sum_{(i,j)\in E} A_{ij}x_ix_j \geq \gamma \sum_{i,j\in V: i<j} x_ix_j \quad \forall\, i,j \in V, \tag{5.12b}$$

$$x_i \in \{0,1\}\ \forall\, i \in V, \tag{5.12c}$$

where A_{ij} is the (i,j)-th entry of the adjacency matrix of the graph. The constraint (5.12b) is non-linear. To linearize it, define $z_{ij} = x_ix_j$. This is equivalent to the following three linear constraints:

$$z_{ij} \leq x_i, \quad z_{ij} \leq x_j, \quad x_i + x_j - 1 \leq z_{ij}. \tag{5.13}$$

Hence, problem (5.12) becomes:

$$\omega_\gamma = \max \sum_{i \in V} x_i, \tag{5.14a}$$

subject to:

$$\sum_{(i,j) \in E} (A_{ij} - \gamma) z_{ij} \geq 0 \quad \forall\, i, j \in V, \tag{5.14b}$$

$$z_{ij} \leq x_i, \quad z_{ij} \leq x_j, \quad x_i + x_j - 1 \leq z_{ij}, \tag{5.14c}$$

$$x_i, z_{ij} \in \{0, 1\}\ \forall\, i, j \in V. \tag{5.14d}$$

An alternative formulation was also presented in [140] by defining a new variable

$$h_i = x_i(\gamma x_i + \sum_{(i,j) \in E} (A_{ij} - \gamma) x_j), \tag{5.15}$$

based on the constraint (5.12b). The resulting MIP model is:

$$\omega_\gamma = \max \sum_{i \in V} x_i, \tag{5.16a}$$

subject to:

$$\sum_{i \in V} h_i \geq 0 \tag{5.16b}$$

$$h_i \leq \nu x_i, \quad h_i \geq -\nu x_i \quad \forall\, i \in V, \tag{5.16c}$$

$$h_i \geq \gamma x_i + \sum_{j \in V} (A_{ij} - \gamma) x_j - \nu(1 - x_i) \quad \forall\, i \in V, \tag{5.16d}$$

$$h_i \leq \gamma x_i + \sum_{j \in V} (A_{ij} + \gamma) x_j - \nu(1 - x_i) \quad \forall\, i \in V, \tag{5.16e}$$

$$x_i \in \{0, 1\} \quad \forall\, i \in V, \tag{5.16f}$$

where, A_{ij} are the entries of the adjacency matrix of the graph, ν is a constant that is large enough and h_i is defined by 5.15. (5.16) can only handle problems with small graph size. Because of this drawback, Veremyev et al. [165] reformulated this model by defining z_{ij} as a binary variable, such that $z_{ij} = 1$ if and only if $(i, j) \in E \cap (V' \times V')$. In addition, a binary variable $s_t, t = 1, \ldots, |V|$, which determines the size of the quasi-clique is defined. This implies that $s_t = 1$ if and only if $|V'| = t$. With these additional variables and notations, the improved MIP model presented in [165] is:

$$\max \sum_{i \in V} x_i, \tag{5.17a}$$

$$\text{subject to} \sum_{(i,j) \in E} z_{ij} \geq \gamma \sum_{t=\omega^l}^{\omega^u} \frac{t(t-1)}{2} s_t, \tag{5.17b}$$

$$z_{ij} \leq x_i, \quad z_{ij} \leq x_j, \ \forall \ (i, j) \in E, \tag{5.17c}$$

$$\sum_{i \in V} x_i = \sum_{t=\omega^l}^{\omega^u} t s_t, \quad \sum_{t=\omega^l}^{\omega^u} s_t = 1, \tag{5.17d}$$

$$x_i \in \{0, 1\}, z_{ij} \geq 0, \ \forall \ i, j \in V, i < j, \tag{5.17e}$$

$$s_t \geq 0, \ \forall \ t \in \{\omega^l, \ldots, \omega^u\}, \tag{5.17f}$$

where ω^l and ω^u are the lower and upper bounds, respectively, on the size of quasi-clique that could be found in the input graph. These lower and upper bounds can be set to 0 and $|V|$, respectively, if no estimates are available. In addition, the lower bound can be set to 1 if the input graph is non-empty or 2 if it has at least one edge. The constraint (5.17b) is the edge density requirement while (5.17c) ensures that $z_{ij} = 1$ if and only if i and j belong to the quasi-clique. Observe that the left hand side of (5.17b) can be written as

$$\sum_{(i,j)\in E} z_{ij} = 1/2 \sum_{i\in V} \sum_{j:(i,j)\in E} x_i x_j = 1/2 \sum_{i\in V} \left(x_i \sum_{j:(i,j)\in E} x_j \right). \tag{5.18}$$

Setting w_i to the quantity in the bracket in equation (5.18) above, (5.17) can be reformulated as [165]:

$$\max \sum_{i\in V} x_i \tag{5.19a}$$

$$\text{subject to } \sum_{i\in V} w_i \geq \gamma \sum_{t=\omega^l}^{\omega^u} t(t-1)s_t, \tag{5.19b}$$

$$w_i \leq \psi_i x_i, \quad w_i \leq \sum_{j:(i,j)\in E} x_j, \ \forall\, i \in V, \tag{5.19c}$$

$$\sum_{i\in V} x_i = \sum_{t=\omega^l}^{\omega^u} t s_t, \quad \sum_{t=\omega^l}^{\omega^u} s_t = 1, \tag{5.19d}$$

$$x_i \in \{0,1\}, z_{ij} \geq 0, \ \forall\, i,j \in V, i < j, , \tag{5.19e}$$

$$s_t \geq 0, \ \forall\, t \in \{\omega^l, \ldots, \omega^u\}, \tag{5.19f}$$

where ψ_i is a parameter that is sufficiently large. In particular, $\psi_i = deg_G(i)$.

Other various solution techniques have recently been developed for maximum quasi-clique problem (see e.g, [113, 118, 149, 177])

5.8 The Planted Quasi-clique Problem

An variant of the MCP is the planted (hidden) clique problem. This problem is applicable in community detection [88], computation of Nash equilibrium [23, 89], cryptographic security [92], etc. The problem can be formulated in two different ways namely: the randomized case and the adversarial case. For the randomized case, a graph of order n is considered and a clique of size n_c ($n_c \leq n$) is inserted using randomly chosen n_c nodes. The remaining pair of nodes are then connected depending on a

given probability p. For the adversarial case, on the contrary, instead of joining the diversionary edges in a probabilistic manner, an adversary is allowed add up $\mathcal{O}(n_c^2)$ edges to the graph. According to Kučera [99], if $n_c = \Omega(c\sqrt{n \log n})$, where n_c is the clique size and c is a constant that is large enough, then the nodes of the planted clique will be those with the largest degree with a very high probability. In this case, the planted clique can easily be recovered. Alon et al. [10] and Feige and Krauthgamer [74] improved this bound by developing an algorithm that can find a clique of size $n_c = \Omega(c\sqrt{n})$ when c is large enough. The algorithm of Alon et al. relies on the spectral properties of the adjacency matrix, A, of the input graph. Since A is symmetric, all its eigenvalues and corresponding eigenvectors are real. Suppose G contains a planted clique, C, of size $n_c = \Omega(c\sqrt{n})$. Alon et al. claim that, when the eigenvectors are arranged in descending order of magnitude (absolute values), the nodes belonging to C will correspond to the first n_c-largest elements of the second largest eigenvector of the adjacency matrix of G. Another work on planted clique recovery, also based on spectral analysis is [124]. Nadakuditi [124] discovered that there is a sharp transition between success and failure in clique detection based on eigen-analysis. Frieze and Kannan [77] solved the planted clique problem by maximizing a cubic form or tensor defined on D. D is a three dimensional array, obtained based on the properties of the input graph. They declared that their method recovers the planted clique inasmuch as the local maximum of the cubic form is attained. Furthermore, they conjectured that the function attains local maximum if $k = \Omega(n^{1/3}(\log n)^c)$, where k is the size of the planted clique, $n = |V|$ and c is a positive constant. Ames and Vavasis [12] on the other hand presented a convex relaxation of this problem using the nuclear norm. Let C be a clique contained in G. The adjacency matrix of C', obtained by taking the union of C and the set of loop of all the nodes of C is a rank-one matrix. The entries of $C' \times C'$ corresponding to the indices of $i \in C$ are equal to one and zero everywhere else.

Therefore, the planted maximum clique recovery in this setting is equivalent to finding the largest rank one submatrix of the adjacency matrix of the input graph. Techniques from matrix completion was adopted in [12] to achieve this, and the following rank minimization problem was presented:

$$\min rank(X) \tag{5.20a}$$

$$\text{subject to} \sum_{i \in V} \sum_{j \in V} X_{ij} \geq n_c^2, \tag{5.20b}$$

$$X_{ij} = 0, \ \forall \ (i,j) \notin E \text{ and } i \neq j, \tag{5.20c}$$

$$X = X^T, \tag{5.20d}$$

$$X_{ij} \in \{0,1\}, \tag{5.20e}$$

where $X \in \mathbf{R}^{n \times n}$ and n_c is the size of the planted clique. The rank minimization problem is known to be NP-hard. Fortunately, the nuclear norm is the convex surrogate of the rank function. The nuclear norm of a matrix is the sum of its singular values. That is, $||X||_* = \sigma_1(X) + \sigma_2(X) + \ldots + \sigma_n(X)$, where $\sigma_i(X), i \in \{1, \ldots, n\}$, are the singular values. Hence, (5.20) becomes:

$$\min ||X||_*, \tag{5.21a}$$

$$\text{subject to} \sum_{i \in V} \sum_{j \in V} X_{ij} \geq n_c^2, \tag{5.21b}$$

$$X_{ij} = 0, \ \forall \ (i,j) \notin E \text{ and } i \neq j, \tag{5.21c}$$

$$X = X^T, \tag{5.21d}$$

$$X_{ij} \in [0,1]. \tag{5.21e}$$

In our case, we adopt the technique from matrix decomposition (see Section 3.5.3) to recover the planted quasi-clique in a graph. The planted quasi-clique problem is a more

difficult problem than the planted clique. This is because the latter is a special case of the former. Our proposed formulation is the following:

$$\min ||X||_* + \lambda ||\hat{X}||_1, \tag{5.22a}$$

$$\text{subject to } \sum_i \sum_j X_{ij} \geq \gamma \eta^2, \tag{5.22b}$$

$$X + \hat{X} = A, \tag{5.22c}$$

$$X_{ij}, \hat{X}_{ij} \in [0, 1], \quad \eta \in \mathbb{N}, \tag{5.22d}$$

where $X, \hat{X} \in \mathbf{R}^{n \times n}$ are matrix variables corresponding to the quasi-clique and the diversionary edges, λ is a parameter, A is the adjacency matrix of the input graph, $\mathbb{N}$ is the set of natural number while the parameter $\gamma \in (0, 1]$ is the desired edge density of the quasi-clique to be recovered. The constraint (5.22b) ensures that the solution satisfies the edge density requirement, while (5.22c) makes sure that the decomposition agrees with the input matrix. η is a positive integer valued variable that determines the size of the recovered quasi-clique. Indeed, our formulation (5.22) for planted quasi-clique mirrors the rank sparsity decomposition problem since the setup and the output of the quasi-clique problem is similar to that of rank-sparsity decomposition, i.e, our matrix is sampled using Bernoulli model which has been shown to satisfy the incoherence conditions required by the rank-sparsity problem and our expected solution is a low rank matrix.

Since we are only interested in X, we can eliminate constraint (5.22c) and write $\hat{X} = A - X$. Therefore, (5.22) can be reformulated as

$$\min ||X||_* + \lambda ||A - X||_1, \tag{5.23a}$$

$$\text{subject to } \sum_i \sum_j X_{ij} \geq \gamma\eta^2, \tag{5.23b}$$

$$X_{ij} \in [0,1], \quad \eta \in \mathbb{N}. \tag{5.23c}$$

The semidefinite programming formulation for (5.23) is the following

$$\begin{aligned}
&\text{minimize } \frac{1}{2}(trace(Z_1) + trace(Z_2)) + \lambda \mathbf{1}_n^T W \mathbf{1}_n, \\
&\text{subject to } \begin{bmatrix} Z_1 & X \\ X^T & Z_2 \end{bmatrix} \succeq 0, \\
&\quad - W_{ij} \leq A_{ij} - X_{ij} \leq W_{ij}, \quad \forall\, ij, \\
&\quad \sum_i \sum_j X_{ij} \geq \gamma\eta^2, \\
&\quad X_{ij} \in [0,1], \quad \eta \in \mathbb{N}.
\end{aligned} \tag{5.24}$$

Problems (5.22) - (5.23) are convex optimization problems that can be solved using one of the available convex optimization solvers. With the formulation above, Theorem 3.5.3.2 specializes to the following:

Theorem 5.8.1. *Suppose $Q_\gamma = (V', E')$ is an n_c-vertex γ-clique contained in a graph $G = (V, E)$ of n-vertices with the adjacency matrix A. Let $X^* \in \mathbf{R}^{n \times n}$. If the conditions of Theorem 3.5.3.2 holds with $p = \gamma$, where γ is the edge probability within the quasi-clique, then X^* is the unique optimal solution of (5.23) and Q_γ is the unique maximum γ-clique in G.*

Theorems 3.5.3.2 and 5.8.1 provide the conditions, under which (5.23) will successfully recover a planted quasi-clique. The proof of these Theorems is provided in the next chapter.

Chapter 6

Theoretical Guarantee for Planted Maximum Quasi-Clique Recovery

6.1 Introduction

In this chapter, we present the proof of our main result, Theorems 3.5.3.2 and 5.8.1. The main steps follow the general idea in the low-rank matrix recovery literature [48, 49, 52, 58, 102]. The steps involved include constructing a dual matrix Q, which certifies the optimality of (B^*, C^*) for the convex problem (3.28). This dual certificate must obey some subgradient-type conditions. One of these conditions is that the spectral norm of Q, i.e $||Q||$, must be small. In many of the previous approaches, $||Q||$ is bounded by the matrix l_∞ norm. Q can be decomposed to $Q = U\Sigma V$. Therefore, by virtue of Equation (3.22), there is a relationship between the bound on the dual certificate and the incoherence condition. We bound the norm of the dual certificate by the l_∞ and $l_{\infty,2}$ norm. The $l_{\infty,2}$ was first used in [57] to derive a tighter bound in the case of matrix completion. Ours is an extension of this concept to the matrix decomposition setting.

6.2 Incoherence Property for Matrix $l_{\infty,2}$ Norm

The $l_{\infty,2}$ norm is defined on a matrix M as:

$$||M||_{\infty,2} := \max\left\{\max_i \sqrt{\sum_b M_{ib}^2}, \max_j \sqrt{\sum_a M_{aj}^2}\right\}. \tag{6.1}$$

It is noteworthy that for any matrix $Z \in \mathbf{R}^{n_1 \times n_2}$, $||Z||_{\infty,2} \leq \sqrt{\max\{n_1, n_2\}}||Z||_{\infty}$ [57]. Therefore, from the incoherence property, we have

$$||UV^T||_{\infty,2} \leq \sqrt{\max\{n_1, n_2\}}||UV^T||_{\infty} \leq \sqrt{\frac{\mu r}{\min\{n_1, n_2\}}},$$

or

$$||UV^T||_{\infty,2} \leq \sqrt{n}||UV^T||_{\infty} \leq \sqrt{\frac{\mu r}{n}}, \tag{6.2}$$

for a square matrix.

There is a norm similar to the $l_{\infty,2}$ norm which is the $l_{2,\infty}$ norm defined as $||M||_{2,\infty} := \max_{\mathbf{x}\neq 0} \frac{||M\mathbf{x}||_{\infty}}{||\mathbf{x}||_2}$. The $l_{2,\infty}$ is the maximum Euclidean norm of the rows of M while the $l_{\infty,2}$ is the maximum of both the rows and column norm of M [57, 144]. These norms yield a tighter bound on the entries of a matrix than the commonly used ones [53, 57]. We show that the $l_{\infty,2}$ norm yields a tighter bound on the norm of the dual matrix. This is achieved by expressing the bound as a sum of the l_{∞} and $l_{\infty,2}$ norm instead of the previously derived bounds in matrix decomposition literature which use the l_{∞} norm only. This is one of the main contributions of this thesis. In addition, we adopt some novel simplifying ideas different from the existing works.

The $l_{2,\infty}$ norm has been used previously in [110] to prove that *adjacency spectral embedding* clusters graphs perfectly, for some given stochastic block model. It was also used in [53] to derive a new Procrustean matrix decomposition [4]. More so, $l_{\infty,2}$ and $l_{2,\infty}$ were used in [144] to derive a bound for any random matrix with independent and identically distributed entries, with mean zero and unit variance.

[4]The orthogonal Procrustes problem in linear algebra is about matrix approximation. In its basic form, given two matrices A and B, one is required to find an orthogonal matrix C, which closely maps A to B.

6.3 Background to the Proof

Since every graph has a square adjacency matrix, our proof is for square matrices only. The arguments follow easily for rectangular case. We denote universal constants that does not depend on the parameters of the problem (i.e n, r, μ, etc) with c, c', c_1, c_2, etc. By with high probability, we mean with probability at least $1 - c_1 n^{-c_2}, c_1, c_2 > 0$.

Furthermore, we will make use of linear operators which act on the space of matrices. We denote these operators by calligraphic letters, e.g $\mathcal{P}_\Gamma(X)$. In addition, we let Γ represent the linear space of matrices with their support ($supp(.)$) contained in Γ, by abuse of notation. Therefore, $\mathcal{P}_{\Gamma^\perp}$ is the projection onto the space of matrices with support on Γ^C. By implication, $\mathcal{P}_\Gamma + \mathcal{P}_{\Gamma^\perp} = \mathcal{I}$, the identity operator.

Suppose $B_0 \in \mathbf{R}^{n \times n}$ is of rank r and has the singular value decomposition $U\Sigma V^T$, with $U, V \in \mathbf{R}^{n \times r}$. The subgradient of the nuclear norm at B_0 is of the form:

$$UV^T + Q,$$

where $U^T Q = 0, QV = 0$ and $||Q|| \leq 1$. In addition, we define T, the linear space of matrices that share the same row space or column space as B_0, as

$$T := \{UX^T + YV^T : X, Y \in \mathbf{R}^{n \times r}\}, \tag{6.3}$$

and $T^\perp$ is its orthogonal complement. Clearly, $B_0 \in T$ holds always. Recall that the orthogonal projection of Q onto T is given as

$$\begin{aligned}
\mathcal{P}_T Q &= \mathcal{P}_U Q + Q\mathcal{P}_V - \mathcal{P}_U Q \mathcal{P}_V \\
&= UU^T Q + QVV^T - UU^T QVV^T \\
&= 0.
\end{aligned}$$

This implies that $\mathcal{P}_{T^\perp}Q = Q$, since $\mathcal{P}_T Q + \mathcal{P}_{T^\perp}Q = Q$. For any matrix Z,

$$\mathcal{P}_{T^\perp}Z = (I - UU^T)Z(I - VV^T), \tag{6.4}$$

where $I - UU^T$ and $I - VV^T$ are the orthogonal projections onto the orthogonal complement of the linear space spanned by the columns of U and V, respectively. As a result of this, for any matrix Z, $||\mathcal{P}_{T^\perp}Z|| \leq ||Z||$. We will make use of this fact several times later.

The subgradient of l_1- norm at C_0, where $\Gamma = supp(C_0)$, is of the form:

$$Sgn(C_0) + D$$

where $\mathcal{P}_\Gamma D = 0$ and $||D||_\infty \leq 1$

Definition 6.3.1. *Given a matrix M, M' is a trimmed form of M if the support of M' is contained in the support of M, i.e, $supp(M') \subset supp(M)$ and $M'_{ij} = M_{ij}$ whenever $M'_{ij} \neq 0$.*

Simply put, we get a trimmed version of M by setting few entries of its entries to zero. The following theorem establishes that if a low-rank plus sparse decomposition of $M_0 = B_0 + C_0$ is exact, then it is also exact for $M'_0 = B_0 + C'_0$, where C'_0 is a trimmed version of C_0.

Theorem 6.3.1 (Theorem 2.2 of [52])**.** *Suppose problem (3.28) has a unique and exact solution with input $M_0 = B_0 + C_0$. If $M'_0 = B_0 + C'_0$, where C'_0 is a trimmed version of C_0, then the solution to (3.28) is also unique and exact with input M'_0.*

The proof if this theorem can be found in [52].

6.3.1 The Bernoulli Model and Derandomization

Bernoulli Model

Rather than showing that Theorem (3.5.3.2) holds with Γ sampled uniformly, where Γ is a random subset such that $\Gamma = \{(i,j) : C_{ij} \neq 0\}$ of cardinality k, it is easier to prove Theorem 3.5.3.2 for Γ sampled according to the Bernoulli model, with $\Gamma = \{(i,j) : \Delta_{ij} = 1\}$, where Δ_{ij}'s are independent and identically distributed Bernoulli random variables. $\Delta_{ij} = 1$ with probability ϱ and 0 with probability $1 - \varrho$. Hence, ϱn^2 is the expected cardinality of Γ. By doing this, we exploit the statistical independence of measurements. Based on the arguments presented in [48, 49, 52], any theoretical guarantee proved for Bernoulli model is also valid for the uniform model and the converse holds as well. Henceforth, we will write $\Gamma \sim Bern(\varrho)$, for short, to mean that Γ follows Bernoulli distribution with parameter ϱ.

Note that the sign of the entries of C^* in Theorem 3.5.3.2 is fixed. Surprisingly, it is more convenient to prove the theorem under a stronger condition. We assume that the sign of C^*_{ij}, for $C^*_{ij} \neq 0$, are independent symmetric Bernoulli random variables which assume the value 1 or -1 with probability $1/2$. The probability of recovering C^* with random sign on the support set of Γ is at least the same with C^* with fixed sign. This randomization technique was invented in [52] and has been previously used in [128] and [102]. The supporting theorem is stated formally here.

Lemma 6.3.1.1 (Theorem 2.3 of [52])**.** *Suppose B_0 obeys the conditions of Theorem (3.5.3.2) and that the positions of the non-zero entries of C_0 follow the Bernoulli model with parameter 2ϱ, with the signs of C_0 independent and identically distributed ± 1 with probability $1/2$. Then if the solution to problem (3.28) is exact with high probability, it is also exact for the model with fixed signs and location sampled from the Bernoulli model with parameter ϱ with at least the same probability.*

6.4 Subgradient Condition for Optimality

Now, we state a sufficient condition for the pair (B^*, C^*) to be the unique optimal solution to problem (3.28). The conditions are stated in terms of the dual matrix, whose existence certifies optimality. These conditions are given in the following lemma which is similar to Proposition 2 of [54] and Lemma 2.4 of [52].

Lemma 6.4.1 (Proposition 2 of [54], Lemma 2.4 of [52])**.** *Let* $M = B_0 + C_0$*. Suppose* $\Gamma \cap T = \{0\}$ *(i.e,* $||\mathcal{P}_\Gamma \mathcal{P}_T|| < 1$*). Then* $(B^*, C^*) = (B_0, C_0)$ *is the unique optimal solution to (3.28) if there exists a pair of dual matrix* (D, Q) *such that*

$$UV^T + Q = \lambda(Sgn(C_0) + D),$$

with $\mathcal{P}_T Q = 0, ||Q|| < 1, \mathcal{P}_\Gamma D = 0$ *and* $||D||_\infty < 1$.

Proof. To prove this proposition, we first show that (B_0, C_0) is an optimal pair to the problem (3.28) and then show that it is unique. From the optimality condition base on the subgradient condition at (B_0, C_0), there must exist a dual matrix, $\mathcal{F}$, which simultaneously belong to the subdifferential of the norms of B_0 $(\partial||B_0||_*)$ and C_0 $(\partial||C_0||_1)$. The second condition of Lemma 6.4.1 shows that such matrix exists. Hence (B_0, C_0) is an optimal pair. To show uniqueness, we consider a perturbation of (B_0, C_0), i.e, $(B_0 + \nu_B, C_0 + \nu_C)$, which is also a minimizer. Since $B_0 + C_0 = B_0 + \nu_B + C_0 + \nu_C, \nu_B + \nu_c$ must be equal to zero. Now, applying the subgradient condition, we have:

$$\begin{aligned} ||B_0 + \nu_B||_* + \lambda||C_0 + \nu_C||_1 &\geq ||B_0||_* + \lambda||C_0||_1 + \langle UV^T + Q_0, \nu_B\rangle \\ &\quad + \lambda\langle Sgn(C_0) + D_0, \nu_C\rangle \\ &= ||B_0||_* + \lambda||C_0||_1 + \Delta + \delta. \end{aligned} \tag{6.5}$$

Recall that $\mathcal{F} \in \partial||B_0||_*$ and $\mathcal{F} \in \partial||C_0||_1 \implies$ there exists Q and D such that $\mathcal{F} = UV^T + Q = \lambda(Sgn(C) + D)$. Therefore, it follows that

$$\begin{aligned}\Delta &= \langle UV^T + Q_0, \nu_B\rangle = \langle UV^T + \mathcal{P}_{T^\perp}(Q_0), \nu_B\rangle \\ &= \langle \mathcal{F} - \mathcal{P}_{T^\perp}(Q) + \mathcal{P}_{T^\perp}(Q_0), \nu_B\rangle \\ &= \langle \mathcal{P}_{T^\perp}(Q_0) - \mathcal{P}_{T^\perp}(Q), \nu_B\rangle + \langle \mathcal{F}, \nu_B\rangle.\end{aligned}$$

Likewise,

$$\begin{aligned}\delta &= \lambda\langle Sgn(C_0) + D_0, \nu_C\rangle = \langle \lambda Sgn(C_0) + \lambda\mathcal{P}_{\Gamma^C}(D_0), \nu_C\rangle \\ &= \langle \mathcal{F} - \lambda\mathcal{P}_{\Gamma^C}(D) + \lambda\mathcal{P}_{\Gamma^C}(D_0), \nu_C\rangle \\ &= \langle \lambda\mathcal{P}_{\Gamma^C}(D_0) - \lambda\mathcal{P}_{\Gamma^C}(D), \nu_C\rangle + \langle \mathcal{F}, \nu_C\rangle \\ &= \lambda\langle \mathcal{P}_{\Gamma^C}(D_0) - \mathcal{P}_{\Gamma^C}(D), \nu_C\rangle + \langle \mathcal{F}, \nu_C\rangle.\end{aligned}$$

Hence,

$$\begin{aligned}\Delta + \delta &= \langle \mathcal{P}_{T^\perp}(Q_0) - \mathcal{P}_{T^\perp}(Q), \nu_B\rangle + \langle \mathcal{F}, \nu_B\rangle + \lambda\langle \mathcal{P}_{\Gamma^C}(D_0) - \mathcal{P}_{\Gamma^C}(D), \nu_C\rangle \\ &\quad + \langle \mathcal{F}, \nu_C\rangle \\ &= \langle \mathcal{P}_{T^\perp}(Q_0) - \mathcal{P}_{T^\perp}(Q), \nu_B\rangle + \lambda\langle \mathcal{P}_{\Gamma^C}(D_0) - \mathcal{P}_{\Gamma^C}(D), \nu_C\rangle + \langle \mathcal{F}, \nu_B + \nu_C\rangle \\ &= \langle \mathcal{P}_{T^\perp}(Q_0) - \mathcal{P}_{T^\perp}(Q), \nu_B\rangle + \lambda\langle \mathcal{P}_{\Gamma^C}(D_0) - \mathcal{P}_{\Gamma^C}(D), \nu_C\rangle.\end{aligned}$$

Therefore, equation (6.5) becomes:

$$\begin{aligned}||B_0 + \nu_B||_* + \lambda||C_0 + \nu_C||_1 \geq ||B_0||_* + \lambda||C_0||_1 + \langle \mathcal{P}_{T^\perp}(Q_0) - \mathcal{P}_{T^\perp}(Q), \nu_B\rangle + \\ \lambda\langle \mathcal{P}_{\Gamma^C}(D_0) - \mathcal{P}_{\Gamma^C}(D), \nu_C\rangle \\ = ||B_0||_* + \lambda||C_0||_1 + (||\mathcal{P}_{T^\perp}(Q_0)|| - ||\mathcal{P}_{T^\perp}(Q))\,(||\nu_B||_*) +\end{aligned}$$

$$\lambda \left(||\mathcal{P}_{\Gamma^C}(D_0)||_\infty - ||\mathcal{P}_{\Gamma^C}(D)||_\infty \right) \left(||\nu_C||_1 \right).$$

Since (Q_0, D_0) is any subgradient of $||B||_* + \lambda||C||_1$ at (B_0, C_0), we can choose $\mathcal{P}_{T^\perp}(Q_0)$ and $\mathcal{P}_{\Gamma^C}(D_0)$ freely inasmuch as they satisfy the following conditions:

$$||\mathcal{P}_{T^\perp}(Q_0)|| \leq 1, \text{ and } ||\mathcal{P}_{\Gamma^C}(D_0)||_\infty \leq 1.$$

Letting $\mathcal{P}_{\Gamma^C}(D_0) = Sgn(\mathcal{P}_{\Gamma^C}(\nu_C))$ implies that $||\mathcal{P}_{\Gamma^C}(D_0)||_\infty = 1$. Also, since the dual of nuclear norm is the operator norm, there exists a matrix Q_0 with $||\mathcal{P}_{T^\perp}(Q_0)|| = 1$ such that $\langle Q_0, \nu_B \rangle = ||\mathcal{P}_{T^\perp}\nu_B||_*$. Therefore, we have

$$\begin{aligned} ||B_0 + \nu_B||_* + \lambda||C_0 + \nu_C||_1 \geq ||B_0||_* + \lambda||C_0||_1 + (1 - ||\mathcal{P}_{T^\perp}(Q)||)(||\mathcal{P}_{T^\perp}(\nu_B)||_*) \\ + \lambda(1 - ||\mathcal{P}_{\Gamma^C}(D)||_\infty)(||\mathcal{P}_{\Gamma^C}(\nu_C)||_1). \end{aligned}$$

Since $||\mathcal{P}_{T^\perp}(Q)|| < 1$ and $||\mathcal{P}_{\Gamma^C}(D)|| < 1$, the last two terms in the equation above are strictly positive except if $||\mathcal{P}_{T^\perp}(\nu_B)||_* = 0$ and $||\mathcal{P}_{\Gamma^C}(\nu_C)||_1 = 0$. Hence, $||B_0+\nu_B||_* + \lambda||C_0 + \nu_C||_1 = ||B_0||_* + \lambda||C_0||_1$ if and only if $\mathcal{P}_{T^\perp}(\nu_B) = 0$ and $\mathcal{P}_{\Gamma^C}(\nu_C) = 0$. Note that $\mathcal{P}_{T^\perp}(\nu_B) = \mathcal{P}_{\Gamma^C}(\nu_C) = 0$ implies that $\mathcal{P}_T(\nu_B) + \mathcal{P}_\Gamma(\nu_C) = 0$, since $\nu_B + \nu_C = 0$. So that $\mathcal{P}_T(\nu_B) = -\mathcal{P}_\Gamma(\nu_C) = 0$ (since $\Gamma \cap T = \{0\}$). This implies that $\nu_B = \nu_C = 0$. Therefore, $||B_0+\nu_B||_* + \lambda||C_0+\nu_C||_1 > ||B_0||_* + \lambda||C_0||_1$ except if $\nu_B = \nu_C = 0$. $\square$

Consequently, to prove exact recovery, it suffices to derive a dual certificate Q which satisfies:

(a) $Q \in T^\perp$,

(b) $||Q|| < 1$,

(c) $\mathcal{P}_\Gamma(UV^T + Q) = \lambda Sgn(C_0)$,

(d) $||\mathcal{P}_{\Gamma^\perp}(UV^T + Q)||_\infty < \lambda$.

However, relaxing on the constraint $\mathcal{P}_\Gamma(UV^T + Q) = \lambda Sgn(C_0)$ yields a somewhat different certificate with high probability. This relaxation was introduced in [85] and used in [52] previously. This leads to the following lemma.

Lemma 6.4.2 (Lemma 2.5 of [52]). *Suppose* $||\mathcal{P}_\Gamma \mathcal{P}_T|| \leq 1/2$ *and* $\lambda < 1$. *Then* (B^*, C^*) *is the unique optimal solution of (3.28) if there exists a pair of dual matrix* (Q, D)*, such that*

$$UV^T + Q = \lambda(Sgn(C_0) + F)$$

with $\mathcal{P}_T Q = 0,\ ||Q|| \leq 1/2,\ \mathcal{P}_\Gamma D = 0,\ ||F||_\infty \leq 1/2,$ *and* $||\mathcal{P}_\Gamma F|| \leq 1/4$.

Interested reader can check the proof of this Lemma in [52]. As a result of Lemma 6.4.2, it is sufficient to derive a dual matrix Q, which obeys

$$\begin{aligned} &Q \in T^\perp, \\ &||Q|| < 1/2, \\ &||\mathcal{P}_\Gamma(UV^T - \lambda Sgn(C_0) + Q)||_F \leq \lambda/4, \\ &||\mathcal{P}_{\Gamma^\perp}(UV^T + Q)||_\infty < \lambda/2. \end{aligned} \tag{6.6}$$

6.5 Construction of Dual Certificate

We need to construct a dual matrix Q satisfying the conditions in (6.6). Q will be constructed using a modified version of the *Golfing Scheme* [52, 57, 58, 85]. Set $k_0 = 20\lceil \log n \rceil$. We decompose the observed entries of Γ into independent entries, so that we sample with respect to a different batch at every step. Recall that by initial setup, $\Gamma \sim Bern(p)$, which implies that $\Gamma^C \sim Bern(1-p)$. Now, we think of Γ^C as a union of k_0 independent samples, so that $\Gamma^C = \bigcup_{1 \leq k \leq k_0} \Gamma_k$, with each Γ_k following the Bernoulli model with parameter q (probability of sampling for each batch). This gives

the following Binomial model:

$$\mathbb{P}\left((i,j) \in \Gamma\right) = \mathbb{P}(Bin(k_0, q) = 0) = (1-q)^{k_0}, \tag{6.7}$$

and the two models are equivalent if $p = (1-q)^{k_0}$. Hence, this Γ has the same distribution with the initial model. The Bin in equation (6.7) means binomial. Because of the intersections between the ${\Gamma_k}'s$ $q \geq \frac{(1-p)}{k_0}$ [52].

We now proceed to construct a dual certificate $Q = Q_B + Q_C$. Details about the two components of Q are as follow.

Constructing Q_B using Golfing Scheme

Let k_0 and $\Gamma_k, k \in [1, k_0]$, be as defined above. Also, recall that $\Gamma^C = \cup_{1 \leq k \leq k_0} \Gamma_k$. Starting with $Y_0 = 0$ and proceeding inductively, define

$$Y_k = Y_{k-1} + p^{-1}\mathcal{P}_{\Gamma_k}\mathcal{P}_T(UV^T - Y_{k-1}), \tag{6.8}$$

and set

$$Q_B = \mathcal{P}_{T^\perp} Y_{k_0}. \tag{6.9}$$

Constructing Q_C using Least Square Method

Suppose $||\mathcal{P}_\Gamma \mathcal{P}_T|| < 1/2$, then $||\mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma|| < 1/4$. Thus, the operator $\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma$ is invertible. We write $(\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1}$ for the inverse and set

$$Q_C = \lambda \mathcal{P}_{T^\perp}(\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1} Sgn(C_0). \tag{6.10}$$

There is an alternative definition of (6.10) using the convergent Neumann series [50, 52]. The definition is as follows:

$$Q_C = \lambda \mathcal{P}_{T^\perp} \sum_{k \geq 0} (\mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^k Sgn(C_0). \tag{6.11}$$

Observe that

$$\begin{aligned}
\mathcal{P}_\Gamma Q_C &= \lambda \mathcal{P}_\Gamma \mathcal{P}_{T^\perp} (\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1} Sgn(C_0) \\
&= \lambda \mathcal{P}_\Gamma (\mathcal{I} - \mathcal{P}_T)(\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1} Sgn(C_0) \\
&= \lambda (\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)(\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1} Sgn(C_0) \qquad \heartsuit \\
&= \lambda Sgn(C_0).
\end{aligned}$$

$\heartsuit$ follows from properties of the projection operators, i.e for any two projection operators $\mathcal{P}_1$ and $\mathcal{P}_2$, $\mathcal{P}_1 = \mathcal{P}_1^2$ while $\mathcal{P}_1 \mathcal{P}_2 = \mathcal{P}_2 \mathcal{P}_1$ must hold for the product $\mathcal{P}_1 \mathcal{P}_2$ to be a projector [16]. Consequently, one can verify that among all the matrices $Q \in T^\perp$ which satisfies $\mathcal{P}_\Gamma Q = \lambda Sgn(C_0)$, Q_C is the one with minimum Frobenius norm [52]. Since, by construction, $Q_B, Q_C \in T^\perp$ and $\mathcal{P}_\Gamma Q_C = \lambda Sgn(C_0)$, it remains to show that $Q = Q_B + Q_C$ obeys:

$$\begin{aligned}
&||Q_B + Q_C|| < 1/2 \\
&||\mathcal{P}_\Gamma (UV^T + Q_B)||_F \leq \lambda/4 \\
&||\mathcal{P}_{\Gamma^\perp} (UV^T + Q_B + Q_C)||_\infty < \lambda/2
\end{aligned}$$

to establish (6.6). Indeed, our approach yields a better bound on Q. Our new results is proposed in the following lemma.

Lemma 6.5.1. *Suppose* $\Gamma \sim Bern(p)$ *such that* $p \leq p_u$ *with* $p_u > 0$. *Let* $k_0 = 20\lceil \log n \rceil$, *then under the assumption of Theorem 3.5.3.2, the matrix* Q_B *satisfies:*

i. $||Q_B|| < 1/8$,

ii. $||\mathcal{P}_{\Gamma}(UV^T + Q_B)|| < \lambda/8$,

iii. $||\mathcal{P}_{\Gamma^{\perp}}(UV^T + Q_B)||_{\infty} < \lambda/4$.

Furthermore, if $supp(C_0) = \Gamma$ *(*Γ *is as sampled earlier at the beginning of Section 6.5), and the signs of* C_0 *are symmetric* iid*, then when the assumptions of Theorem 3.5.3.2 holds, the matrix* Q_C *satisfies:*

iv. $||Q_C|| < 1/8$,

v. $||\mathcal{P}_{\Gamma^{\perp}} Q_C||_{\infty} < 1/4$.

6.6 Key Lemmas

We now state some important Lemmas which are used to establish our results, Theorem 3.5.3.2. Our proof differs here, substantially, from the existing works. We derive bound on Q in terms of $l_{\infty,2}$ norm. This is done with the aid of the following two Lemmas. The first lemma is used to bound the operator norm of $(p^{-1}\mathcal{P}_{\Gamma} - \mathcal{I})Z$, for a matrix Z, in terms of the $l_{\infty,2}$ and l_{∞} norms of Z. The bound produced this way is tighter [57] than the previous ones [52, 58].

Lemma 6.6.1 (Lemma 2 of [57])**.** *Let* Z *be a square matrix of dimension* n*. For a universal constant* $c > 1$*, we have*

$$||(p^{-1}\mathcal{P}_{\Gamma} - \mathcal{I})Z|| \leq c\left(\frac{\log n}{p}||Z||_{\infty} + \sqrt{\frac{\log n}{p}}||Z||_{\infty,2}\right),$$

with high probability.

The next Lemma provides for further controls on the $l_{\infty,2}$ norm.

Lemma 6.6.2 (Lemma 3 of [57]). *Suppose $Z \in \mathbf{R}^{n\times n}$ is a fixed matrix. If $p \geq c_0 \frac{\mu r \log n}{n}$, for some $c_0 > 0$ which is large enough, then*

$$||(\mathcal{P}_\Gamma - p^{-1}\mathcal{P}_T\mathcal{P}_\Gamma)Z||_{\infty,2} \leq 1/2\sqrt{\frac{n}{\mu r}}||Z||_\infty + 1/2||Z||_{\infty,2},$$

with high probability.

The proof of these two lemmas can be found in [57]. We also need the following standard results which enable us to manipulate the l_∞ norm.

Lemma 6.6.3 (Lemma 4 of [57], Lemma 3.1 of [52], Lemma 13 of [58]). *Suppose $Z \in \mathbf{R}^{n\times n}$ is a fixed matrix and that $\Gamma_0 \sim Bern(p)$. If $p \geq c_0 \frac{\mu r \log n}{n}$, for some $c_0 > 0$ large enough, then*

$$||(\mathcal{P}_T - p^{-1}\mathcal{P}_T\mathcal{P}_\Gamma\mathcal{P}_T)Z||_\infty \leq 1/2||Z||_\infty,$$

with high probability.

Lemma 6.6.4 (Theorem 6.3 of [49], Lemma 3.2 of [52]). *Suppose $Z \in \mathbf{R}^{n\times n}$ is a fixed matrix and that $\Gamma_0 \sim Bern(p)$. If $p \geq c_0 \frac{\mu r \log n}{n}$, for some $c_0 > 0$ sufficiently large, then with high probability,*

$$||(p^{-1}\mathcal{P}_{\Gamma_0} - \mathcal{I})Z|| \leq c_0'\sqrt{\frac{n\log n}{p}}||Z||_\infty,$$

for some small numerical constant $c_0' > 0$.

Lemma 6.6.5 (Theorem 4.1 of [49], Theorem 2.6 of [52], Lemma 1 of [57]). *Suppose $p \geq c_0 \frac{\mu r \log n}{n}$ for some sufficiently large c_0, then (with high probability);*

$$||\mathcal{P}_T - p^{-1}\mathcal{P}_T\mathcal{P}_\Gamma\mathcal{P}_T|| \leq 1/2.$$

We will also need the following results on bounds of the operator norm of random matrices.

Lemma 6.6.6 (Corollary 2.3.5 of [156]). *Let $Z \in \mathbf{R}^{n\times n}$ be a random matrix with Z_{ij} being independent and identically distributed random variables. Suppose Z_{ij} is uniformly bounded in magnitude by one and has mean zero. Then there exist constants $c, c^* > 0$ such that*

$$\mathbb{P}(||Z|| > \varpi\sqrt{n}) \leq c^* \exp(-c\varpi n)$$

for all $\varpi \in \mathbf{R}$ greater than or equal to c^*.

As a consequence, we have $||Z|| \leq \varpi\sqrt{n}$ with high probability.

Definition 6.6.1 (Definition 5.1 of [167], Nets). *Let (X, d) be a metric space and $\delta > 0$. A subset $\mathcal{N}_\delta$ of X is called a δ-net of X if every point $x \in X$ can be approximated to within δ by some point $y \in \mathcal{N}_\delta$, such that $d(x, y) \leq \delta$.*

Lemma 6.6.7 (Lemma 5.2 of [167]). *The unit Euclidean sphere[5] $\mathbb{S}^{n-1}$ equipped with the Euclidean metric satisfies, for every $\delta > 0$, $\mathcal{N}(\mathbb{S}^{n-1}, \delta) \leq \left(1 + \frac{2}{\delta}\right)^n$.*

Lemma 6.6.8 (Lemma 5.3 of [167]). *Let $Z \in \mathbf{R}^{n\times n}$ be a matrix, and suppose that $\mathcal{N}_\delta$ is a $\delta - net$ of $\mathbb{S}^{n-1}$ for some $\delta \in [0, 1)$. Then*

$$||Z|| \leq (1 - \delta)^{-2} \sup_{x,y\in\mathcal{N}_\delta} \langle y, Zx\rangle.$$

Equipped with these Lemmas, we are now ready to prove Lemma 6.5.1.

6.7 Proof of Lemma 6.5.1

Proof of (i.)

[5]The unit Euclidean $(n-1)$-sphere is defined as $\mathbb{S}^{n-1} = \{x \in \mathbf{R}^n : ||x|| = 1\}$

Proof. Note that from the construction of the dual certificate, $Q = Q_B + Q_C$. Also, from Equation (6.8), $Y_k = Y_{k-1} + p^{-1}\mathcal{P}_{\Gamma_k}\mathcal{P}_T(UV^T - Y_{k-1})$ and $Y_0 = 0$. We set $Z_k = UV^T - \mathcal{P}_T Y_k$ for $k = 0, \ldots, k_0$; so that $Z_0 = UV^T$ and

$$Z_k = (\mathcal{P}_T - p^{-1}\mathcal{P}_T\mathcal{P}_{\Gamma_k}\mathcal{P}_T)Z_{k-1}. \tag{6.12}$$

Hence,

$$Y_{k_0} = \sum_{k=1}^{k_0} p^{-1}\mathcal{P}_{\Gamma_k} Z_{k-1}. \tag{6.13}$$

Observe that $Z_k \in T$, therefore $\mathcal{P}_{T^\perp} Z_k = 0$. Since $Q_B = \mathcal{P}_{T^\perp} Y_{k_0}$ (see Equation 6.9), then

$$\begin{aligned}
||Q_B|| = ||\mathcal{P}_{T^\perp} Y_{k_0}|| &\leq \sum_{k=1}^{k_0} ||p^{-1}\mathcal{P}_{T^\perp}\mathcal{P}_{\Gamma_k} Z_{k-1}|| \\
&= \sum_{k=1}^{k_0} ||\mathcal{P}_{T^\perp}(p^{-1}\mathcal{P}_{\Gamma_k} Z_{k-1} - Z_{k-1})||^{\diamondsuit} \\
&\leq \sum_{k=1}^{k_0} ||p^{-1}\mathcal{P}_{\Gamma_k} Z_{k-1} - Z_{k-1}|| \\
&= \sum_{k=1}^{k_0} ||(p^{-1}\mathcal{P}_{\Gamma_k} - \mathcal{I}) Z_{k-1}|| \\
&\leq c \sum_{k=1}^{k_0} \left(\frac{\log n}{p} ||Z_{k-1}||_\infty + \sqrt{\frac{\log n}{p}} ||Z_{k-1}||_{\infty,2} \right)^{\spadesuit} \\
&\leq c \sum_{k=1}^{k_0} \left(\frac{n}{c_0 \mu r} ||Z_{k-1}||_\infty + \sqrt{\frac{n}{c_0 \mu r}} ||Z_{k-1}||_{\infty,2} \right)^{\clubsuit} \\
&\leq c \sum_{k=1}^{k_0} \left(\frac{n}{\sqrt{c_0 \mu r}} ||Z_{k-1}||_\infty + \sqrt{\frac{n}{c_0 \mu r}} ||Z_{k-1}||_{\infty,2} \right)^{\dagger} \\
&\leq \frac{c}{\sqrt{c_0}} \sum_{k=1}^{k_0} \left(\frac{n}{\sqrt{\mu r}} ||Z_{k-1}||_\infty + \sqrt{\frac{n}{\mu r}} ||Z_{k-1}||_{\infty,2} \right).
\end{aligned} \tag{6.14}$$

The expression in the first line is as a result of Equation 6.13. $\diamondsuit$ follows from the fact that $\mathcal{P}_{T^\perp} Z_k = 0$. $\spadesuit$ is application of Lemma 6.6.1. $\clubsuit$ holds for $p \geq \frac{c_0 \mu r \log n}{n}$ while $\dagger$ is valid since we can choose c_0 such that $c_0 \mu r > 1$ so that $\frac{n}{c_0 \mu r} \leq \frac{n}{\sqrt{c_0 \mu r}}$ holds. We will now bound $||Z_{k-1}||_\infty$ and $||Z_{k-1}||_{\infty,2}$ by applying Lemma 6.6.3. Using (6.12), applying Lemma 6.6.3 repeatedly (replacing Γ with Γ_k), we have (with high probability);

$$\begin{aligned} ||Z_{k-1}||_\infty &= ||(\mathcal{P}_T - p^{-1}\mathcal{P}_T\mathcal{P}_{\Gamma_{k-1}}\mathcal{P}_T)\dots(\mathcal{P}_T - p^{-1}\mathcal{P}_T\mathcal{P}_{\Gamma_1}\mathcal{P}_T)Z_0||_\infty \\ &= \left|\left|\prod_{k=1}^{k-1}(\mathcal{P}_T - p^{-1}\mathcal{P}_T\mathcal{P}_{\Gamma_k}\mathcal{P}_T)Z_0\right|\right|_\infty \\ &\leq (1/2)^{k-1}||UV^T||_\infty, \end{aligned} \tag{6.15}$$

since $Z_0 = UV^T$. In like manner, we apply Lemma 6.6.2 to $||Z_{k-1}||_{\infty,2}$, using Equation (6.12) again and replacing Γ with Γ_k to get, with high probability,

$$\begin{aligned} ||Z_{k-1}||_{\infty,2} &= ||(\mathcal{P}_T - p^{-1}\mathcal{P}_T\mathcal{P}_{\Gamma_{k-1}}\mathcal{P}_T)Z_{k-2}||_{\infty,2} \\ &\leq 1/2\sqrt{\frac{n}{\mu r}}||Z_{k-2}||_\infty + 1/2||Z_{k-2}||_{\infty,2} \end{aligned} \tag{6.16}$$

Combining (6.15) and (6.16) and using (6.12) repeatedly, then

$$||Z_{k-1}||_{\infty,2} \leq k(1/2)^{k-1}\sqrt{\frac{n}{\mu r}}||UV^T||_\infty + (1/2)^{k-1}||UV^T||_{\infty,2}. \tag{6.17}$$

Substituting (6.15) and (6.17) back into (6.14), we get;

$$\begin{aligned} ||Q_B|| &\leq \frac{c}{\sqrt{c_0}}\sum_{k=1}^{k_0}\left(\frac{n}{\sqrt{\mu r}}(1/2)^{k-1}||UV^T||_\infty + \right. \\ &\left.\sqrt{\frac{n}{\mu r}}\left(k(1/2)^{k-1}\sqrt{\frac{n}{\mu r}}||UV^T||_\infty + (1/2)^{k-1}||UV^T||_{\infty,2}\right)\right) \\ &= \frac{c}{\sqrt{c_0}}\sum_{k=1}^{k_0}\left(\frac{n}{\sqrt{\mu r}}(k+1)(1/2)^{k-1}||UV^T||_\infty + \sqrt{\frac{n}{\mu r}}(1/2)^{k-1}||UV^T||_{\infty,2}\right) \end{aligned}$$

$$\leq \frac{c}{\sqrt{c_0}}\frac{n}{\sqrt{\mu r}}||UV^T||_\infty \sum_{k=1}^{k_0}(k+1)(1/2)^{k-1} + \frac{c}{\sqrt{c_0}}\sqrt{\frac{n}{\mu r}}||UV^T||_{\infty,2}\sum_{k=1}^{k_0}(1/2)^{k-1}$$
$$\leq \frac{6c}{\sqrt{c_0}}\frac{n}{\sqrt{\mu r}}||UV^T||_\infty + \frac{2c}{\sqrt{c_0}}\sqrt{\frac{n}{\mu r}}||UV^T||_{\infty,2}.$$

From (3.22) and (6.2), $||UV^T||_\infty \leq \frac{\sqrt{\mu r}}{n}$ and

$$||UV^T||_{\infty,2} = \max\left\{\max_i ||U^T e_i||^2, \max_j ||V^T e_j||^2\right\} \leq \sqrt{\frac{\mu r}{n}}.$$

So, we have

$$\begin{aligned}||Q_B|| = ||\mathcal{P}_{T^\perp} Y_{k_0}|| &\leq \frac{6c}{\sqrt{c_0}} + \frac{2c}{\sqrt{c_0}} \\ &= \frac{8c}{\sqrt{c_0}} \leq 1/8,\end{aligned}$$

provided $c_0 \geq (64c)^2$. □

Proof of (ii.)

Proof.

$$\begin{aligned}\mathcal{P}_\Gamma(UV^T + Q_B) &= \mathcal{P}_\Gamma(UV^T + \mathcal{P}_{T^\perp} Y_{k_0}) \\ &= \mathcal{P}_\Gamma(UV^T + Y_{k_0} - \mathcal{P}_T Y_{k_0}) \\ &= \mathcal{P}_\Gamma(UV^T - \mathcal{P}_T Y_{k_0})^\bigstar \\ &= \mathcal{P}_\Gamma(UV^T - \mathcal{P}_T(Y_{k_0-1} + p^{-1}\mathcal{P}_{\Gamma k_0}\mathcal{P}_T(UV^T - \mathcal{P}_T Y_{k_0-1}))) \\ &= \mathcal{P}_\Gamma(UV^T - \mathcal{P}_T Y_{k_0-1} - p^{-1}\mathcal{P}_T\mathcal{P}_{\Gamma_{k_0}}\mathcal{P}_T(UV^T - \mathcal{P}_T Y_{k_0-1})) \\ &= \mathcal{P}_\Gamma(Z_{k_0-1} - p^{-1}\mathcal{P}_T\mathcal{P}_{\Gamma_{k_0}}\mathcal{P}_T Z_{k_0-1}) \\ &= \mathcal{P}_\Gamma(\mathcal{P}_T - p^{-1}\mathcal{P}_T\mathcal{P}_{\Gamma_{k_0}}\mathcal{P}_T)Z_{k_0-1} = \mathcal{P}_\Gamma(Z_{k_0}).\end{aligned}$$

$\mathcal{P}_\Gamma Y_{k_0} = 0$ (since $\Gamma \cap T = \{0\}$ from the conditions of Lemma 6.4.1), therefore $\bigstar$ holds. The last equality in the last line above follows from Equation (6.12). Hence,

$$||\mathcal{P}_\Gamma(UV^T + Q_B)||_F \leq ||Z_{k_0}||_F = ||\mathcal{P}_T - p^{-1}\mathcal{P}_T\mathcal{P}_{\Gamma_{k_0}}\mathcal{P}_T)||.||Z_{k_0-1}||_F. \tag{6.18}$$

Notice that by the conditions of Theorem 3.5.3.2, $p \geq \frac{c_0 \mu r \log n}{n}$ and Γ_k is independent of Z_{k-1}. Using the recurrence relation (6.12) again and applying Lemma 6.6.5 repeatedly (replacing Γ with Γ_k), we get with high probability;

$$\begin{aligned} ||\mathcal{P}_\Gamma(UV^T + Q_B)||_F &\leq (1/2)^{k_0}||UV^T||_\infty \\ &= (1/2)^{k_0}\sqrt{\frac{\mu r}{n^2}} \\ &< \frac{1}{8\sqrt{n}} = \frac{\lambda}{8}, (\lambda = \frac{1}{\sqrt{n}}) \end{aligned} \tag{6.19}$$

The second line holds from Equation (3.22) while the last inequality is valid since $(1/2)^{k_0}$ tends to zero as k_0 increases. $\square$

Proof of (iii.)

Proof. We are required to show that $||\mathcal{P}_{\Gamma^C}(UV^T + Q_B)||_\infty < \lambda/4$. Observe that $Z_k = UV^T - \mathcal{P}_T Y_k$ implies that $UV^T = Z_k + \mathcal{P}_T Y_k$. Hence,

$$\begin{aligned} UV^T + Q_B &= Z_{k_0} + \mathcal{P}_T Y_{k_0} + \mathcal{P}_{T^\perp} Y_{k_0} \\ &= Z_{k_0} + Y_{k_0}. \end{aligned}$$

We know that the support of Y_{k_0} is Γ^C and, by virtue of (6.18) and (6.19), we have shown that $||Z_{k_0}||_F = ||\mathcal{P}_T - p^{-1}\mathcal{P}_T\mathcal{P}_{\Gamma_{k_0}}\mathcal{P}_T)Z_{k_0-1}||_F < \lambda/8$. Therefore, to show that $||\mathcal{P}_{\Gamma^C}(UV^T + Q_B)||_\infty = ||Z_{k_0} + Y_{k_0}||_\infty < \lambda/4$, it is sufficient to show that $||Y_{k_0}||_\infty < \lambda/8$. Recall from (6.13) that $Y_{k_0} = \sum_{k=1}^{k_0} p^{-1}\mathcal{P}_{\Gamma_k} Z_{k-1}$. It then follows that

$$
\begin{aligned}
||Y_{k_0}||_\infty &\le p^{-1} \sum_{k=1}^{k_0} ||\mathcal{P}_{\Gamma_k} Z_{k-1}||_\infty \\
&\le p^{-1} \sum_{k=1}^{k_0} ||Z_{k-1}||_\infty \\
&\le p^{-1}(1/2)^{k_0} ||UV^T||_\infty \\
&\le p^{-1}(1/2)^{k_0} \frac{\sqrt{\mu r}}{n} \\
&\le \frac{n}{c_0 \mu r \log n}(1/2)^{k_0} \frac{\sqrt{\mu r}}{n} \\
&\le \frac{1}{c_0 \sqrt{\mu r} \log n}(1/2)^{k_0} < \frac{1}{8\sqrt{n}} = \frac{\lambda}{8}.
\end{aligned}
$$

The same argument for the last line of (6.19) is applicable for the last inequality here as well. □

Proof of (iv.)

Proof. From equation (6.11),

$$
\begin{aligned}
Q_C &= \lambda \mathcal{P}_{T^\perp} \sum_{k \ge 0} (\mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^k Sgn(C_0) \\
&= \lambda \mathcal{P}_{T^\perp} Sgn(C_0) + \lambda \mathcal{P}_{T^\perp} \sum_{k \ge 1} (\mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^k Sgn(C_0).
\end{aligned}
$$

We define $\Phi = Sgn(C_0)$ with the following probability distribution:

$$
\Phi_{ij} = \begin{cases} 1 & \text{with probability } p/2, \\ 0 & \text{with probability } 1-p, \\ -1 & \text{with probability } p/2. \end{cases}
$$

Then,

$$Q_C = \lambda \mathcal{P}_{T^\perp}\Phi + \lambda \mathcal{P}_{T^\perp} \sum_{k\geq 1} (\mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^k \Phi$$
$$:= \mathcal{P}_{T^\perp} Q_C^0 + \mathcal{P}_{T^\perp} Q_C^1. \tag{6.20}$$

Therefore, to derive a bound for Q_C, we only need to derive a bound for $\mathcal{P}_{T^\perp} Q_C^0$ and $\mathcal{P}_{T^\perp} Q_C^1$. Applying Lemma 6.6.6 to the first term of (6.20), the following holds with high probability;

$$||\mathcal{P}_{T^\perp} Q_C^0|| \leq |\lambda|||\Phi||$$
$$\leq \lambda c^* \sqrt{pn} = c^* \sqrt{p}.$$

Recall that $\lambda = \frac{1}{\sqrt{n}}$, hence the last part of the equation above follows. For the second term of (6.20), let $\mathcal{R} = \mathcal{P}_{T^\perp} \sum_{k\geq 1} (\mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^k$, so that $||Q_C^1|| = |\lambda|||\mathcal{R}(\Phi)||$. We are required to find a bound on the operator norm of $\mathcal{R}(\Phi)$. We make use of Lemmas 6.6.7 and 6.6.8 to achieve this. Let $\mathcal{N}_\delta$ be a $\delta - net$ for a sphere $\mathbb{S}^{n-1}$ of size at most $(3/\delta)^n$. Then by Lemma 6.6.8,

$$||\mathcal{R}(\Phi)|| \leq (1-\delta)^{-2} \max_{x,y \in \mathcal{N}_\delta} \langle y, \mathcal{R}(\Phi) x \rangle$$

We define a random variable $Y(x, y)$ such that

$$Y(x,y) := \langle y, \mathcal{R}(\Phi) x \rangle$$
$$= \langle \mathcal{R}(yx^*), \Phi \rangle \tag{6.21}$$

for a fixed pair $(x, y) \in \mathcal{N}_\delta \times \mathcal{N}_\delta$ with unit norm. (6.21) holds because $\mathcal{R}$ is a self-adjoint operator [52]. Note that the support of Φ is Γ and Φ is a symmetric matrix with independent and identically distributed random signs. Using Hoeffding's Inequality,

conditioning on Γ, we have

$$\mathbb{P}(|Y(x,y)| > \tau|\Gamma) \leq 2\exp\left(\frac{-2\tau^2}{||\mathcal{R}(yx^*)||_F^2}\right),$$

for $\tau \geq 0$. Since (yx^*) is a unit-normed vector, $||yx^*||_F = 1$. Therefore, $||\mathcal{R}(yx^*)||_F \leq ||\mathcal{R}||$ holds. So, we have

$$\mathbb{P}\left(\max_{x,y\in\mathcal{N}_\delta} |Y(x,y)| > \tau|\Gamma\right) \leq 2|\mathcal{N}_\delta|^2 \exp\left(\frac{-2\tau^2}{||\mathcal{R}||^2}\right).$$

$$\begin{aligned}
||\mathcal{R}|| &= ||\mathcal{P}_{T^\perp} \sum_{k\geq 1} (\mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^k|| \\
&\leq \sum_{k\geq 1} ||(\mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^k|| = \sum_{k\geq 1} ||(\mathcal{P}_\Gamma \mathcal{P}_T)||^{2k} \\
&= \frac{||\mathcal{P}_\Gamma \mathcal{P}_T||^2}{1 - ||\mathcal{P}_\Gamma \mathcal{P}_T||^2}.
\end{aligned}$$

The equality in the immediate equation before the last line above holds with the assumption that the product of $\mathcal{P}_T$ and $\mathcal{P}_\Gamma$ commute and for the fact that $\mathcal{P}_\Gamma^2 = \mathcal{P}_\Gamma$. Considering the event $\psi \coloneqq \{||\mathcal{P}_\Gamma \mathcal{P}_T|| \leq \beta\}$,

$$||\mathcal{R}|| \leq \frac{||\mathcal{P}_\Gamma \mathcal{P}_T||^2}{1 - ||\mathcal{P}_\Gamma \mathcal{P}_T||^2} = \frac{\beta^2}{1-\beta^2}$$

Hence,

$$\begin{aligned}
\mathbb{P}\left(||\mathcal{R}(\Phi)|| > \tau|\Gamma\right) &\leq 2(3/\delta)^{2n} \exp\left(\frac{-2\tau^2}{((1-\delta)^{-2}(\frac{\beta^2}{1-\beta^2}))^2}\right) \\
&= 2(3/\delta)^{2n} \exp\left(\frac{-2\tau^2(1-\beta^2)^2(1-\delta)^4}{\beta^4}\right),
\end{aligned}$$

and marginally,

$$\mathbb{P}\left(||\mathcal{R}(\Phi)|| > \tau\right) \leq 2(3/\delta)^{2n} \exp\left(\frac{-2\tau^2(1-\beta^2)^2(1-\delta)^4}{\beta^4}\right) + \mathbb{P}(||\mathcal{P}_\Gamma \mathcal{P}_T|| \geq \beta).$$

Consequently,

$$\mathbb{P}\left(\lambda||\mathcal{R}(\Phi)|| > \tau\right) \leq 2(3/\delta)^{2n} \exp\left(\frac{-2\tau^2(1-\beta^2)^2(1-\delta)^4}{\lambda^2\beta^4}\right) + \mathbb{P}(||\mathcal{P}_\Gamma \mathcal{P}_T|| \geq \beta),$$

with $\lambda = \frac{1}{\sqrt{n}}$. Conclusively,

$$||Q_C|| \leq c^*\sqrt{p} + |\lambda|||\mathcal{R}(\Phi)|| \leq 1/8,$$

with high probability, provided that β is small enought and for an appropriate choice of c^* and δ. □

Proof of (v.)

Proof. Recall that $Q_C = \lambda \mathcal{P}_{T^\perp}(\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1} Sgn(C_0)$. Therefore,

$$\begin{aligned}
\mathcal{P}_{\Gamma^\perp} Q_C &= \lambda \mathcal{P}_{\Gamma^\perp} \mathcal{P}_{T^\perp}(\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1} Sgn(C_0) \\
&= \lambda \mathcal{P}_{\Gamma^\perp}(\mathcal{I} - \mathcal{P}_T)(\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1} Sgn(C_0) \\
&= \lambda[\mathcal{P}_{\Gamma^\perp}(\mathcal{P}_T \mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1} - \mathcal{P}_T(\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1}] Sgn(C_0).
\end{aligned}$$

Since the operator $\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma$ maps Γ onto itself, $(\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1}$ is in Γ and $\mathcal{P}_{\Gamma^\perp}(\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1} = 0$. So, we have

$$\mathcal{P}_{\Gamma^\perp} Q_C = -\lambda \mathcal{P}_{\Gamma^\perp} \mathcal{P}_T (\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1} \Phi, \tag{6.22}$$

where $\Phi = Sgn(C_0)$. Now, for $(i,j) \in \Gamma^C$,

$$\begin{aligned} Q_{C_{ij}} &= \langle e_i, Q_C e_j \rangle \\ &= \langle e_i e_j^T, Q_C \rangle \\ &= \lambda \langle W(i,j), \Phi \rangle, \end{aligned}$$

where $W(i,j)$ is the matrix $-(\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1} \mathcal{P}_\Gamma \mathcal{P}_T (e_i e_j^T)$. Since $\Gamma = supp(\Phi)$, the signs of Φ are independent, symmetric and identically distributed. Applying Hoeffding's inequality, we have

$$\mathbb{P}(|Q_{C_{ij}}| > \tau\lambda|\Gamma) \leq 2exp\left(\frac{-2\tau^2}{||W(i,j)||_F^2}\right),$$

therefore,

$$\mathbb{P}(\sup_{i,j} |Q_{C_{ij}}| > \tau\lambda|\Gamma) \leq 2n^2 exp\left(\frac{-2\tau^2}{\sup_{i,j} ||W(i,j)||_F^2}\right).$$

For a matrix of the form $e_i e_j^T$, from (6.4), we have

$$||\mathcal{P}_{T^\perp} e_i e_j^T||_F^2 = ||(\mathcal{I} - UU^T)e_i||^2 ||(\mathcal{I} - VV^T)e_j||^2 \geq \left(1 - \frac{\mu r}{n}\right)^2. \tag{6.23}$$

The last inequality is based on (3.21) with the assumption that $\frac{\mu r}{n} \leq 1$. From the fact that $||\mathcal{P}_T e_i e_j^T||_F^2 + ||\mathcal{P}_{T^\perp} e_i e_j^T||_F^2 = 1$, we deduce

$$||\mathcal{P}_T e_i e_j^T||_F \leq \sqrt{\frac{2\mu r}{n}}, \tag{6.24}$$

so that

$$\begin{aligned} ||\mathcal{P}_\Gamma \mathcal{P}_T (e_i e_j^T)||_F &\leq ||\mathcal{P}_\Gamma \mathcal{P}_T|| ||\mathcal{P}_T (e_i e_j^T)||_F \\ &\leq \beta \sqrt{\frac{2\mu r}{n}}, \end{aligned}$$

on the event $\{||\mathcal{P}_\Gamma \mathcal{P}_T|| \leq \beta\}$. Similarly, on the same event,

$$||(\mathcal{P}_\Gamma - \mathcal{P}_\Gamma \mathcal{P}_T \mathcal{P}_\Gamma)^{-1}||_F \leq \frac{1}{1-\beta^2}. \tag{6.25}$$

Therefore,

$$||W(i,j)||_F^2 \leq \frac{2\beta^2}{(1-\beta^2)^2} \frac{\mu r}{n}. \tag{6.26}$$

Hence, unconditionally, we have

$$\mathbb{P}\left(\sup_{i,j} |Q_{C_{ij}}| > \tau\lambda\right) \leq 2n^2 exp\left(\frac{-n(1-\beta^2)^2\tau^2}{2\beta^2\mu r}\right) + \mathbb{P}(||\mathcal{P}_\Gamma \mathcal{P}_T|| \geq \beta). \tag{6.27}$$

□

Similar proof to this part can be found in [52]. This concludes the proof of the Theorems. We performed series of numerical experiments to corroborate our claim. The report of these experiments is presented in the next chapter.

Chapter 7

Numerical Experiments

In this chapter, we report the results of the numerical experiments performed to test the efficiency and effectiveness of our convex program in finding planted quasi-clique in a graph. The experiments were performed on a HP computer with 16GB Ram and Intel core i7 processor. The machine runs on Debian Linux. The simulations were performed using CVXPY [63] with NCVX [64]. CVXPY is a python package used to solve convex optimization problems with different solvers, e.g SCS, CVXOPT, and XPRESS. The Mixed Integer programs were solved using XPRESS MP while our convex program was solved using SCS solver. For all the experiments, we chose $\lambda = \frac{1}{\sqrt{n}}$ following the recommendation in [52]. Various other values of λ will also work (see, e.g, [57, 58]). Every instance of the experiment was carried out ten times and the average was taken.

7.1 Performance Comparison with the Existing Mixed Integer Programming Formulations

We compare the performance of our nuclear norm minimization (NNM) formulation (5.23) with the existing mixed integer programming models (5.16), (5.17) and (5.19).

Two types of experiment were performed in this case. In the first case, we checked whether the recovered quasi-clique satisfies the edge density requirement or not. The second experiment focuses on the size of the recovered quasi-clique. The detailed report of both experiments is as follows.

The goal of the first experiment we performed was to examine the error in the edge density of the recovered γ-clique with respect to the edge density of the planted maximum γ-clique. We computed the relative error between the edge density of the recovered γ-clique and the edge density of the planted γ-clique (i.e, the expected edge density) for various γ. That is,

$$\text{Relative Error} = \frac{||\text{recovered } \gamma\text{-clique} - \text{planted } \gamma\text{-clique}||_F}{||\text{planted } \gamma\text{-clique}||_F}. \tag{7.1}$$

All the errors computed in this chapter are relative errors.

We considered graphs with 50 and 100 nodes for this case with planted γ-cliques of sizes 40 and 80, respectively. The planted γ-clique corresponds to a dense submatrix of the 50×50 (100×100) input matrix with 40 (80) non-zero rows/columns. We varied the edge density of the planted γ-clique by setting $\gamma = 0.6, 0.65, 0.7, \ldots, 1$. The probability, γ, determines whether an edge will exist between two nodes in the planted quasi-clique. The smaller the γ, the fewer the edges and consequently, the more difficult it is to recover what is planted. The setup follows the Stochastic Block Model (SBM) [100]. Detail is as follows. For the case $n = 50$, we generate a 50×50 symmetric matrix, M, with zero entries. We choose a 40×40 submatrix of this matrix and assign 1 to its indices with probability 0.6 (suppose $\gamma = 0.6$), using Bernoulli trial. This forms the dense component of the input matrix (the planted γ-clique). The entries of the remaining 10 rows and columns are also assigned values 1 but with a much smaller probability, (say $\rho = 0.2$). This forms the sparse component of the matrix (or the random noise). The goal is to recover the dense submatrix from the input matrix. The results of these experiments are reported in Tables 7.1 and 7.2. In both Tables, columns $2-4$ contain errors in edge density of the planted quasi-cliques recovered using the MIP models (5.16), (5.17) and (5.19) while column 5 contains the errors in edge density of

γ	**Relative Error in Edge Density of Recovered γ-Clique**			
	MIP(5.16)	**MIP**(5.17)	**MIP**(5.19)	**NNM**(5.23)
0.6	0.0922	0.1279	0.2093	0.3085
0.65	0.0688	0.1607	0.1897	0.2053
0.7	0.0809	0.1646	0.2086	0.1170
0.75	0.0809	0.1935	0.1558	0.0230
0.8	0.087	0.2044	0.1526	0
0.85	0.02	0.2265	0.1299	0
0.9	0.0215	0.2233	0.1806	0
0.95	0	0.2245	0.2026	0
1	0	0.2236	0.2434	0

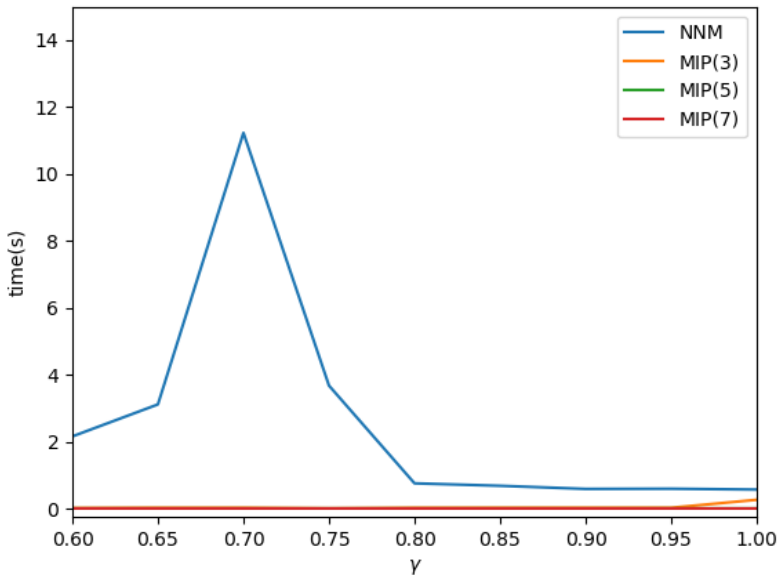
Table 7.1: Relative Errors in edge density of the planted maximum γ-clique compared with recovered γ-clique for a graph with 50 nodes. Result for each γ is average of 10 runs.

the γ-clique recovered using our nuclear norm minimization (NNM) approach. When $n = 50$, MIP(5.16) performed better than the two other MIP models for all values of γ. However, our nuclear norm minimization formulation has the best performance when $\gamma \geq 0.75$. For graphs with 100 nodes (see Table 7.2), MIP(5.19) performed better than other MIP models except for when γ is equal to $0.75, 0.95$ and 1 where MIP(5.16) has shown better performances. Nevertheless, when $\gamma \geq 0.7$, our model outperformed all the mixed integer programs. One can also infer from Tables 7.1 and 7.2 that as the graph size increases, the lower bound on γ for perfect recovery decreases. Figure 7.1 shows the CPU time for each of the methods for the experiments reported in Tables 7.1 and 7.2. Our off-the-shelf solver, SCS (splitting conic solver) [133], is faster than the popular SDP solvers like SeDumi [155] and SDP3 [159]. However, it is not as efficient as the well-developed FICO XPRESS optimizer used to solve the MIP models. Nonethless, as γ increases, there is a drastic drop in the CPU time of our NNM method in both instances.

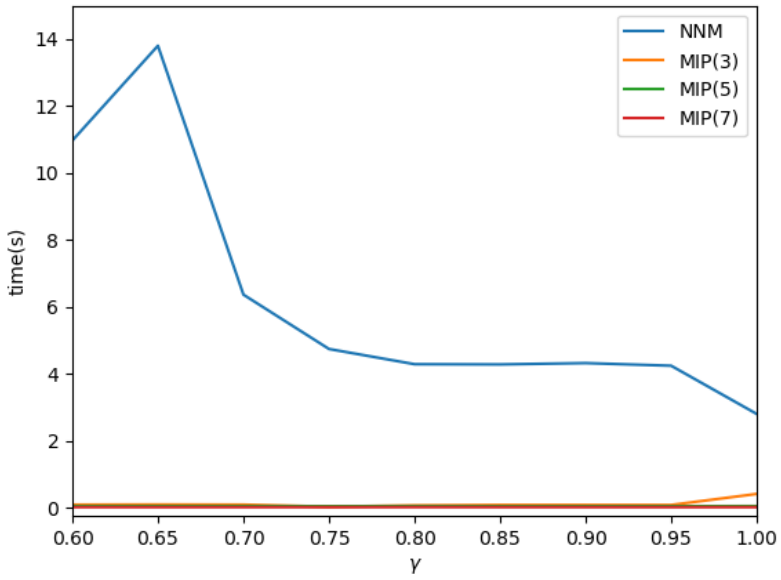
γ	**Relative Error in Edge Density of Recovered γ-Clique**			
	MIP(5.16)	**MIP**(5.17)	**MIP**(5.19)	**NNM**(5.23)
0.6	0.0916	0.0736	0.054	0.2424
0.65	0.0892	0.0843	0.0492	0.1012
0.7	0.0879	0.0719	0.0634	0.0131
0.75	0.0879	0.1378	0.0905	0
0.8	0.0829	0.1001	0.0766	0
0.85	0.0783	0.1563	0.0603	0
0.9	0.0817	0.1144	0.0975	0
0.95	0.0735	0.1432	0.1376	0
1	0.0694	0.1581	0.1717	0

Table 7.2: Relative Errors in edge density of the planted maximum γ-clique compared with recovered γ-clique for a graph with 100 nodes. Result for each γ is average of 10 runs

The second experiment was to find out if the number of nodes in the planted quasi-cliques, n_c, is the same as the number of nodes in the recovered quasi-cliques, η. For this experiment, we considered graphs of sizes $n = 50, 100, \ldots, 250$ and $\gamma = 0.6, 0.7, \ldots, 1$. We fixed the size of the submatrix of the adjacency matrix corresponding to the planted γ-clique to be 80% of the whole matrix, i,e, $n_c = 0.8 \times n$. We varied γ (the probability of an edge within γ-clique) but fixed $\rho = 0.2$ (the probability of diversionary edges or random noise). We again ran the experiment 10 times for each case and averaged the recovered quasi-clique size. The recovered quasi-clique size corresponds to the number of non-zero rows/columns in the recovered dense matrix. We compute the relative error in recovered γ clique size using:



(a) Average CPU time for planted quasi-clique recovery for a graph with 50 nodes



(b) Average CPU time for planted quasi-clique recovery for a graph with 100 nodes

Figure 7.1: Comparison of the CPU time for the MIP and NNM methods

$$\text{Relative Error in recovered } \gamma\text{-clique size } = \frac{|n_c - \eta|}{|n_c|}, \tag{7.2}$$

where n_c is size of the planted γ-clique and η is size of the recovered γ-cliqe. The results obtained are presented in Table 7.3. As shown in the last column of Table 7.3, the relative errors in the size of quasi-clique recovered via nuclear norm minimization are all zero since $n_c = \eta$ throughout. This shows that the convex formulation always returns correct planted quasi-clique size. MIP(5.16) has the overall worst performance in this experiment. Based on these two experiments, when $\gamma > 0.75$, $n_c = \eta$ and the error in edge density is equal to zero. This implies that our convex formulation perfectly recovers maximum planted quasi-clique when $\gamma > 0.75$ for $n \geq 50$ and n_c large enough.

7.2 Exact recovery from varying γ-clique size

As mentioned at the end of the last section, for our model to recover γ-clique, n_c must be large enough. To check what size of γ-clique can be recovered from a given network, we planted γ-clique of various sizes in some graphs and tried to recover them. We randomly generated $n \times n$ symmetric binary matrices with $n = 25, 50, 75, \ldots 250$. We inserted γ-clique of sizes equal to $10\%, 20\%, \ldots, 100\%$ of the size of the matrices using binomial distribution as described before. In each case, we fixed γ (the edge density parameter of the quasi-clique) to be 0.85 and ρ (the probability of adding a diversionary edge) was 0.25. We generated an $n \times n$ zero matrices $X \in \mathbf{R}^{n \times n}$ and $\hat{X} \in \mathbf{R}^{n \times n}$. Let Ω be the set of indices corresponding to the nodes belonging to the quasi-clique. Consequently, Ω^c is the set indices of the nodes of the diversionary edges. The entries of Ω are equal to one with probability γ and zero with probability $1 - \gamma$. Hence, Ω has expected cardinality $|\Omega| = \gamma n_c^2$. Ω^c also follows a Bernoulli distribution with parameter ρ and expected cardinality of $\rho(n^2 - n_c^2)$. n_c is the size of planted quasi-

$\gamma = 0.6$		Average Recovered Quasi-clique size/Average Relative Error							
n	n_c	**MIP**(5.16)		**MIP**(5.17)		**MIP**(5.19)		**NNM**(5.23)	
50	40	41	0.025	40.4	0.01	40.8	0.02	40	0
100	80	81.7	0.021	80.8	0.01	80.8	0.01	80	0
150	120	122.7	0.023	121	0.008	120.6	0.005	120	0
200	160	163.4	0.021	161.6	0.01	160.9	0.006	160	0
250	200	204.4	0.022	201.9	0.01	200.7	0.003	200	0
$\gamma = 0.7$									
50	40	40.5	0.013	40.1	0.003	40.4	0.01	40	0
100	80	81.3	0.016	80.4	0.005	80.5	0.006	80	0
150	120	122.4	0.02	121	0.008	120.7	0.006	120	0
200	160	163	0.019	161	0.006	160.6	0.004	160	0
250	200	204	0.02	201.5	0.008	200.4	0.002	200	0
$\gamma = 0.8$									
50	40	40.2	0.005	40	0	40.5	0.013	40	0
100	80	81	0.025	80.2	0.003	80.2	0.003	80	0
150	120	123	0.025	120.5	0.004	120.3	0.002	120	0
200	160	163.2	0.02	161.2	0.007	160.4	0.003	160	0
250	200	204	0.02	201.3	0.007	200.4	0.002	200	0
$\gamma = 0.9$									
50	40	40.6	0.015	40	0	40.5	0.013	40	0
100	80	81.1	0.014	80.2	0.003	80.2	0.003	80	0
150	120	122	0.017	120.3	0.002	120.1	0.001	120	0
200	160	163	0.019	160.9	0.006	160.3	0.002	160	0
250	200	203.9	0.02	200.9	0.005	200.1	0	200	0
$\gamma = 1$									
50	40	40	0	40	0	40	0	40	0
100	80	81	0.013	80	0	80	0	80	0
150	120	122	0.017	120.3	0.002	120	0	120	0
200	160	163	0.019	160.9	0.006	160	0	160	0
250	200	204	0.02	201	0.005	200	0	200	0

Table 7.3: Errors in the size of planted maximum γ-clique recovered using different methods for γ ranging from 0.6 to 1. n is the graph size while n_c is the size of the planted γ-clique. The first column under each method contains the average size of recovered quasi-clique using the method while the second contains the relative error of the method.

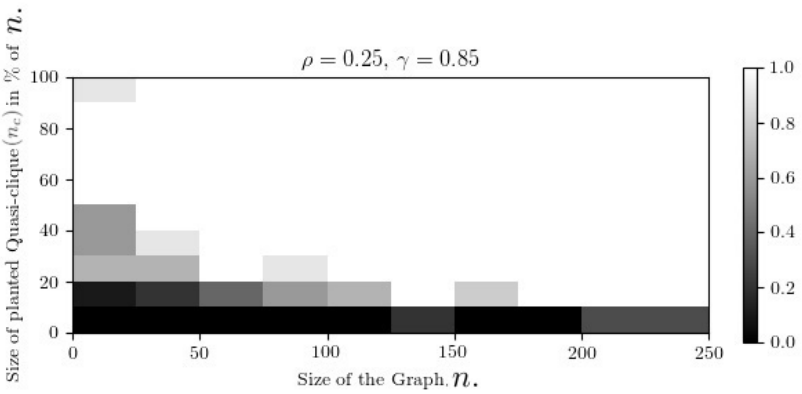


Figure 7.2: Exact recovery of varying quasi-clique size from graphs of different sizes

clique. We formed A by adding X and $\hat{X}$ and tried to recover X from A. We declare a recovery attempt to be successful if the relative error (in Frobenius norm) of the recovered matrix is less than or equal to 10^{-6}, i.e, $||X - X^*||_F/||X||_F \leq 10^{-6}$, where X^* is the recovered γ-clique. Figure 7.2 contains the plot of the probability of recovery from different graph sizes and quasi-clique sizes. White area denotes perfect recovery while the black area means the recovery attempt failed for every trial. We discovered that when the γ-clique is less than 10% of the graph size, recovery is impossible and when the quasi-clique size is bigger than 60% of the graph size, perfect recovery is guaranteed for all the graph sizes in our experiment.

7.3 Recovery from varying edge density and random noise

Theorem 5.8.1 shows that convex programming can recover an incoherent, low-rank noisy matrix with a reasonable level of missing entries. In the next experiment, we determined, experimentally, the maximum tolerable level of diversionary edges (noise) that can be added along with the minimum observing probability (edge density) required for perfect recovery. The set-up follows the previous experiment with $n = 100$ and 200. In both cases, the γ-clique size, n_c, was 85 and 170 respectively. The results of this experiment is presented in Figures 7.3 and 7.4. In both cases, we observed a phase transition, whereby the probability of recovery jumps from zero to one as γ reached a particular threshold. This is in agreement with our theoretical claim that when $\gamma \geq \frac{c_0 \mu r \log n}{n}$, for some constant $c_0 > 0$, exact recovery is guaranteed. In addition, we recorded complete failure when the probability of adding a random noise, ρ, is greater than or equal to 0.6, i.e, when $\rho \geq 0.6$. This implies that when the random noise becomes too much, recovery becomes impossible. As a conclusion, no γ-clique of any size can be recovered by our method when γ is less than the recovery threshold (i.e, the edge density of the planted γ-clique to be recovered is too low), the size of the planted γ-clique is too small or $\rho \geq 0.6$ meaning the random noise is too much.

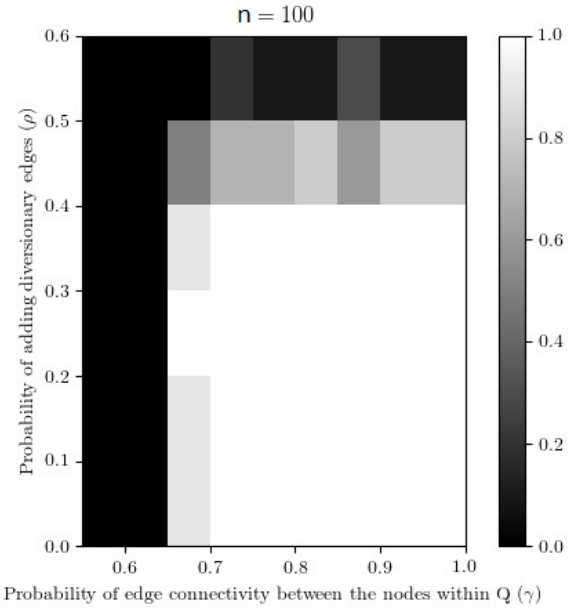


Figure 7.3: γ - clique recovery from varying edge density and random noise with $n = 100$ and $n_c = 85$

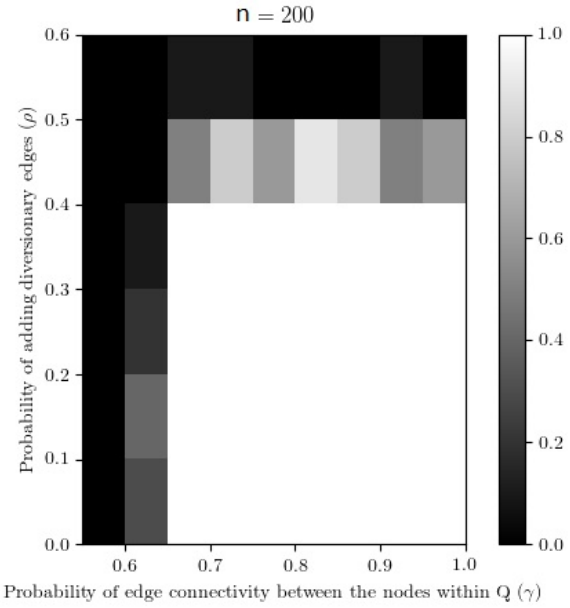


Figure 7.4: γ-clique recovery from varying edge density and random noise with $n = 200$ and $n_c = 170$

Chapter 8

Conclusion

In this thesis, we have considered the mathematical modelling, theoretical framework, and computational aspects of recovering a density-based clique relaxation, known as quasi-clique, in networks. The planted quasi-clique problem is an extension of the planted clique problem. We have shown that the planted quasi-clique can be solved by relaxing it to a convex programming. This was achieved by borrowing techniques from low rank matrix decomposition and adapting it to the problem under consideration. We have used this formulation to solve the planted maximum quasi-clique problem in the randomized case. In the case where the input graph contains the desired single large dense subgraph and a moderate number of diversionary vertices and edges, then the relaxation is exact. Therefore, in planted case, these difficult combinatorial optimization problem can be efficiently solved using the tractable relaxation. Among the methodological contributions of this thesis is sharp theoretical bounds obtained for the dual matrix. Our proof follows the state of the art in the matrix decomposition literature. However, our innovations lie in the tools we used for our analysis to get better results. We improved the results on low rank matrix decomposition by deriving the bound on our dual matrix, using the matrix $l_{\infty,2}$ norm. This norm has previously been used for matrix completion problem. We established conditions under which recovery is achievable by deriving a dual matrix, certifying the optimality of our solution. We present a simplified proof to show that quasi-cliques also posses what is known as quasi-hereditary property. This property can be exploited to develop enumerative algorithm for the problem.

Numerically, we have shown the superiority of our formulation over the existing MIP formulations. The results in Section 7.1 show that when γ is greater than a partic-

ular threshold, the planted quasi-clique formulation performs better than all the existing MIP formulations. This model can be very useful in applications where high accuracy is desired instead of speed. In addition, the results in Section 7.3 is consistent with our theoretical findings that when $\gamma \geq \frac{c_0 \mu r \log n}{n}$, our convex formulation is guaranteed to recover the planted quasi-clique. However, this is constrained by the level of random noise present. No matter what the value of γ is, when $\rho \geq 0.6$ for the cases we considered, recovery is impossible. Furthermore, it can be deduced from Section 7.2 that as the graph size increases, the proportion in size of the the planted quasi-clique required for perfect recovery reduces.

As an extension, our proposed convex program can be applied to other rank minimization problems with little modification. In addition, it will be interesting if this technique can be extended to recover disjoint quasi-cliques. One can then, possibly, apply the method to graph partitioning problems like clustering. Also, there are some special algorithms developed for nuclear norm minimization and low-rank plus sparse matrix recovery like the iterative singular value thresholding [45], accelerated proximal gradient [158] and the alternating direction method [78, 175]. It will be interesting to implement these algorithms for planted quasi-clique recovery to compare their performances with the SCS used for this work. Another aspect will be to use truncated nuclear norm instead of nuclear norm. These will form part of our future study.

List of References

[1] Sakirudeen A. Abdulsalaam and M. Montaz Ali. Convex formulation for planted quasi-clique recovery. *Manuscript submitted for publication*, 2019.

[2] Sakirudeen A. Abdulsalaam and M. Montaz Ali. Recovery guarantee for planted maximum quasi-clique recovery via nuclear norm minimization. *Manuscript in-preparation*, 2020.

[3] James Abello, PM Pardalos, and MGC Resende. On maximum clique problems in very large graphs. *DIMACS series*, 50:119–130, 1999.

[4] James Abello, Mauricio GC Resende, and Sandra Sudarsky. Massive quasi-clique detection. In *Latin American Symposium on Theoretical Informatics*, pages 598–612. Springer, 2002.

[5] Tero Aittokallio and Benno Schwikowski. Graph-based methods for analysing networks in cell biology. *Briefings in bioinformatics*, 7(3):243–255, 2006.

[6] Richard D Alba. A graph-theoretic definition of a sociometric clique. *Journal of Mathematical Sociology*, 3(1):113–126, 1973.

[7] Maria Teresa Almeida and Raúl Brás. The maximum l-triangle k-club problem: Complexity, properties, and algorithms. *Computers & Operations Research*, 111:258–270, 2019.

[8] Maria Teresa Almeida and Filipa D Carvalho. Integer models and upper bounds for the 3-club problem. *Networks*, 60(3):155–166, 2012.

[9] Maria Teresa Almeida and Filipa D Carvalho. An analytical comparison of the lp relaxations of integer models for the k-club problem. *European Journal of Operational Research*, 232(3):489–498, 2014.

[10] Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. *Random Structures and Algorithms*, 13(3-4):457–466, 1998.

[11] Brendan Ames. Convex relaxation for the planted clique, biclique, and clustering problems. 2011.

[12] Brendan PW Ames and Stephen A Vavasis. Nuclear norm minimization for the planted clique and biclique problems. *Mathematical programming*, 129(1):69–89, 2011.

[13] AT Amin and SL Hakimi. Upper bounds on the order of a clique of a graph. *SIAM Journal on Applied Mathematics*, 22(4):569–573, 1972.

[14] Sanjeev Arora, David Karger, and Marek Karpinski. Polynomial time approximation schemes for dense instances of np-hard problems. *Journal of computer and system sciences*, 58(1):193–210, 1999.

[15] Yuichi Asahiro, Kazuo Iwama, Hisao Tamaki, and Takeshi Tokuyama. Greedily finding a dense subgraph. *Journal of Algorithms*, 34(2):203–221, 2000.

[16] Oskar Maria Baksalary and Paulina Kik. On commutativity of projectors. *Linear algebra and its applications*, 417(1):31–41, 2006.

[17] Balasundaram Balabhaskar. *Graph theoretic generalization of clique: Optimization and extensions*. PhD thesis, PhD thesis, Texas A & M University, 2007.

[18] Balabhaskar Balasundaram and Foad Mahdavi Pajouh. Graph theoretic clique relaxations and applications. *Handbook of combinatorial optimization*, pages 1559–1598, 2013.

[19] Balabhaskar Balasundaram, Sergiy Butenko, and Svyatoslav Trukhanov. Novel approaches for analyzing biological networks. *Journal of Combinatorial Optimization*, 10(1):23–39, 2005.

[20] Balabhaskar Balasundaram, Sergiy Butenko, and Illya V Hicks. Clique relaxations in social network analysis: The maximum k-plex problem. *Operations Research*, 59(1):133–142, 2011.

[21] Paul Balister, Béla Bollobás, Julian Sahasrabudhe, and Alexander Veremyev. Dense subgraphs in random graphs. *Discrete Applied Mathematics*, 260:66–74, 2019.

[22] Albert-László Barabási. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987): 20120375, 2013.

[23] Boaz Barak, Samuel Hopkins, Jonathan Kelner, Pravesh K Kothari, Ankur Moitra, and Aaron Potechin. A nearly tight sum-of-squares lower bound for the planted clique problem. *SIAM Journal on Computing*, 48(2):687–735, 2019.

[24] Roberto Battiti and Franco Mascia. Reactive local search for maximum clique: A new implementation. Technical report, University of Trento, 2007.

[25] Roberto Battiti and Marco Protasi. Reactive local search for the maximum clique problem 1. *Algorithmica*, 29(4):610–637, 2001.

[26] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

[27] Carolyn Beck and Raffaello D'Andrea. Computational study and comparisons of lft reducibility methods. In *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207)*, volume 2, pages 1013–1017. IEEE, 1998.

[28] Devora Berlowitz, Sara Cohen, and Benny Kimelfeld. Efficient enumeration of maximal k-plexes. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 431–444. ACM, 2015.

[29] Dimitri P Bertsekas. *Convex optimization theory*. Athena Scientific Belmont, 2009.

[30] Dimitri P Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Academic press, 2014.

[31] Malay Bhattacharyya and Sanghamitra Bandyopadhyay. Mining the largest quasi-clique in human protein interactome. In *2009 International Conference on Adaptive and Intelligent Systems*, pages 194–199. IEEE, 2009.

[32] Alain Billionnet. Different formulations for solving the heaviest k-subgraph problem. *INFOR: Information Systems and Operational Research*, 43(3):171–186, 2005.

[33] Alain Billionnet and Frédéric Roupin. A deterministic approximation algorithm for the densest k-subgraph problem. *International Journal of Operational Research*, 3(3):301–314, 2008.

[34] Thomas Blumensath and Mike E Davies. Iterative hard thresholding for compressed sensing. *Applied and computational harmonic analysis*, 27(3):265–274, 2009.

[35] Béla Bollobás. Random graphs. In *Modern graph theory*, pages 215–252. Springer, 1998.

[36] Immanuel M Bomze, Marco Budinich, Panos M Pardalos, and Marcello Pelillo. The maximum clique problem. In *Handbook of combinatorial optimization*, pages 1–74. Springer, 1999.

[37] John Adrian Bondy, Uppaluri Siva Ramachandra Murty, et al. *Graph theory with applications*, volume 290. Macmillan London, 1976.

[38] Jonathan Borwein and Adrian S Lewis. *Convex analysis and nonlinear optimization: theory and examples*. Springer Science & Business Media, 2010.

[39] Jean-Marie Bourjolly, Gilbert Laporte, and Gilles Pesant. Heuristics for finding k-clubs in an undirected graph. *Computers & Operations Research*, 27(6):559–569, 2000.

[40] Jean-Marie Bourjolly, Gilbert Laporte, and Gilles Pesant. An exact algorithm for the maximum k-club problem in an undirected graph. *European Journal of Operational Research*, 138(1):21–28, 2002.

[41] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[42] Mauro Brunato, Holger H Hoos, and Roberto Battiti. On effectively finding maximal quasi-cliques in graphs. In *International Conference on Learning and Intelligent Optimization*, pages 41–55. Springer, 2007.

[43] Austin Buchanan and Hosseinali Salemi. Parsimonious formulations for low-diameter clusters. *Optimization Online Eprints*, 3:14–21, 2017.

[44] Marco Budinich. Exact bounds on the order of the maximum clique of a graph. *Discrete Applied Mathematics*, 127(3):535–543, 2003.

[45] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on optimization*, 20(4): 1956–1982, 2010.

[46] EJ Candes and T Tao. Decoding by linear programming. *IEEE Transactions on Information Theory*, 51(12):4203–4215, 2005.

[47] Emmanuel Candes, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *arXiv preprint math/0409186*, 2004.

[48] Emmanuel J Candes and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[49] Emmanuel J Candés and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

[50] Emmanuel J Candès and Terence Tao. The power of convex relaxation: Near-optimal matrix completion. *IEEE Transactions on Information Theory*, 56(5): 2053–2080, 2010.

[51] Emmanuel J Candes, Justin K Romberg, and Terence Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(8):1207–1223, 2006.

[52] Emmanuel J Candès, Xiaodong Li, Yi Ma, and John Wright. Robust principal component analysis? *Journal of the ACM (JACM)*, 58(3):11:1 – 11:37, 2011.

[53] Joshua Cape, Minh Tang, and Carey E Priebe. The two-to-infinity norm and singular subspace geometry with applications to high-dimensional statistics. *arXiv preprint arXiv:1705.10735*, 2017.

[54] Venkat Chandrasekaran, Sujay Sanghavi, Pablo A Parrilo, and Alan S Willsky. Rank-sparsity incoherence for matrix decomposition. *SIAM Journal on Optimization*, 21(2):572–596, 2011.

[55] Maw-Shang Chang, Ling-Ju Hung, Chih-Ren Lin, and Ping-Chen Su. Finding large k-clubs in undirected graphs. *Computing*, 95(9):739–758, 2013.

[56] Sheng Chen, Stephen A Billings, and Wan Luo. Orthogonal least squares methods and their application to non-linear system identification. *International Journal of control*, 50(5):1873–1896, 1989.

[57] Yudong Chen. Incoherence-optimal matrix completion. *IEEE Trans. Information Theory*, 61(5):2909–2923, 2015.

[58] Yudong Chen, Ali Jalali, Sujay Sanghavi, and Constantine Caramanis. Low-rank matrix recovery from errors and erasures. *IEEE Transactions on Information Theory*, 59(7):4324–4337, 2013.

[59] Alessio Conte, Donatella Firmani, Caterina Mordente, Maurizio Patrignani, and Riccardo Torlone. Fast enumeration of large k-plexes. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 115–124. ACM, 2017.

[60] Derek G Corneil and Yehoshua Perl. Clustering and domination in perfect graphs. *Discrete Applied Mathematics*, 9(1):27–39, 1984.

[61] Geoffrey M Davis, Stephane G Mallat, and Zhifeng Zhang. Adaptive time-frequency decompositions. *Optical engineering*, 33(7):2183–2192, 1994.

[62] D Deedell and JA Tropp CoSaMP. "iterative signal recovery from incomplete and inaccurate samples. In *Appl*, 2008.

[63] Steven Diamond and Stephen Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17 (83):1–5, 2016.

[64] Steven Diamond, Reza Takapoui, and S Boyd. A general system for heuristic minimization of convex functions over non-convex sets. *Optimization Methods and Software*, 33(1):165–193, 2018.

[65] David L Donoho and Michael Elad. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of the National Academy of Sciences*, 100(5):2197–2202, 2003.

[66] David L Donoho et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS math challenges lecture*, 1(2000):32, 2000.

[67] David L Donoho et al. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[68] Dingzhu Du and Panos M Pardalos. *Handbook of combinatorial optimization*, volume 4. Springer Science & Business Media, 1998.

[69] Laurent El Ghaoui, Francois Oustry, and Mustapha AitRami. A cone complementarity linearization algorithm for static output-feedback and related problems. *IEEE transactions on automatic control*, 42(8):1171–1176, 1997.

[70] Bassem Fares, Pierre Apkarian, and Dominikus Noll. An augmented lagrangian method for a class of lmi-constrained problems in robust control theory. *International Journal of Control*, 74(4):348–360, 2001.

[71] Stanley Wasserman; Katherine Faust. *Social network analysis : methods and applications*, volume 8. Cambridge University Press, 1994.

[72] Maryam Fazel. *Matrix rank minimization with applications*. PhD thesis, PhD thesis, Stanford University, 2002.

[73] Maryam Fazel, Haitham Hindi, Stephen P Boyd, et al. A rank minimization heuristic with application to minimum order system approximation. In *Proceedings of the American control conference*, volume 6, pages 4734–4739. Citeseer, 2001.

[74] Uriel Feige and Robert Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Random Structures & Algorithms*, 16(2):195–208, 2000.

[75] Massimo Fornasier. Numerical methods for sparse recovery. *Theoretical foundations and numerical methods for sparse recovery*, 14:93–200, 2010.

[76] Jerome H Friedman and Werner Stuetzle. Projection pursuit regression. *Journal of the American statistical Association*, 76(376):817–823, 1981.

[77] Alan Frieze and Ravi Kannan. A new approach to the planted clique problem. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science*, volume 2 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 187–198, 2008. ISBN 978-3-939897-08-8.

[78] Arvind Ganesh, Zhouchen Lin, John Wright, Leqin Wu, Minming Chen, and Yi Ma. Fast algorithms for recovering a corrupted low-rank matrix. In *2009 3rd IEEE International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 213–216. IEEE, 2009.

[79] Michael R Garey and David S Johnson. *Computers and intractability*, volume 29. wh freeman New York, 2002.

[80] Bernd Gärtner and Jiri Matousek. *Approximation algorithms and semidefinite programming*. Springer Science & Business Media, 2012.

[81] Fred Glover and Manuel Laguna. Tabu search. In *Handbook of combinatorial optimization*, pages 2093–2229. Springer, 1998.

[82] Michel X Goemans and David P Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6):1115–1145, 1995.

[83] Gene H Golub and Charles F Van Loan. *Matrix computations*, volume 3. JHU press, 2012.

[84] Karolos M Grigoriadis and Eric B Beran. Alternating projection algorithms for linear matrix inequalities problems with rank constraints. In *Advances in linear matrix inequality methods in control*, pages 251–267. SIAM, 2000.

[85] David Gross. Recovering low-rank matrices from few coefficients in any basis. *IEEE Transactions on Information Theory*, 57(3):1548–1566, 2011.

[86] Martin Grötschel, László Lovász, and Alexander Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica*, 1(2):169–197, 1981.

[87] Francesco Gullo. From patterns in data to knowledge discovery: what data mining can do. *Physics Procedia*, 62:18–22, 2015.

[88] Bruce Hajek, Yihong Wu, and Jiaming Xu. Computational lower bounds for community detection on random graphs. In *Conference on Learning Theory*, pages 899–928, 2015.

[89] Elad Hazan and Robert Krauthgamer. How hard is it to approximate the best nash equilibrium? *SIAM Journal on Computing*, 40(1):79–91, 2011.

[90] Christoph Helmberg. Semidefinite programming for combinatorial optimization. 2000.

[91] Daxin Jiang and Jian Pei. Mining frequent cross-graph quasi-cliques. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 2(4):1–42, 2009.

[92] Ari Juels and Marcus Peinado. Hiding cliques for cryptographic security. *Designs, Codes and Cryptography*, 20(3):269–280, 2000.

[93] Björn H Junker and Falk Schreiber. *Analysis of biological networks*, volume 2. John Wiley & Sons, 2011.

[94] J Mark Keil and Timothy B Brecht. The complexity of clustering in planar graphs. *J. Combinatorial Mathematics and Combinatorial Computing*, 9:155–159, 1991.

[95] Raghunandan H Keshavan, Andrea Montanari, and Sewoong Oh. Matrix completion from a few entries. *IEEE transactions on information theory*, 56(6): 2980–2998, 2010.

[96] Rasoul Kiani, Siamak Mahdavi, and Amin Keshavarzi. Analysis and prediction of crimes by clustering and classification. *Analysis*, 4(8), 2015.

[97] Christian Komusiewicz, André Nichterlein, Rolf Niedermeier, and Marten Picker. Exact algorithms for finding well-connected 2-clubs in sparse real-world graphs: Theory and experiments. *European Journal of Operational Research*, 275(3):846–864, 2019.

[98] G KORTSARZ. On choosing a dense subgraph. In *Proceedings of the 34th Annual IEEE Symposium on Foundation of Computer Science, 1993*, 1993.

[99] Luděk Kučera. Expected complexity of graph partitioning problems. *Discrete Applied Mathematics*, 57(2):193–212, 1995.

[100] Clement Lee and Darren J Wilkinson. A review of stochastic block models and extensions for graph clustering. *Applied Network Science*, 4(1):122, 2019.

[101] Adrian S Lewis. The mathematics of eigenvalue optimization. *Mathematical Programming*, 97(1-2):155–176, 2003.

[102] Xiaodong Li. Compressed sensing and matrix completion with constant proportion of corruptions. *Constructive Approximation*, 37(1):73–99, 2013.

[103] Zhouchen Lin, Minming Chen, and Yi Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010.

[104] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, 1995.

[105] Guimei Liu and Limsoon Wong. Effective pruning techniques for mining quasi-cliques. *Machine Learning and Knowledge Discovery in Databases*, pages 33–49, 2008.

[106] Yong-Jin Liu, Defeng Sun, and Kim-Chuan Toh. An implementable proximal point algorithmic framework for nuclear norm minimization. *Mathematical programming*, 133(1-2):399–436, 2012.

[107] Zhang Liu and Lieven Vandenberghe. Interior-point method for nuclear norm approximation with application to system identification. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1235–1256, 2009.

[108] Robert D Luce. Connectivity and generalized cliques in sociometric group structure. *Psychometrika*, 15(2):169–190, 1950.

[109] Robert D Luce and Albert D Perry. A method of matrix analysis of group structure. *Psychometrika*, 14(2):95–116, 1949.

[110] Vince Lyzinski, Daniel Sussman, Minh Tang, Avanti Athreya, and Carey Priebe. Perfect clustering for stochastic blockmodel graphs via adjacency spectral embedding. *arXiv preprint arXiv:1310.0532*, 2013.

[111] Shiqian Ma, Donald Goldfarb, and Lifeng Chen. Fixed point and bregman iterative methods for matrix rank minimization. *Mathematical Programming*, 128 (1-2):321–353, 2011.

[112] Stéphane G Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on signal processing*, 41(12):3397–3415, 1993.

[113] Fabrizio Marinelli, Andrea Pizzuti, and Fabrizio Rossi. Lp-based dual bounds for the maximum quasi-clique problem. *Discrete Applied Mathematics*, 2020.

[114] Hideo Matsuda, Tatsuya Ishihara, and Akihiro Hashimoto. Classifying molecular sequences using a linkage graph with their pairwise similarities. *Theoretical Computer Science*, 210(2):305–325, 1999.

[115] Benjamin McClosky and Illya V Hicks. Combinatorial algorithms for the maximum k-plex problem. *Journal of combinatorial optimization*, 23(1):29–49, 2012.

[116] Ciaran McCreesh and Patrick Prosser. Finding maximum k-cliques faster using lazy global domination. In *Ninth Annual Symposium on Combinatorial Search*, 2016.

[117] Mehran Mesbahi and George P Papavassilopoulos. On the rank minimization problem over a positive semidefinite linear matrix inequality. *IEEE Transactions on Automatic Control*, 42(2):239–243, 1997.

[118] Zhuqi Miao and Balabhaskar Balasundaram. An ellipsoidal bounding scheme for the quasi-clique number of a graph. *INFORMS Journal on Computing*, 2020.

[119] Robert J Mokken. Cliques, clubs and clans. *Quality and quantity*, 13(2):161–173, 1979.

[120] John W Moon and Leo Moser. On cliques in graphs. *Israel journal of Mathematics*, 3(1):23–28, 1965.

[121] Esmaeel Moradi and Balabhaskar Balasundaram. Finding a maximum k-club using the k-clique formulation and canonical hypercube cuts. *Optimization Letters*, 12(8):1947–1957, 2018.

[122] Hannes Moser, Rolf Niedermeier, and Manuel Sorge. Exact combinatorial algorithms and experiments for finding maximum k-plexes. *Journal of combinatorial optimization*, 24(3):347–373, 2012.

[123] Theodore S Motzkin and Ernst G Straus. Maxima for graphs and a new proof of a theorem of turán. *Canadian Journal of Mathematics*, 17:533–540, 1965.

[124] Raj Rao Nadakuditi. On hard limits of eigen-analysis based planted clique detection. In *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pages 129–132. IEEE, 2012.

[125] Yurii Nesterov and Arkadii Nemirovskii. *Interior-point polynomial algorithms in convex programming*, volume 13. Siam, 1994.

[126] Mark Newman. *Networks*. Oxford university press, 2018.

[127] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[128] Nam H Nguyen and Trac D Tran. Exact recoverability from dense corrupted observations via ℓ_1-minimization. *IEEE transactions on information theory*, 59 (4):2017–2035, 2013.

[129] Guillaume Obozinski, Ben Taskar, and Michael I Jordan. Joint covariate selection and joint subspace selection for multiple classification problems. *Statistics and Computing*, 20(2):231–252, 2010.

[130] Robert Orsi, Uwe Helmke, and John B Moore. A newton-like method for solving rank constrained linear matrix inequalities. *Automatica*, 42(11):1875–1882, 2006.

[131] Michael R Osborne, Brett Presnell, and Berwin A Turlach. A new approach to variable selection in least squares problems. *IMA journal of numerical analysis*, 20(3):389–403, 2000.

[132] Patric RJ Östergård. A fast algorithm for the maximum clique problem. *Discrete Applied Mathematics*, 120(1):197–207, 2002.

[133] Brendan O'Donoghue, Eric Chu, Neal Parikh, and Stephen Boyd. Conic optimization via operator splitting and homogeneous self-dual embedding. *Journal of Optimization Theory and Applications*, 169(3):1042–1068, 2016.

[134] F Mahdavi Pajouh and Balabhaskar Balasundaram. On inclusionwise maximal and maximum cardinality k-clubs in graphs. *Discrete Optimization*, 9(2):84–97, 2012.

[135] Panos M Pardalos and Gregory P Rodgers. A branch and bound algorithm for the maximum clique problem. *Computers & operations research*, 19(5):363–375, 1992.

[136] Panos M Pardalos and Jue Xue. The maximum clique problem. *Journal of global Optimization*, 4(3):301–328, 1994.

[137] Pablo A Parrilo and Sven Khatri. On cone-invariant linear matrix inequalities. *IEEE Transactions on Automatic Control*, 45(8):1558–1563, 2000.

[138] Grigory Pastukhov, Alexander Veremyev, Vladimir Boginski, and Oleg A Prokopyev. On maximum degree-based-quasi-clique problem: Complexity and exact approaches. *Networks*, 71(2):136–152, 2018.

[139] Jeffrey Pattillo, Nataly Youssef, and Sergiy Butenko. Clique relaxation models in social network analysis. In *Handbook of Optimization in Complex Networks*, pages 143–162. Springer, 2012.

[140] Jeffrey Pattillo, Alexander Veremyev, Sergiy Butenko, and Vladimir Boginski. On the maximum quasi-clique problem. *Discrete Applied Mathematics*, 161(1): 244–257, 2013.

[141] Jeffrey Pattillo, Nataly Youssef, and Sergiy Butenko. On clique relaxation models in network analysis. *European Journal of Operational Research*, 226(1): 9–18, 2013.

[142] Jian Pei, Daxin Jiang, and Aidong Zhang. On mining cross-graph quasi-cliques. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 228–238. ACM, 2005.

[143] Arnaud Quirin, Oscar Cordón, Benjamín Vargas-Quesada, and Félix de Moya-Anegón. Graph-based data mining: A new tool for the analysis and comparison of scientific domains represented as scientograms. *Journal of Informetrics*, 4(3): 291–312, 2010.

[144] Elizaveta Rebrova and Roman Vershynin. Norms of random matrices: local and global problems. *Advances in Mathematics*, 324:40–83, 2018.

[145] Benjamin Recht. A simpler approach to matrix completion. *Journal of Machine Learning Research*, 12(Dec):3413–3430, 2011.

[146] Benjamin Recht, Weiyu Xu, and Babak Hassibi. Necessary and sufficient conditions for success of the nuclear norm heuristic for rank minimization. In *Decision and Control, 2008. CDC 2008. 47th IEEE Conference on*, pages 3065–3070. IEEE, 2008.

[147] Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

[148] Jasson DM Rennie and Nathan Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, pages 713–719. ACM, 2005.

[149] Celso C Ribeiro and José A Riveaux. An exact algorithm for the maximum quasi-clique problem. *International Transactions in Operational Research*, 26 (6):2199–2229, 2019.

[150] R Tyrrell Rockafellar. *Convex analysis*, volume 28. Princeton university press, 1970.

[151] Stephen B Seidman and Brian L Foster. A graph-theoretic generalization of the clique concept*. *Journal of Mathematical sociology*, 6(1):139–154, 1978.

[152] Shahram Shahinpour. *Optimization-Based Network Analysis with Applications in Clustering and Data Mining*. PhD thesis, 2013.

[153] Oleg A Shirokikh. *Degree-based Clique Relaxations: Theoretical Bounds, Computational Issues, and Applications*. PhD thesis, University of Florida, 2013.

[154] Robert E Skelton, Tetsuya Iwasaki, and Dimitri E Grigoriadis. *A unified algebraic approach to control design*. CRC Press, 1997.

[155] Jos F Sturm. Using sedumi 1.02, a matlab toolbox for optimization over symmetric cones. *Optimization methods and software*, 11(1-4):625–653, 1999.

[156] Terence Tao. Topics in random matrix theory.

[157] Loren Terveen, Will Hill, and Brian Amento. Constructing, organizing, and visualizing collections of topically related web resources. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 6(1):67–94, 1999.

[158] Kim-Chuan Toh and Sangwoon Yun. An accelerated proximal gradient algorithm for nuclear norm regularized linear least squares problems. *Pacific Journal of optimization*, 6(615-640):15, 2010.

[159] Kim-Chuan Toh, Michael J Todd, and Reha H Tütüncü. Sdpt3—a matlab software package for semidefinite programming, version 1.3. *Optimization methods and software*, 11(1-4):545–581, 1999.

[160] Lloyd N Trefethen and David Bau III. *Numerical linear algebra*, volume 50. Siam, 1997.

[161] Paul Tseng. On accelerated proximal gradient methods for convex-concave optimization. *submitted to SIAM Journal on Optimization*, 2:3, 2008.

[162] Jean-Baptiste Hiriart Urruty and Claude Lemaréchal. *Convex analysis and minimization algorithms*. Springer-Verlag, 1996.

[163] Alexander Veremyev and Vladimir Boginski. Identifying large robust network clusters via new compact formulations of maximum k-club problems. *European Journal of Operational Research*, 218(2):316–326, 2012.

[164] Alexander Veremyev, Oleg A Prokopyev, and Eduardo L Pasiliao. Critical nodes for distance-based connectivity and related problems in graphs. *Networks*, 66(3): 170–195, 2015.

[165] Alexander Veremyev, Oleg A Prokopyev, Sergiy Butenko, and Eduardo L Pasiliao. Exact mip-based approaches for finding maximum quasi-cliques and dense subgraphs. *Computational Optimization and Applications*, 64(1):177–214, 2016.

[166] Anurag Verma and Sergiy Butenko. Network clustering via clique relaxations: A community based. *Graph Partitioning and Graph Clustering*, 588:129, 2013.

[167] Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

[168] G Alistair Watson. Characterization of the subdifferential of some matrix norms. *Linear algebra and its applications*, 170:33–45, 1992.

[169] Kilian Q Weinberger and Lawrence K Saul. Unsupervised learning of image manifolds by semidefinite programming. *International journal of computer vision*, 70(1):77–90, 2006.

[170] Douglas Brent West et al. *Introduction to graph theory*, volume 2. Prentice hall Upper Saddle River, 2001.

[171] John Wright, Arvind Ganesh, Shankar Rao, Yigang Peng, and Yi Ma. Robust principal component analysis: Exact recovery of corrupted low-rank matrices via convex optimization. In *Advances in neural information processing systems*, pages 2080–2088, 2009.

[172] Jennifer Xu and Hsinchun Chen. Criminal network analysis and visualization. *Communications of the ACM*, 48(6):100–107, 2005.

[173] Yong-Quan Yin, Zhi-Dong Bai, and Pathak R Krishnaiah. On the limit of the largest eigenvalue of the large dimensional sample covariance matrix. *Probability theory and related fields*, 78(4):509–521, 1988.

[174] Haiyuan Yu, Alberto Paccanaro, Valery Trifonov, and Mark Gerstein. Predicting interactions in protein networks by completing defective cliques. *Bioinformatics*, 22(7):823–829, 2006.

[175] Xiaoming Yuan and Junfeng Yang. Sparse and low-rank matrix decomposition via alternating direction methods. *preprint*, 12:2, 2009.

[176] Xin-Yuan Zhao, Defeng Sun, and Kim-Chuan Toh. A newton-cg augmented lagrangian method for semidefinite programming. *SIAM Journal on Optimization*, 20(4):1737–1765, 2010.

[177] Qing Zhou, Una Benlic, and Qinghua Wu. An opposition-based memetic algorithm for the maximum quasi-clique problem. *European Journal of Operational Research*, 2020.