Air Force Institute of Technology
AFIT Scholar

Theses and Dissertations Student Graduate Works

3-24-2016

Clustering Theory and Data Driven Health Care Strategies

Takayuki Iguchi

Recommended Citation
Iguchi, Takayuki, "Clustering Theory and Data Driven Health Care Strategies" (2016). *Theses and Dissertations*. 287.
https://scholar.afit.edu/etd/287

Clustering Theory and Data-Driven Health Care Strategies

THESIS

Takayuki Iguchi, Second Lieutenant, USAF

AFIT-ENC-MS-16-M-001

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENC-MS-16-M-001

CLUSTERING THEORY AND DATA-DRIVEN HEALTH CARE STRATEGIES

THESIS

Presented to the Faculty
Department of Mathematics and Statistics
Graduate School of Engineering and Management
Air Force Institute of Technology
Air University
Air Education and Training Command
in Partial Fulfillment of the Requirements for the
Degree of Master of Applied Mathematics

Takayuki Iguchi, B.S. Mathematics
Second Lieutenant, USAF

March 2016

DISTRIBUTION STATEMENT A
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT-ENC-MS-16-M-001

CLUSTERING THEORY AND DATA-DRIVEN HEALTH CARE STRATEGIES

THESIS

Takayuki Iguchi, B.S. Mathematics
Second Lieutenant, USAF

Committee Membership:

Captain J. D. Peterson
Chair

Dr. M. C. Fickus
Member

Major D. G. Mixon
Member

AFIT-ENC-MS-16-M-001

Abstract

DoD health care requires reform with growing costs causing concerns of decreased military capability. Porter in [1] proposes a radical strategy to fix current health care delivery systems via organizing medical teams around patients with similar treatment requirements. This is a clustering problem; how do you partition the set of patients so that each group has similar treatment needs? We provide advances in clustering theory relevant to this new health care strategy. In particular, we create fast certifiably optimal k-means clustering using what is known as Probably Certifiably Correct (PCC) algorithms which achieves state-of-the-art performance under certain models. Inspired by the health-care clustering problem, we pay particular attention to a Bipartite Stochastic Block Model and produce an alternative PCC algorithm specific to this model. We conclude by demonstrating the potential utility of applying these clustering methods in health-care. Using conditional entropy as a metric, clusters obtained from our methods vastly outperform partitions prescribed by subject matter experts.

AFIT-ENC-MS-16-M-001

I dedicate this thesis to my family and friends, especially the Watts family for their support through a difficult time. Without the encouragement and instruction of Dr. John Gorman, Dr. Kurt Herzinger, and Dr. Michelle Ghrist, I would not have pursued a graduate degree in mathematics. I am deeply grateful to you all.

Acknowledgements

Creating this thesis would not have been possible without the tremendous support of many individuals. I would like to acknowledge the contributions of a few people who spent a considerable amount of time and energy assisting me in this venture.

I thank the 711th Human Performance Wing for allowing me access to their dataset and providing the motivation for this thesis. I thoroughly enjoyed the opportunity to work with Major Dustin Mixon, Captain Jesse Peterson and Soledad Villar in writing a paper which I have included in this thesis. I thank Dr. Matthew Fickus for his instruction and his assistance in creating this thesis and Major Mixon for his overwhelmingly generous support and willingness to entertain my questions upon a moment's notice. I am deeply grateful Captain Peterson chose to be my advisor, and I have placed much value in his patient mentorship.

Takayuki Iguchi

Table of Contents

List of Figures

List of Tables

CLUSTERING THEORY AND
DATA-DRIVEN HEALTH CARE STRATEGIES

I. Introduction

The military needs to fundamentally change the way it provides health care. According to the Congressional Budget Office (CBO), health care costs in the Department of Defense (DoD) are growing at an unsustainable rate: DoD health care spending grew “rapidly as a share of the defense budget [from 2004 to 2014], outpacing growth in the economy, growth in per capita health care spending in the United States, and growth in funding for DoD’s base budget”[2]. The CBO projects this trend to continue, with health care taking up a growing proportion of the DoD’s budget until 2028 [2]. This is cause for concern, since it leaves less money for other important defense programs, potentially hindering the overall capability of the U.S. military [3]. In response, the DoD has made it a priority to find ways to save money throughout the Military Health System [3].

Professor Michael Porter of the Harvard Business School offers a radical proposal to deliver higher-quality health care with less money [1, 4]. His main idea is to assign patients to different teams of medical professionals called **Integrated Practice Units** (IPUs) according to medical needs. Since patients in a common IPU will have similar needs, each IPU will be able to provide focused care. To date, the IPU framework has found success in several isolated instances outside the DoD [1, 5]. In response to this success, the Air Force Surgeon General has requested that the Air Force Medical Service implement Porter’s IPU framework [6]. Following the 2015 Air Force Future Operating Concept [7], we seek a data-driven approach to the Air

Force's implementation of Porter's strategy. Fortunately, in cooperation with the 711th Human Performance Wing, we have access to a large set of Air Force health care data, in particular, the anonymized medical histories of all Air Force personnel, retirees, and dependents over the course of a decade. The purpose of this thesis is to leverage this dataset to initiate a data-driven partition of patients for IPU assignments.

1.1 Partitioning Patients with Clustering Theory

Our task is to divide a population of patients according to similar health care needs. In addition, we want our patient partition to be time-robust, meaning we can expect patients assigned to a common IPU to have similar needs for an extended period of time (say, five to ten years). Porter in [4] offers a patient partition consisting of the following categories:

1. Healthy adults
2. Mothers and young children
3. Adults at risk of a chronic disease
4. Adults with a chronic disease
5. Adults with a rare condition
6. The disabled and the frail elderly.

Unfortunately, there is no analysis regarding the time-robustness of this categorization. In this thesis, we examine time-robustness by dividing our ten-year database of medical records: we use the first five years to create a partition, and then we use the second five years to evaluate the partition by comparing patients' medical usage. This makes intuitive sense since patients with similar records should continue to have related health care needs. We therefore seek a partition based on the first five years of medical records which minimizes some notion of dissimilarity within divisions. This is an instance of a fundamental machine learning task known as **clustering**.

Clustering is the general problem of partitioning a given finite set P into k subsets $A_1, \ldots, A_k$ such that some dissimilarity function is minimized. Usually, this function is chosen for a specific purpose with an application in mind. In the context of the Air Force health care problem, we desire to group patients in such a way that each cluster has the most predictable health care needs. In particular, suppose a patient in cluster t receives a diagnosis w with probability $\Pr(W = w|T = t)$. A common way to quantify unpredictability of this random variable is called entropy $H(W|T = t)$ defined by

$$H(W|\mathbf{T} = t) = -\sum_{w \in \mathscr{W}} \Pr(w|t) \log_2 \big(\Pr(w|t) \big).$$

Since we seek to minimize uncertainty over all clusters simultaneously, we consider the natural weighted average of these entropies known as conditional entropy:

$$H(W|\mathbf{T}) = -\sum_{t \in \mathscr{T}} \Pr(t) \sum_{w \in \mathscr{W}} \Pr(w|t) \log_2 \big(\Pr(w|t) \big).$$

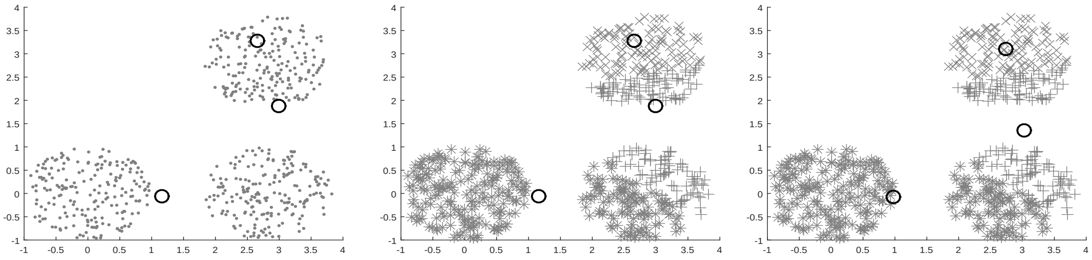
Here, W is a random variable of a patient being given a diagnosis w with $\mathscr{W}$ denoting a set diagnoses, and $\mathbf{T}$ is a random variable to represent the partition to which a patient belongs with support $\mathscr{T}$. Originating in information theory, conditional entropy describes how well a clustering $\mathbf{T}$ can be used to compress data while retaining all of the information in the data. The lower the entropy, the more $\mathbf{T}$ says about W. Although we assert conditional entropy is the appropriate metric to be used for this problem, it is not a commonly used dissimilarity function. As one might then suspect, the availability of software implementations of algorithms which find partitions of minimal conditional entropy pales in comparison to more common clustering algorithms.

One such algorithm, called **Lloyd's algorithm** or the k-means algorithm [8], seeks to solve the what is known as the k-means problem:

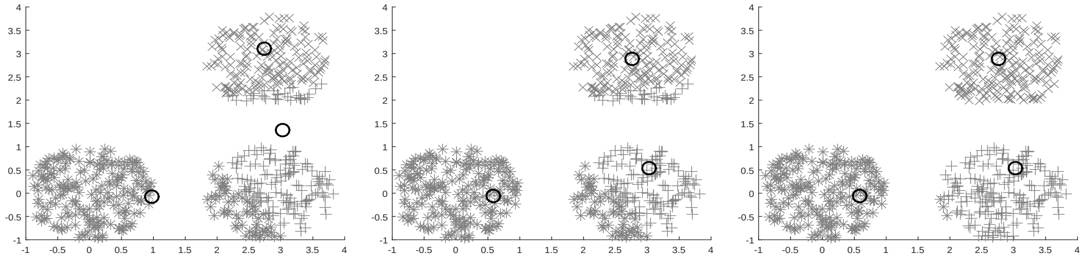
$$\begin{aligned} \text{minimize} \quad & \sum_{t=1}^{k} \sum_{x \in A_t} \left\| x - \frac{1}{|A_t|} \sum_{y \in A_t} y \right\|_2^2 \\ \text{subject to} \quad & A_1 \sqcup \cdots \sqcup A_k = P. \end{aligned} \tag{1}$$

This algorithm alternates between calculating centroids of proto-clusters and reassigning points according to the nearest centroid. An illustration of an implementation of Lloyd's algorithm is given in Figure 1. Due to its algorithmic simplicity, the real world applications of Lloyd's algorithm to solve the k-means problem are numerous. Take for example the case of image classification studied in [9] which used the k-means algorithm to achieve state-of-the-art classification accuracy on two benchmark databases commonly used in image classification. In [10], Lloyd's algorithm was used in medical image segmentation. The authors of [11] used it to map human vulnerability to toxic chemicals in Shanghai. Artificial neural networks and Lloyd's algorithm were used in [12, 13] to perform market segmentation. In [14], Lloyd's algorithm is applied dynamically to a database of songs to create a recommender system. Dimensionality reduction on a set of climatological performance metrics is performed in [15]. The work in [16] demonstrated the applicability of the Lloyd's algorithm in identifying geographic patterns in criminal activity and proposed their methodology could be used in homeland security. The authors of [17] use Lloyd's algorithm in detecting outliers in network traffic. Clearly, this algorithm is versatile.

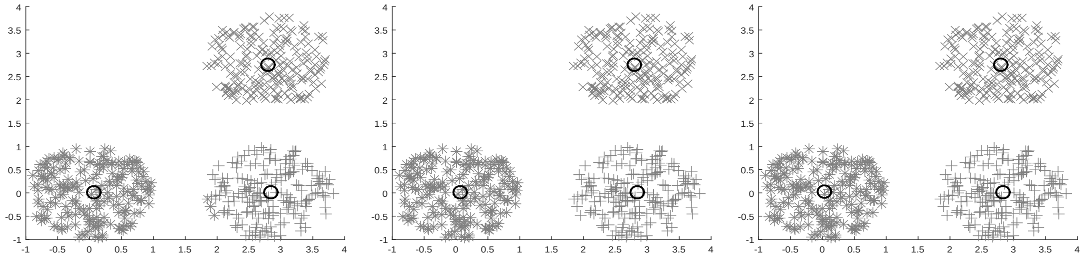
Moreover, Problem (1) is approximately derived from a related notion of entropy [18]. As such, one might suspect that near-optimal k-means partitions are also good clusterings under conditional entropy. In fact, under a certain reasonable distribu-



(a) Random Initialization **(b)** Assign to Clusters **(c)** Update Centroids



(d) Assign to Clusters **(e)** Update Centroids **(f)** Assign to Clusters



(g) Update Centroids **(h)** Assign to Clusters **(i)** Update Centroids; Completed Clustering

Figure 1. An Illustration of Lloyd's Algorithm The following has been performed using the k-means++ algorithm in MATLAB 2015a. Here we generate data for three clusters each having 200 points in $\mathbb{R}^2$ drawn uniformly from a ball of radius one with centers $(0,0), (2\sqrt{2},0)$, and $(2\sqrt{2},2\sqrt{2})$, respectively. In (a), we randomly initialize cluster centroids according to the k-means++ initialization scheme. We indicate cluster centroids with a circle. In (b), (d), (f), and (h) we assign points to clusters according to the nearest cluster centroid. A point is labeled as a $*, +$, or $\times$ depending on cluster membership. In (c), (e), (g), and (i) we calculate cluster centroids based on the points assigned to the previous clustering. Finally, we terminate the algorithm in (i) as the clustering does not change upon one more iteration of Lloyd's algorithm.

tion of data, the two objective functions are indeed closely related. Consider planted clusters where the data is tightly packed around cluster centroids. The planted partition gives a small k-means objective as all of the points are near their centriods. In the case of conditional entropy, the probability of a datapoint being near a centroid under the planted partition is close to either zero or one driving down entropy. Since the k-means problem has been extensively studied and quickly employed in various real-world applications, and further relates to the conditional entropy objective we desire, we seek a patient partition which optimizes the k-means objective in the hopes that it will lead to low-entropy, time-robust patient clusterings.

1.2 Certifiably Optimal k-means Clustering

Despite its widespread use, speed and applicability, Lloyd's algorithm has a few shortcomings. First, solving problem (1) is NP-hard in general [19], so Lloyd's algorithm does not exactly solve the k-means problem. Rather, Lloyd's algorithm (and its variants [20, 21]) may converge to local minima of the k-means objective (e.g., see section 5 of [22]). Furthermore, the output of Lloyd's algorithm does not indicate how far it is from optimal. As such, a slower algorithm that emits such a certificate may be preferable. Along these lines, convex relaxations provide a framework to approach such NP-hard problems. This framework is known as the "relax and round" paradigm: Given an optimization problem, first relax the feasibility region to a convex set, optimize subject to this larger set, and then round the optimal solution to a point in the original feasibility region.

One may seek approximation guarantees in this framework by relating the value of the rounded solution to the value of the optimal solution. Sometimes, the rounding step of the approximation algorithm is unnecessary because the convex relaxation happens to find a solution that is feasible in the original problem. This phenomenon is known as tightness (or integrality) of the convex relaxation. When exact recovery occurs, the algorithm not only provides a solution, but also a certificate of its optimality thanks to a key concept in convex optimization called convex duality. We refer the reader to [23, 24] for more information on this subject.

To date, the best convex relaxation of k-means is the semidefinite program (SDP)

Table 1. Comparison of k-means Clustering Algorithms

Clustering Method	Certificate of optimality?	Fast clustering?
Lloyd's algorithm	No	Yes
Semidefinite relaxation in [22]	Yes	No
PCC algorithm (Chapter II)	Yes	Yes

proposed in [22]. Unfortunately, semidefinite programming is quite slow, especially compared to Lloyd's algorithm. On the other hand, Lloyd's algorithm currently lacks a performance guarantee for an optimal partitioning. One could attempt to resolve this problem by creating a k-means solver which recovers planted clusters under a model with a reasonable definition of a cluster. The solver, however, may not produce a certificate of an optimal partitioning and the performance of the solver is overly dependent on the data following the model for which it was designed, which is not guaranteed.

Recently, Bandeira [25] combined fast solvers with the optimality certificates of convex relaxations to produce a new class of algorithms: **probably certifiably correct** (PCC) **algorithms**. The main idea here is that in some cases, given an integral solution of a convex relaxation, one may quickly compute a corresponding dual certificate. As such, given a solution from a fast solver, the computational burden of demonstrating optimality is simply to check whether that certificate is dual feasible. By applying the PCC framework, we achieve fast certifiable k-means clustering in this thesis.

1.3 Overview

We provide new results in clustering theory and apply these results to partition patients for the betterment of healthcare delivery in the Air Force. The following subsections give more detail regarding the content of the thesis, organized by chapter.

Note in this work we often refer the reader to the Appendix A for the proof of a statement to maintain greater readability in the body of the thesis.

Obtaining and Certifying k-means Optimal Partitions.

Given a clustering of data (say, from some fast algorithm such as Lloyd's algorithm), how do we certify k-means optimality? Moreover, can we certify optimality quickly? Currently, there is an answer to the former given in [22], but none to the latter. In Chapter II, we use a convex relaxation of problem (1) to develop a different certificate than the one given in [22], and then we leverage this certificate to develop a fast k-means certifier.

More precisely, we develop our certificate by finding a sufficient, deterministic condition on a set of points for which an SDP relaxation of the k-means problem is integral. To do so, we begin with the SDP proposed in [22] and characterize all of its dual certificates. With this characterization in mind, we seek to find a unique dual certificate given a set of points. As a slow certification process would be useless in applications, we provide a fast implementation of our prosed dual certificate that we call the **power iteration detector**. We conclude Chapter II with an aside by creating a 2-means solver that runs in quasilinear time which we call **spectral k-means clustering**.

k-means under the Stochastic Ball Model.

The deterministic condition we derive in Chapter II is sufficient to certify optimality. Naturally, we ask ourselves how close our sufficient condition is to being equivalent to k-means optimality. In other words, how "small" is the set of instances of optimal partitions which our deterministic condition fails to certify? Additionally, how does this condition perform relative to the certificate in [22]? It turns out that in many cases, our certificate achieves state-of-the-art performance.

To measure the performance of our deterministic condition, it is useful to define a distribution of points which we are tasked to cluster. We formalize a reasonable distribution currently existing in the literature which we call the **Stochastic Ball Model**. This model consists of balls of points each created from some rotation-invariant distribution. Notice in Figure 1, k-means was implemented on points distributed with the Stochastic Ball Model. The sufficient condition in Chapter II is used to obtain a lower bound on the minimum distance between centroids needed to guarantee planted cluster recovery under the Stochastic Ball Model. If we wish to cluster points in $\mathbb{R}^m$ into k clusters, our bound on the minimum inter-cluster distance demonstrates optimal performance if $k = O(m^{1/2})$. We provide a brief discussion on the limitations of our methodology and the applicability of k-means. In particular, we disprove a conjecture made in [22] regarding the minimum inter-cluster distance for which the k-means objective recovers planted clusters under the Stochastic Ball Model. Chapter III concludes with a discussion on a new method called spectral k-means clustering relevant for $k = 2$.

Bipartite Stochastic Block Model.

Despite the intuitiveness and usefulness of the Stochastic Ball Model, it is not well-suited to model the Air Force health care dataset. To correct this mismatch, Chapter IV provides a more appropriate model and analyzes different clustering algorithms with this model.

A well-studied and commonly used generative model for social and biological networks is called the **Stochastic Block Model** (see [26, 27, 28, 29]). For this model, a clustering amounts to a partition of a graph into well-connected subgraphs. Recent work in [30] has demonstrated the utility of algorithms which exploit data exhibiting

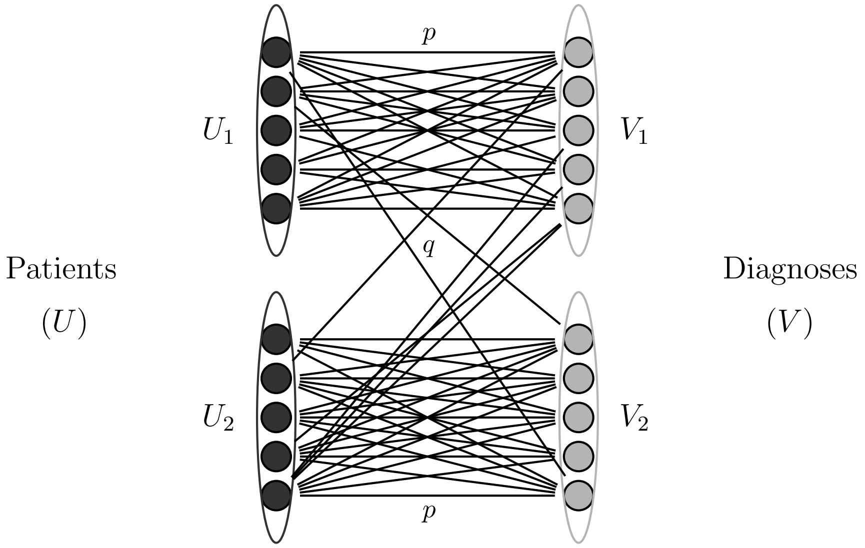


Figure 2. **Visualizing a BiSBM** The following random graph was generated using MATLAB.The partite on the left symbolizes a population of patients segmented into two sets U_1, U_2 with each vertex representing a patient. Similarly, the partite on the right symbolizes a set of diagnoses segmented into two sets V_1, V_2 with vertices representing diagnoses. An edge between the two partites symbolize a patient receiving a certain diagnosis. Here we generate a graph where edges between vertices in U_r and V_s where $r = s$ are drawn from a Bernoulli distribution with probability $p = 0.9$. For edges in the graph between vertices in U_r and V_s where $r \neq s$ are drawn from a Bernoulli distribution with probability $q = 0.1$.

additional bipartite structure. This follows what is called the **Bipartite Stochastic Block Model** (BiSBM) of which we give a visualization in Figure 2. Note that our dataset is naturally bipartite: every patient and every medical condition may be represented by a vertex, and we draw an edge whenever a patient exhibits a medical condition. Motivated by the applicability of this graph model, we demonstrate theoretical performance guarantees for clustering under the BiSBM.

To pursue such guarantees, a precise definition of a BiSBM is required. Definitions in the literature differ slightly between [31, 32] and [30]. Furthermore, we independently developed a similar notion of a BiSBM which differs from those that exist in the literature. As such, we provide a generalized BiSBM that encompasses all of the pre-existing, closely related notions of the BiSBM.

Next, we develop a PCC algorithm for the BiSBM. By employing the “relax and round” scheme, we devise a certificate for a maximum likelihood-based partition under the BiSBM. We demonstrate guarantees of recovering two planted clusters of patients with a thresholding algorithm. As it often happens that minimizing the k-means objective suffices to find a threshold-based clustering, we may use any of the previously discussed k-means algorithms to perform a thresholding. Furthermore, we provide an extension of patient thresholding under a BiSBM using k-means to obtain a heuristic to achieve bipartite planted cluster recovery. We provide numerical simulations which indicate that this bipartite cluster recovery heuristic performs well for a large region of the parameter space of our BiSBM. The PCC algorithm then concludes with an application of the power iteration detector to the previously developed certificate for a maximum likelihood partition with complexity time $O(n^2)$, where n denotes the number of vertices in a partition. Chapter IV concludes with another aside. Namely, we close a void in the literature by extending the power iteration detector to create a fast PCC algorithm (with complexity $O(n \log n)$ for n vertices) for the more general Stochastic Ball Model.

Creating a Robust Clustering for Air Force Health Care.

The DoD needs a plan to reduce the cost of health care. As discussed earlier, we seek time-robust partitions of patients for assignments to different IPUs. From a practical perspective, having too many clusters is undesirable. With this in mind, we illustrate how to determine the best number of clusters by finding the greatest decrease in conditional entropy as the number of clusters increase.

We apply Porter's patient partition to the first half of our dataset of medical records and calculate the conditional entropy of this partition on the second five years of the dataset. Afterwards, we create k-means partitions of the first five years of the dataset with the number of clusters ranging from two to ten. We measure the conditional entropy of these k-means partitions. We demonstrate that all of the

k-means clusterings outperform Porter's proposed partition in terms of conditional entropy, with the 2-means clustering achieving the best performance overall.

Using the heuristic developed in Chapter IV to recover a planted partition under the BiSBM, we provide a visualization of this bipartite structure which gives stark contrast between the clusters in the 2-means partition.

With the k-means partitions outperforming Porter's proposed partition, we have successfully demonstrated a proof of concept for data-driven decision making in the construction of IPUs in the Air Force. Of course, there are additional real-world circumstances and constraints which ought to be considered before organizing large-scale IPU assignments. In particular, modifications to Air Force policy would first require more input from health care experts.

II. Obtaining and Certifying k-means Optimal Partitions

In this chapter, we apply Bandeira's ideas to the k-means problem. Before proceeding we refer the reader to [24] for a brief overview of duality and certificates in convex programming and [23] for more detail regarding the topic of convex optimization in general.

We begin by using the convex relaxation of the k-means problem employed in [22] and characterize all of its dual certificates. Despite the relaxation not having a unique dual certificate for a k-means optimal solution, we develop a different certificate than the one developed in [22] which we design to certify a large family of k-means optimal solutions. From this relaxation we develop a different certificate than the one developed in [22]. With this certificate in hand, we develop a method to quickly certify a clustering of points achieves k-means optimality in quasilinear time. Namely, with the creation of this fast k-means certifier we have created a PCC algorithm for the k-means problem.

2.1 Previous Work

In this section we introduce the semidefinite relaxation of k-means which is presented in [22]. We conclude by characterizing all dual certificates of the SDP relaxation in a concise manner.

Let 1_A denote the indicator function of $A \subseteq \{1, \ldots, N\}$. Also, define the $N \times N$ matrix D by $D_{ij} := \|x_i - x_j\|_2^2$. Then denoting $c_t := \frac{1}{|A_t|}\sum_{x \in A_t} x$, we have

$$\begin{aligned}\frac{1}{|A_t|}\operatorname{Tr}(D 1_{A_t} 1_{A_t}^\top) &= \frac{1}{|A_t|}\sum_{x_i, x_j \in A_t} \|x_i - x_j\|_2^2 \\ &= \frac{1}{|A_t|}\sum_{x_i, x_j \in A_t} \|(x_i - c_t) - (x_j - c_t)\|_2^2 = 2\sum_{x_i \in A_t} \|x_i - c_t\|_2^2,\end{aligned}$$

where the last step can be seen by expanding the square and distributing the sums. As such, taking $X := \sum_{t=1}^{k} \frac{1}{|A_t|} 1_{A_t} 1_{A_t}^\top$, the k-means objective may be expressed as

$$\sum_{t=1}^{k} \sum_{x_i \in A_t} \|x_i - c_t\|_2^2 = \frac{1}{2} \sum_{t=1}^{k} \frac{1}{|A_t|} \operatorname{Tr}(D 1_{A_t} 1_{A_t}^\top) = \frac{1}{2} \operatorname{Tr}\left(D \sum_{t=1}^{k} \frac{1}{|A_t|} 1_{A_t} 1_{A_t}^\top \right) = \frac{1}{2} \operatorname{Tr}(DX).$$

Observe that X satisfies several convex constraints, and so we may relax the region of optimization to produce a convex program, namely, the semidefinite relaxation proposed in [22]:

$$\begin{aligned} \text{minimize} \quad & \operatorname{Tr}(DX) \\ \text{subject to} \quad & \operatorname{Tr}(X) = k \\ & X1 = 1 \\ & X \geq 0 \\ & X \succeq 0 \end{aligned} \tag{2}$$

where $X \succeq 0$ is a statement that the eigenvalues of X are nonnegative (more generally, for square matrices K and L, if the greatest eigenvalue of L is no more than the least eigenvalue of K, then $K \succeq L$). The following is the dual program of (2):

$$\begin{aligned} \text{minimize} \quad & kz + \sum_{i=1}^{N} \alpha_i \\ \text{subject to} \quad & zI + \sum_{i=1}^{N} \alpha_i \cdot \frac{1}{2}(e_i 1^\top + 1 e_i^\top) - \sum_{i=1}^{N} \sum_{j=i}^{N} \beta_{ij} \cdot \frac{1}{2}(e_i e_j^\top + e_j e_i^\top) + D \succeq 0 \\ & \beta \geq 0 \end{aligned} \tag{3}$$

For notational simplicity, from this point forward, we organize indices according to clusters. For example, 1_a shall denote the indicator function of the ath cluster. Also, we shuffle the rows and columns of X and D into blocks that correspond to clusters; for example, the (i,j)th entry of the (a,b)th block of D is given by $D_{ij}^{(a,b)}$. We also index α in terms of clusters; for example, the ith entry of the ath block of α is denoted $\alpha_{a,i}$. For β, we identify

$$\beta := \sum_{i=1}^{N} \sum_{j=i}^{N} \beta_{ij}(e_i e_j^\top + e_j e_i^\top).$$

Indeed, when $i \leq j$, the (i,j)th entry of β is β_{ij}. We also consider β as having its rows and columns shuffled according to clusters, so that the (i,j)th entry of the (a,b)th block is $\beta_{ij}^{(a,b)}$.

With this notation, the following proposition characterizes all possible dual certificates of (2). The proof for the proposition is placed in Appendix A as it is merely a more concise rewriting of the proof in [22]:

Proposition 1 (cf. [22])**.** *Take* $X := \sum_{a=1}^{k} \frac{1}{n_a} 1_a 1_a^\top$*, where* n_a *denotes the number of points in cluster* a*. The following are equivalent:*

(a) X *is a solution to the semidefinite relaxation* (2)*.*

(b) Every solution to the dual program (3) *satisfies*

$$Q^{(a,a)} 1 = 0, \qquad \beta^{(a,a)} = 0 \qquad \forall a \in \{1, \ldots, k\},$$

where Q *is the matrix in the semidefinite constraint in* (3)*.*

(c) Every solution to the dual program (3) *satisfies*

$$\alpha_{a,r} = -\frac{1}{n_a} z + \frac{1}{n_a^2} 1^\top D^{(a,a)} 1 - \frac{2}{n_a} e_r^\top D^{(a,a)} 1 \qquad \forall a \in \{1, \ldots, k\},\ r \in a.$$

The following section will leverage this result to identify a condition on D that implies that the SDP (2) has an integral solution.

2.2 Selecting a Dual Certificate

The goal is to certify when the SDP-optimal solution is integral. In this event, Proposition 1 characterizes acceptable dual certificates (z, α, β), but this information fails to uniquely determine a certificate. In this subsection, we will motivate the application of additional constraints on dual certificates so as to identify certifiable instances.

We start by reviewing the characterization of dual certificates (z, α, β) provided in Proposition 1. In particular, α is completely determined by z, and so z and β are the only remaining free variables. Indeed, for every $a, b \in \{1, \ldots, k\}$, we have

$$\begin{aligned}
&\left(\sum_{t=1}^{k} \sum_{i \in t} \alpha_{t,i} \cdot \frac{1}{2}(e_{t,i} 1^\top + 1 e_{t,i}^\top)\right)^{(a,b)} \\
&\quad = \sum_{i \in a} \alpha_{a,i} \cdot \frac{1}{2} e_i 1^\top + \sum_{j \in b} \alpha_{b,j} \cdot \frac{1}{2} 1 e_j^\top \\
&\quad = -\frac{1}{2}\left(\frac{1}{n_a} + \frac{1}{n_b}\right) z + \sum_{i \in a} \left(\frac{1}{n_a^2} 1^\top D^{(a,a)} 1 - \frac{2}{n_a} e_i^\top D^{(a,a)} 1\right) \frac{1}{2} e_i 1^\top \\
&\qquad + \sum_{j \in b} \left(\frac{1}{n_b^2} 1^\top D^{(b,b)} 1 - \frac{2}{n_b} e_j^\top D^{(b,b)} 1\right) \frac{1}{2} 1 e_j^\top,
\end{aligned}$$

and since

$$Q = zI + \sum_{t=1}^{k} \sum_{i \in t} \alpha_{t,i} \cdot \frac{1}{2}(e_{t,i} 1^\top + 1 e_{t,i}^\top) - \frac{1}{2}\beta + D,$$

we may write $Q = z(I - E) + M - B$, where

$$\begin{aligned}
E^{(a,b)} &:= \frac{1}{2}\bigg(\frac{1}{n_a} + \frac{1}{n_b}\bigg)11^\top && (4)\\
M^{(a,b)} &:= D^{(a,b)} + \sum_{i\in a}\bigg(\frac{1}{n_a^2}1^\top D^{(a,a)}1 - \frac{2}{n_a}e_i^\top D^{(a,a)}1\bigg)\frac{1}{2}e_i1^\top \\
&\qquad + \sum_{j\in b}\bigg(\frac{1}{n_b^2}1^\top D^{(b,b)}1 - \frac{2}{n_b}e_j^\top D^{(b,b)}1\bigg)\frac{1}{2}1e_j^\top && (5)\\
B^{(a,b)} &= \frac{1}{2}\beta^{(a,b)}
\end{aligned}$$

for every $a, b \in \{1, \ldots, k\}$. The following is one way to formulate our task: Given D and a clustering (which in turn determines E and M), determine whether there exist feasible z and B such that $Q \succeq 0$; here, feasibility only requires B to be symmetric with nonnegative entries and $B^{(a,a)} = 0$ for every $a \in \{1, \ldots, k\}$. We opt for a slightly more modest goal: Find $z = z(D)$ and $B = B(D)$ such that $Q \succeq 0$ for a large family of D's.

Before determining z and B, we first analyze E:

Lemma 2. *Let E be the matrix defined by* (4). *Then* $\operatorname{rank}(E) \in \{1, 2\}$. *The eigenvalue of largest magnitude is $\lambda \geq k$, and when* $\operatorname{rank}(E) = 2$, *the other nonzero eigenvalue of E is negative. The eigenvectors corresponding to nonzero eigenvalues lie in the span of $\{1_a\}_{a=1}^k$.*

Proof. Writing

$$E = \sum_{a=1}^k \sum_{b=1}^k \frac{1}{2}\bigg(\frac{1}{n_a} + \frac{1}{n_b}\bigg)1_a1_b^\top = \frac{1}{2}\bigg(\sum_{a=1}^k \frac{1}{n_a}1_a\bigg)1^\top + \frac{1}{2}1\bigg(\sum_{b=1}^k \frac{1}{n_b}1_b\bigg)^\top,$$

we see that $\operatorname{rank}(E) \in \{1, 2\}$, and it is easy to calculate $1^\top E1 = Nk$ and $\operatorname{Tr}(E) = k$.

Observe that

$$\lambda = \sup_{\substack{x \in \mathbb{R}^N \\ \|x\|_2 = 1}} x^\top E x \geq \frac{1}{N} 1^\top E 1 = k,$$

and combining with $\operatorname{rank}(E) \leq 2$ and $\operatorname{Tr}(E) = k$ then implies that the other nonzero eigenvalue (if there is one) is negative. Finally, any eigenvector of E with a nonzero eigenvalue necessarily lies in the column space of E, which is a subspace of $\operatorname{span}\{1_a\}_{a=1}^k$ by the definition of E. $\square$

When finding z and B such that $Q = z(I - E) + M - B \succeq 0$, it will be useful that $I - E$ has only one negative eigenvalue to correct. Let v denote the corresponding eigenvector. Then we will pick B so that v is also an eigenvector of $M - B$. Since we want $Q \succeq 0$ for as many instances of D as possible, we will then pick z as large as possible, thereby sending v to the nullspace of Q. Unfortunately, the authors found that this constraint fails to uniquely determine B in general. Instead, we impose a stronger constraint:

$$Q 1_a = 0 \qquad \forall a \in \{1, \ldots, k\}.$$

(This constraint implies $Qv = 0$ by Lemma 2.) To see the implications of this constraint, note that we already necessarily have

$$(Q1_a)_a = \Big((z(I-E)+M-B)1_a \Big)_a = z(I-E^{(a,a)})1 + M^{(a,a)}1 - B^{(a,a)}1 = z - \frac{z}{n_a} 11^\top 1 = 0,$$

and so it remains to impose

$$\begin{aligned} 0 = (Q1_b)_a &= \Big((z(I-E) + M - B)1_b \Big)_a \\ &= -zE^{(a,b)}1 + M^{(a,b)}1 - B^{(a,b)}1 = -z\frac{n_a + n_b}{2n_a}1 + M^{(a,b)}1 - B^{(a,b)}1. \quad (6) \end{aligned}$$

In order for there to exist a vector $B^{(a,b)}1 \geq 0$ that satisfies (6), z must satisfy

$$z\frac{n_a + n_b}{2n_a} \leq \min(M^{(a,b)}1),$$

and since z is independent of (a, b), we conclude that

$$z \leq \min_{\substack{a,b\in\{1,\ldots,k\} \\ a\neq b}} \frac{2n_a}{n_a + n_b} \min(M^{(a,b)}1). \tag{7}$$

Again, in order to ensure $z(I - E) + M - B \succeq 0$ for as many instances of D as possible, we intend to choose z as large as possible. Luckily, there is a choice of B which satisfies (6) for every (a, b), even when z satisfies equality in (7). Indeed, we define

$$u_{(a,b)} := M^{(a,b)}1 - z\frac{n_a + n_b}{2n_a}1, \qquad \rho_{(a,b)} := u_{(a,b)}^\top 1, \qquad B^{(a,b)} := \frac{1}{\rho_{(b,a)}} u_{(a,b)} u_{(b,a)}^\top \tag{8}$$

for every $a, b \in \{1, \ldots, k\}$ with $a \neq b$. Then by design, B immediately satisfies (6). Also, note that $\rho_{(a,b)} = \rho_{(b,a)}$, and so $B^{(b,a)} = (B^{(a,b)})^\top$, meaning B is symmetric. Finally, we necessarily have $u_{(a,b)} \geq 0$ (and thus $\rho_{(a,b)} \geq 0$) by (7), and we implicitly require $\rho_{(a,b)} > 0$ for division to be permissible. As such, we also have $B^{(a,b)} \geq 0$, as desired.

Now that we have selected z and B, it remains to check that $Q \succeq 0$. By construction, we already have $\Lambda := \operatorname{span}\{1_a\}_{a=1}^k$ in the nullspace of Q, and so it suffices to ensure

$$0 \preceq P_{\Lambda^\perp} Q P_{\Lambda^\perp} = P_{\Lambda^\perp}\Big(z(I - E) + M - B\Big)P_{\Lambda^\perp} = zP_{\Lambda^\perp} + P_{\Lambda^\perp}(M - B)P_{\Lambda^\perp}.$$

Here, $P_{\Lambda^\perp}$ denotes the orthogonal projection onto the orthogonal complement of Λ. Rearranging then gives the following result:

Theorem 3. *Take* $X := \sum_{t=1}^{k} \frac{1}{n_t} 1_t 1_t^\top$*, where* n_t *denotes the number of points in cluster* t*. Consider* M *defined by* (5)*, pick* z *so as to satisfy equality in* (7)*, take* B *defined by* (8)*, and let* Λ *denote the span of* $\{1_t\}_{t=1}^{k}$*. Then* X *is a solution to the semidefinite relaxation* (2) *if*

$$P_{\Lambda^\perp}(B - M)P_{\Lambda^\perp} \preceq zP_{\Lambda^\perp}. \tag{9}$$

We take a moment to emphasize the importance of Theorem 3. First, the theorem is the sufficient deterministic condition for k-means optimality for which we have been searching. That is, we can certify k-means optimality merely given a clustering of data. Furthermore, the theorem allows us to create the fast k-means certifier which fills the void in the literature of practically certifying k-means optimal partitions.

2.3 A Fast Implementation of the Dual Certificate

As discussed in the Introduction, a probably certifiably correct algorithm depends on the ability to test whether a candidate solution is optimal. In this section, we leverage the certificate (9) to provide such a test for the k-means problem. We first show how to solve a more general problem from linear algebra, and then we apply our solution to devise a fast test for k-means optimality.

Leading Eigenvector Hypothesis Test.

In this subsection, we are concerned with the following fundamental problem from linear algebra:

Problem 4. *Given a symmetric matrix* $A \in \mathbb{R}^{n\times n}$ *and an eigenvector* v *of* A*, determine whether the span of* v *is the unique leading eigenspace, that is, the corresponding eigenvalue* λ *has multiplicity* 1 *and satisfies* $|\lambda| > |\lambda'|$ *for every other eigenvalue* λ' *of*

A.

To solve this problem, one might be inclined to apply the power method:

Proposition 5 (Theorem 8.2.1 in [33])**.** *Let $A \in \mathbb{R}^{n\times n}$ be a symmetric matrix with eigenvalues $\{\lambda_i\}_{i=1}^n$ (counting multiplicities) satisfying*

$$|\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|,$$

and with corresponding orthonormal eigenvectors $\{v_i\}_{i=1}^n$. Pick a unit-norm vector $q_0 \in \mathbb{R}^n$ and consider the power iteration $q_{j+1} := Aq_j/\|Aq_j\|_2$. If q_0 is not orthogonal to v_1, then

$$(v_1^\top q_j)^2 \geq 1 - \Big((v_1^\top q_0)^{-2} - 1\Big)\left(\frac{\lambda_2}{\lambda_1}\right)^{2j}.$$

Notice that the above convergence guarantee depends on the quality of the initialization q_0. To use this guarantee, draw q_0 at random from the unit sphere so that q_0 is not orthogonal to v_1 almost surely; one might then analyze the statistics of $v_1^\top q_0$ to produce statistics on the time required for convergence. The power method is typically used to find a leading eigenvector, but for our problem, we already have access to an eigenvector v, and we are tasked with determining whether v is the unique leading eigenvector. Intuitively, if you run the power method from a random initialization and it happens to converge to v, then this would have been a remarkable coincidence if v were not the unique leading eigenvector. Since we will only run finitely many iterations, how do we decide when we are sufficiently confident? The remainder of this subsection answers this question.

Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ and a unit eigenvector v of A, consider the

Algorithm 1: Power iteration detector

```
Input: Symmetric matrix $A \in \mathbb{R}^{n\times n}$, unit eigenvector $v \in \mathbb{R}^n$, tolerance $\epsilon > 0$
Output: Decision of whether to accept $H_0$ or to reject $H_0$ and accept $H_1$ as
        given in (10)
$\lambda \leftarrow v^\top A v$
Draw $q$ uniformly at random from the unit sphere in $\mathbb{R}^n$
while no decision has been made do
    if $|q^\top A q| > |\lambda|$ then
        Print accept $H_0$
    else if $(v^\top q)^2 \geq 1 - \epsilon$ then
        Print reject $H_0$ and accept $H_1$
    end
    $q \leftarrow Aq/\|Aq\|_2$
end
```

hypotheses

$$\begin{aligned} H_0\colon &\quad \operatorname{span}(v) \text{ is not the unique leading eigenspace of } A, \\ H_1\colon &\quad \operatorname{span}(v) \text{ is the unique leading eigenspace of } A. \end{aligned} \tag{10}$$

To test these hypotheses, pick a tolerance $\epsilon > 0$ and run the power iteration detector (see Algorithm 1). This detector terminates either by accepting H_0 or by rejecting H_0 and accepting H_1. We say the detector **fails to reject** H_0 if it either accepts H_0 or fails to terminate. Before analyzing this detector, we consider the following definition:

Definition 6. *Given a symmetric matrix $A \in \mathbb{R}^{n\times n}$ and unit eigenvector v of A, put $\lambda = v^\top A v$, and let λ_1 denote a leading eigenvalue of A (i.e., $|\lambda_1| = \|A\|_{2\to 2}$). We say (A, v) is* ***degenerate*** *if*

(a) the eigenvalue λ of A has multiplicity ≥ 2,

(b) $-\lambda$ is an eigenvalue of A, or

(c) $-\lambda_1$ is an eigenvalue of A.

Theorem 7. *Consider the power iteration detector (Algorithm 1), let q_j denote q at the jth iteration (with q_0 being the initialization), and let π_ϵ denote the probability that $(e_1^\top q_0)^2 < \epsilon$.*

(i) (A, v) is degenerate only if H_0 holds. If (A, v) is non-degenerate, then the power iteration detector terminates in finite time with probability 1.

(ii) The power iteration detector incurs the following error rates:

$$\Pr\Big(\textit{reject } H_0 \textit{ and accept } H_1 \,\Big|\, H_0 \Big) \leq \pi_\epsilon, \qquad \Pr\Big(\textit{fail to reject } H_0 \,\Big|\, H_1 \Big) = 0.$$

(iii) If H_1 holds, then

$$\min\Big\{ j : (v^\top q_j)^2 > 1 - \epsilon \Big\} \leq \frac{3 \log(1/\epsilon)}{2 \log(\lambda_1/\lambda_2)} + 1$$

with probability $\geq 1 - \pi_\epsilon$.

Proof. Denote the eigenvalues of A by $\{\lambda_i\}_{i=1}^n$ (counting multiplicities), ordered in such a way that $|\lambda_1| \geq \cdots \geq |\lambda_n|$, and consider the corresponding orthonormal eigenvectors $\{v_i\}_{i=1}^n$, where $v = v_p$ for some p.

For (i), first note that H_1 implies that (A, v) is non-degenerate, and so the contrapositive gives the first claim. Next, suppose (A, v) is non-degenerate. If H_1 holds, then $(v^\top q_j)^2 \to 1$ by Proposition 5 provided q_0 is not orthogonal to v, and so the power iteration detector terminates with probability 1. Otherwise, H_0 holds, and so the non-degeneracy of (A, v) implies that the eigenspace corresponding to λ_1 is the unique leading eigenspace of A, and furthermore, $|\lambda_1| > |\lambda|$. Following the proof of Theorem 8.2.1 in [33], we also have

$$q_j^\top A q_j = \frac{q_0^\top A^{2j+1} q_0}{q_0^\top A^{2j} q_0} = \frac{\sum_{i=1}^n (v_i^\top q_j)^2 \lambda_i^{2j+1}}{\sum_{i=1}^n (v_i^\top q_j)^2 \lambda_i^{2j}}.$$

Putting $r := \min\{i : |\lambda_i| < |\lambda_1|\}$, then

$$\begin{aligned} |q_j^\top A q_j - \lambda_1| &= \left| \frac{\sum_{i=1}^n (v_i^\top q_j)^2 \lambda_i^{2j} (\lambda_i - \lambda_1)}{\sum_{i=1}^n (v_i^\top q_j)^2 \lambda_i^{2j}} \right| \\ &\leq \frac{|\lambda_1 - \lambda_n|}{\|P_{\lambda_1} q_0\|_2^2} \sum_{i=r}^n (v_i^\top q_j)^2 \left(\frac{\lambda_i}{\lambda_1}\right)^{2j} \leq |\lambda_1 - \lambda_n| \left(\frac{1 - \|P_{\lambda_1} q_0\|_2^2}{\|P_{\lambda_1} q_0\|_2^2}\right) \left(\frac{\lambda_r}{\lambda_1}\right)^{2j}, \end{aligned}$$

where P_{λ_1} denotes the orthogonal projection onto the eigenspace corresponding to λ_1. As such, $|q_j^\top A q_j| \to |\lambda_1| > |\lambda|$ provided $P_{\lambda_1} q_0 \neq 0$, and so the power iteration detector terminates with probability 1.

For (ii), we first consider the case of a false positive. Taking $v = v_p$ for $p \neq 1$, note that $(v^\top q_j)^2 > 1 - \epsilon$ implies

$$\epsilon > 1 - (v^\top q_j)^2 = \|q_j\|_2^2 - (v_p^\top q_j)^2 = \sum_{\substack{i=1 \\ i \neq p}}^n (v_i^\top q_j)^2 \geq (v_1^\top q_j)^2.$$

Also, since $\|Ax\|_2 \leq |\lambda_1| \|x\|_2$ for all $x \in \mathbb{R}^n$, we have that $(v_1^\top q_j)^2$ monotonically increases with j:

$$(v_1^\top q_{j+1})^2 = \left(v_1^\top \frac{A q_j}{\|A q_j\|_2}\right)^2 = \frac{(\lambda_1 v_1^\top q_j)^2}{\|A q_j\|_2^2} \geq \frac{(v_1^\top q_j)^2}{\|q_j\|^2} = (v_1^\top q_j)^2.$$

As such, $\epsilon > (v_1^\top q_j)^2 \geq (v_1^\top q_0)^2$. Overall, when H_0 holds, the power iteration detector rejects H_0 only if q_0 is initialized poorly, i.e., $(v_1^\top q_0)^2 < \epsilon$, which occurs with probability π_ϵ (since q_0 has a rotation-invariant probability distribution). For the false negative error rate, note that Proposition 5 gives that H_1 implies convergence $(v^\top q_j)^2 \to 1$ provided q_0 is not orthogonal to v, i.e., with probability 1.

For (iii), we want j such that $(v^\top q_j)^2 > 1 - \epsilon$. By Proposition 5, it suffices to have

$$\Big((v_1^\top q_0)^{-2} - 1\Big) \left(\frac{\lambda_2}{\lambda_1}\right)^{2j} < \epsilon.$$

In the event that $(v_1^\top q_0)^2 \geq \epsilon$ (which has probability $1-\pi_\epsilon$), it further suffices to have

$$\epsilon^{-2}\left(\frac{\lambda_2}{\lambda_1}\right)^{2j} < \epsilon.$$

Taking logs and rearranging then gives the result. $\square$

To estimate ϵ and π_ϵ, first note that q_0 has a rotation-invariant probability distribution, and so linearity of expectation gives

$$\mathbb{E}\big[(e_1^\top q_0)^2\big] = \frac{1}{n}\sum_{i=1}^{n}\mathbb{E}\big[(e_i^\top q_0)^2\big] = \frac{1}{n}\mathbb{E}\|q_0\|_2^2 = \frac{1}{n}.$$

Thus, in order to make π_ϵ small, we should expect to have $\epsilon \ll 1/n$. The following lemma gives that such choices of ϵ suffice for π_ϵ to be small:

Lemma 8. *If $\epsilon \geq n^{-1}e^{-2n}$, then $\pi_\epsilon \leq 3\sqrt{n\epsilon}$.*

Proof. First, observe that $(e_1^\top q_0)^2$ is equal in distribution to Z^2/Q, where Z has standard normal distribution and Q has chi-squared distribution with n degrees of freedom (Z and Q are independent). The probability density function of Z has a maximal value of $1/\sqrt{2\pi}$ at zero, and so

$$\Pr\Big(Z^2 < a\Big) \leq \sqrt{\frac{2a}{\pi}}.$$

Also, Lemma 1 in [34] gives

$$\Pr\Big(Q \geq n + 2\sqrt{nx} + 2x\Big) \leq e^{-x} \qquad \forall x > 0.$$

Therefore, picking $a = 5n\epsilon$ and $x = n$, the union bound gives

$$\begin{aligned}\Pr\Big((e_1^\top q_0)^2 < \epsilon\Big) = \Pr\left(\frac{Z^2}{Q} < \epsilon\right) &\leq \Pr\Big(Z^2 < 5n\epsilon\Big) + \Pr\Big(Q > 5n\Big) \\ &\leq \sqrt{\frac{10n\epsilon}{\pi}} + e^{-n} \leq 3\sqrt{n\epsilon}. \qquad \square\end{aligned}$$

Overall, if we take $\epsilon = n^{-(2c+1)}$ for $c > 0$, then if H_0 is true, our detector will produce a false positive with probability $O(n^{-c})$. On the other hand, if H_1 is true, then with probability $1 - O(n^{-c})$, our detector will reject H_0 after $O_\delta(c \log n)$ power iterations, provided $|\lambda_2| \leq (1-\delta)|\lambda_1|$.

Testing Optimality with the Power Iteration Detector.

In this subsection, we leverage the power iteration detector to test k-means optimality. Note that the sufficient condition (9) holds if and only if $v := \frac{1}{\sqrt{N}}1$ is a leading eigenvector of the matrix

$$A := \frac{z}{N}11^\top + P_{\Lambda^\perp}(B - M)P_{\Lambda^\perp} = \frac{z}{N}11^\top + P_{\Lambda^\perp}(B - D)P_{\Lambda^\perp}. \tag{11}$$

(The second equality follows from distributing the $P_{\Lambda^\perp}$'s and recalling the definition of M in (5).) As such, it suffices that (A, v) satisfy H_1 in (10). Overall, given a collection of points $P = \{x_i\}_{i=1}^N \subseteq \mathbb{R}^m$ and a proposed partition $A_1 \sqcup \cdots \sqcup A_k = P$, we can produce the corresponding matrix A (defined above) and then run the power iteration detector of the previous subsection to test (9). In particular, a positive test with tolerance ϵ will yield $\geq 1 - \pi_\epsilon$ confidence that the proposed partition is optimal under the k-means objective. Furthermore, as we detail below, the matrix–vector products computed in the power iteration detector have a computationally cheap implementation.

Given an $m \times n_a$ matrix $\Phi_a = [x_{a,1} \cdots x_{a,n_a}]$ for each $a \in \{1, \ldots, k\}$, we follow the following procedure to implement the corresponding function $x \mapsto Ax$ as defined in (11):

1. Compute $\nu_a \in \mathbb{R}^{n_a}$ such that $(\nu_a)_i = \|x_{a,i}\|_2^2$ for every $a \in \{1, \ldots, k\}$ in $O(mN)$ operations.

   Let $\nu \in \mathbb{R}^N$ denote the vector whose ath block is ν_a.

2. Define the function $(a, b, x) \mapsto D^{(a,b)}x$ such that $D^{(a,b)} = \nu_a 1^\top - 2\Phi_a^\top \Phi_b + 1\nu_b^\top$.

   Running this function costs $O(m(n_a + n_b))$ operations.

3. Define the function $x \mapsto Dx$ such that $D = \nu 1^\top - 2\Phi^\top \Phi + 1\nu^\top$, where $\Phi = [\Phi_1 \cdots \Phi_k]$.

   Running this function costs $O(mN)$ operations.

4. Compute $\mu_a = \frac{1}{2}(\frac{1}{n_a^2} 11^\top - \frac{2}{n_a} I) D^{(a,a)} 1$ for every $a \in \{1, \ldots, k\}$ in $O(mN)$ operations.

5. Define the function $(a, b, x) \mapsto M^{(a,b)}x$ such that $M^{(a,b)} = D^{(a,b)} + \mu_a 1^\top + 1\mu_b^\top$.

   Running this function costs $O(m(n_a + n_b))$ operations.

6. Compute $z = \min_{a \neq b} \frac{2n_a}{n_a + n_b} \min(M^{(a,b)} 1)$ in $O(kmN)$ operations.

7. Compute $u_{(a,b)} = M^{(a,b)} 1 - z \frac{n_a + n_b}{2n_a} 1$ for every $a, b \in \{1, \ldots, k\}$, $a \neq b$ in $O(kmN)$ operations.

8. Compute $\rho_{(a,b)} = u_{(a,b)}^\top 1$ for every $a, b \in \{1, \ldots, k\}$, $a \neq b$ in $O(kN)$ operations.

9. Define the function $x \mapsto Bx$ such that the ath block of the output is given by

$$(Bx)_a = \sum_{\substack{b=1 \\ b \neq a}}^{k} \frac{u_{(a,b)} u_{(b,a)}^\top x_b}{\rho_{(b,a)}}.$$

   Running this function costs $O(kmN)$ operations.

10. Define the function $x \mapsto P_{\Lambda^\perp}x$ such that $P_{\Lambda^\perp} = I - \sum_{a=1}^{k} \frac{1}{n_a} 1_a 1_a^\top$.

    Running this function costs $O(N)$ operations.

11. Define the function $x \mapsto Ax$ such that $A = \frac{z}{N} 11^\top + P_{\Lambda^\perp}(B - D)P_{\Lambda^\perp}$.

    Running this function costs $O(kmN)$ operations.

Overall, after $O(kmN)$ operations of preprocessing, one may compute the function $x \mapsto Ax$ for any given x in $O(kmN)$ operations. (Observe that this is the same complexity as each iteration of Lloyd's algorithm, and as we illustrate in Figure 3, the runtimes are comparable.)

2.4 Miscellaneous Extentions

In this section we employ a slightly different relax and round scheme to obtain a fast 2-means solver as an aside. In [35], it was shown that k-means is equivalent to the following program:

$$\begin{aligned} \text{minimize} \quad & \operatorname{Tr}(DX) \\ \text{subject to} \quad & X^\top = X \\ & X^2 = X \\ & \operatorname{Tr}(X) = k \\ & X1 = 1 \\ & X \geq 0 \end{aligned} \tag{12}$$

One may quickly observe that the SDP (2) we analyzed in Section 2.1 is a relaxation of this program. In this section, we consider another relaxation of (12), obtained by discarding the $X \geq 0$ constraint (this is known as the **spectral clustering** relaxation [36, 37]). To analyze this relaxation, we first denote the $m \times N$ matrix $\Phi = [x_1 \cdots x_N]$. Without loss of generality, the data set is centered at the origin so that $\Phi 1 = 0$. Letting ν denote the $N \times 1$ vector with $\nu_i = \|x_i\|_2^2$, then

$$D_{ij} = \|x_i - x_j\|_2^2 = \|x_i\|_2^2 - 2x_i^\top x_j + \|x_j\|_2^2 = (\nu 1^\top - 2\Phi^\top \Phi + 1\nu^\top)_{ij}.$$

As such, $D = \nu 1^\top - 2\Phi^\top \Phi + 1\nu^\top$, and so the constraints $X = X^\top$ and $X1 = 1$ together imply an alternative expression for the objective function:

$$\begin{aligned}\operatorname{Tr}(DX) &= \operatorname{Tr}(\nu 1^\top X - 2\Phi^\top \Phi X + 1\nu^\top X)\\ &= \operatorname{Tr}(\nu 1^\top X^\top) - 2\operatorname{Tr}(\Phi^\top \Phi X) + \operatorname{Tr}(X1\nu^\top)\\ &= 2\nu^\top 1 - 2\operatorname{Tr}(\Phi^\top \Phi X).\end{aligned}$$

We conclude that minimizing $\operatorname{Tr}(DX)$ is equivalent to maximizing $\operatorname{Tr}(\Phi^\top \Phi X)$.

Next, we observe that the feasible X in our relaxation are precisely the rank-k $N \times N$ orthogonal projection matrices satisfying $X1 = 1$. This in turn is equivalent to X having the form $X = \frac{1}{N}11^\top + Y$, where Y is a rank-$(k-1)$ $N \times N$ orthogonal projection matrix satisfying $Y1 = 0$. Discarding the $Y1 = 0$ constraint produces the following relaxation of (12):

$$\begin{aligned}\text{maximize} \quad & \operatorname{Tr}(\Phi^\top \Phi Y)\\ \text{subject to} \quad & Y^\top = Y\\ & Y^2 = Y\\ & \operatorname{Tr}(Y) = k-1\end{aligned} \tag{13}$$

For general values of k, this program amounts to finding $k-1$ principal components of the data. Recalling our initial clustering goal, after finding the optimal Y, it remains to take $X = \frac{1}{N}11^\top + Y$ and then round to a nearby member of the feasibility region in (12). Below, we describe a suitable rounding procedure for the special case where

$k = 2$.

When $k = 2$, the solution to (13) has the form $Y = yy^\top$, where y is a leading unit eigenvector of $\Phi^\top\Phi$. Our task is to find a matrix of the form $\frac{1}{|A|}1_A1_A^\top + \frac{1}{|B|}1_B1_B^\top$ with $A \sqcup B = \{1, \ldots, N\}$ that is close to $\frac{1}{N}11^\top + yy^\top$. To this end, it seems natural to consider

$$A_\theta := \{i : y_i < \theta\}, \qquad B_\theta := A_\theta^c$$

for some threshold θ. Since the data is centered ($\Phi 1 = 0$), one may be inclined to take $\theta = 0$, but this will be a poor choice if the true clusters have significantly different numbers of points. Instead, we select the θ which minimizes the k-means objective of (A_θ, B_θ). Since we only need to consider $N - 1$ choices of θ, this is plausibly tractable, although computing the k-means objective once costs $O(mN)$ operations, and so some care is necessary to keep the algorithm fast.

We will show how to find the optimal (A_θ, B_θ) in $O((m + \log N)N)$ operations using a simple dynamic program. Order the indices so that $y_1 \leq \cdots \leq y_N$. Then the function to minimize is

$$f(i) := \frac{1}{i} \underbrace{\sum_{j=1}^{i}\sum_{j'=1}^{i} \|x_j - x_{j'}\|_2^2}_{v_i} + \frac{1}{N-i} \underbrace{\sum_{j=i+1}^{N}\sum_{j'=i+1}^{N} \|x_j - x_{j'}\|_2^2}_{v_i^c}.$$

Expanding the square and distributing sums gives

$$v_{i+1} = v_i + 2\sum_{j=1}^{i} \|x_j\|_2^2 - 4x_{i+1}^\top \sum_{j=1}^{i} x_j + 2i\|x_{i+1}\|_2^2,$$

and the v_i^c's satisfy a similar recursion rule. As such, one may iteratively compute the v_i's and v_i^c's before computing the $f(i)$'s and then minimizing. Overall, the following procedure finds the optimal (A_θ, B_θ) in $O((m + \log N)N)$ operations:

```
Algorithm 2: Spectral $k$-means clustering (for two clusters)
    Input: $m \times N$ matrix $\Phi = [x_1 \cdots x_N]$ of points to be clustered
    Output: Clusters $A \sqcup B = \{1, \ldots, N\}$
    Subtract centroid $\frac{1}{N}\sum_{i=1}^{N} x_i$ from each column of $\Phi$ to produce $\Phi_0$
    Compute leading eigenvector $y$ of $\Phi_0^\top \Phi_0$
    Find $\theta$ that minimizes the $k$-means objective of $(\{i : y_i < \theta\}, \{i : y_i \geq \theta\})$
    $(A, B) \leftarrow (\{i : y_i < \theta\}, \{i : y_i \geq \theta\})$
```

1. Sort the entries $y_1 \leq \cdots \leq y_N$ in $O(N \log N)$ operations.

2. Iteratively compute

$$s_1(i) := \sum_{j=1}^{i} x_j, \quad s_1^c(i) := \sum_{j=i+1}^{N} x_j, \quad s_2(i) := \sum_{j=1}^{i} \|x_j\|_2^2, \quad s_2^c(i) := \sum_{j=i+1}^{N} \|x_j\|_2^2$$

   for every $i \in \{1, \ldots, N-1\}$ in $O(mN)$ operations.

3. Compute $v_1 = 0$ and $v_{i+1} = v_i + 2s_2(i) - 4x_{i+1}^\top s_1(i) + 2i\|x_{i+1}\|_2^2$ for every $i \in \{1, \ldots, N-2\}$ in $O(mN)$ operations.

4. Compute $v_{N-1}^c = 0$ and $v_{i-1}^c = v_i^c + 2s_2^c(i) - 4x_i^\top s_1^c(i) + 2(N-i)\|x_i\|_2^2$ for every $i \in \{N-1, \ldots, 2\}$ in $O(mN)$ operations.

5. Compute $f(i) = v_i/i + v_i^c/(N-i)$ for every $i \in \{1, \ldots, N-1\}$ in $O(N)$ operations.

6. Find i that minimizes $f(i)$ and output $\{1, \ldots, i\}$ and $\{i+1, \ldots, N\}$ in $O(N)$ operations.

Note that in the special case where $m = 1$, the above method exactly solves the k-means problem when $k = 2$ in only $O(N \log N)$ operations. For comparison, [38] leverages more sophisticated dynamic programming for the $m = 1$ case, but k is arbitrary and the algorithm costs $O(kN^2)$ operations.

We call our proposed relax-and-round algorithm **spectral k-means clustering** (due to its similarity to spectral clustering); see Algorithm 2. As a spectral method, this algorithm enjoys quasilinear computational complexity; see Figure 3 for an illustration. In particular, when computing the leading eigenvector of $\Phi_0^\top \Phi_0$, each matrix–vector multiply in the power method costs only $O(mN)$ operations.

In summary, this chapter demonstrated the creation of a different dual certificate of the k-means relaxation used in [22]. Using this certificate, we created a fast k-means certifier completing a PCC algorithm for the k-means problem. We further demonstrated how the relax and round scheme can be used to create a 2-means solver. In the next chapter we analyze the performance of our k-means certificate.

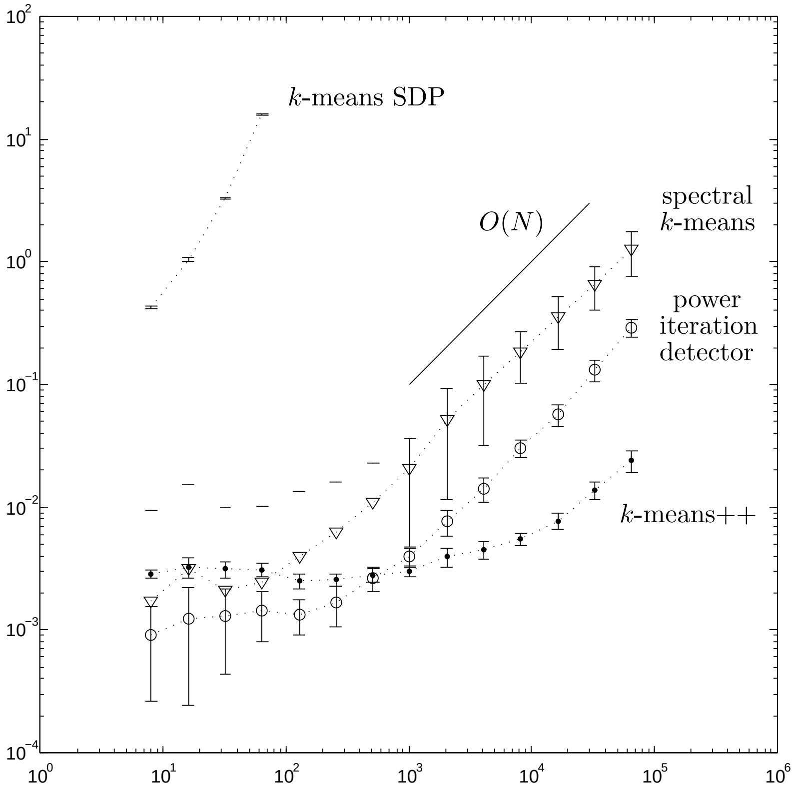


Figure 3. Simulated Computational Complexity of Relevant Algorithms Take two unit balls in $\mathbb{R}^6$ at distance 2.3 apart. For each $N \in \{2^3, 2^4, \ldots, 2^{16}\}$, we perform 300 trials of the following experiment: Draw $N/2$ points uniformly at random from each ball, and then compute four different functions: (a) MATLAB's built-in implementation of k-means++, (b) a CVX implementation [39] of the k-means SDP (2), (c) the power iteration detector (Algorithm 1) with A given by (11), and (d) spectral k-means clustering (Algorithm 2). For each trial, we recorded the runtime in seconds. Above, we plot the average runtime along with error bars for standard deviation. A line between average spectral k-means runtimes for $N \in \{2^3, \ldots, 2^9\}$ and its upper error bar is omitted for purposes of clarity as the standard deviation of runtimes in these instances were sufficiently large that the lower bar would have been negative. For the record, the power iteration detector failed to reject H_0 (10) in at most 3% of the trials with $N \leq 2^7$, but rejected H_0 in every trial otherwise; similarly, spectral k-means failed to recover the planted clusters in two of the trials with $N = 2^3$. Our implementation of the k-means SDP was too slow to perform trials with $N \geq 2^7$ in a reasonable amount of time, so we only recorded runtimes for $N \in \{2^3, 2^4, 2^5, 2^6\}$. As the plot illustrates, the other algorithms ran in quasilinear time, as expected.

III. k-means under the Stochastic Ball Model

In the previous chapter we developed a sufficient, deterministic condition on a set of points under which the SDP finds the k-means optimal solution. Now we desire to test the performance of this condition. That is, we wish to see how close this sufficient condition is to being equivalent to k-means optimality. As such, we require a set of points upon which we perform our test of planted cluster recovery. We formalize such a set of points called the Stochastic Ball Model and demonstrate our condition achieves state-of-the-art performance under this model. As an aside, we demonstrate how spectral k-means recovers planted clusters under this model with high probability when $k = 2$.

3.1 Previous Work

Recall the "relax and round" paradigm discussed in the previous chapter. Instead of solving a hard problem such as the k-means problem, we relax the feasibility region and provide an approximation to the optimal solution in the rounding step. If in the case where the convex relaxation happens to find a solution that is feasible in the original problem, then the relaxation is called tight or integral. As the k-means problem and other clustering algorithms have geometric interpretations, a natural question to consider is when is a convex relaxation of geometric clustering tight? This question seems to have first appeared in [40], which studies an LP relaxation of the k-medians objective (a problem that is similar to k-means). This thesis proved tightness of the relaxation provided the set of points P admits a partition into k clusters of equal size, and the separation distance between any two clusters is sufficiently large. Later on, [41] studied integrality of another LP relaxation to the k-medians objective. This paper introduced a distribution on the input P, which we refer to as the **stochastic**

Method	**Sufficient Condition**	**Optimal?**	**Reference**
Thresholding	$\Delta > 4$	Yes	(simple exercise)
k-medians LP	$\Delta \geq 4$	No	Theorem 2 in [40]
	$\Delta \geq 3.75$	No	Theorem 1 in [41]
	$\Delta > 2$	Yes	Theorem 1 in [22]
k-means LP	$\Delta > 4$	Yes	Theorem 9 in [22]
k-means SDP	$\Delta > 2\sqrt{2}(1 + 1/\sqrt{m})$	No	Theorem 3 in [22]
	$\Delta > 2 + k^2/m$	No	Theorem 11
Spectral k-means	$\Delta > \Delta^\star$, $k = 2$	Yes	Theorem 12

Table 2. Summary of Cluster Recovery Guarantees under the Stochastic Ball Model The second column reports sufficient separation between ball centers in order for the corresponding method to provably give exact recovery with high probability. Here, $\Delta^\star = \Delta^\star(\mathcal{D}, k)$ denotes the smallest value for which $\Delta > \Delta^\star$ implies that minimizing the k-means objective recovers planted clusters under the $(\mathcal{D}, \gamma, n)$-stochastic ball model with probability $1 - e^{-\Omega_{\mathcal{D},\gamma}(n)}$.

ball model:

Definition 9 ($(\mathcal{D}, \gamma, n)$-stochastic ball model)**.** *Let* $\{\gamma_a\}_{a=1}^k$ *be ball centers in* $\mathbb{R}^m$*. For each* a*, draw i.i.d. vectors* $\{r_{a,i}\}_{i=1}^n$ *from some rotation-invariant distribution* $\mathcal{D}$ *supported on the unit ball. The points from cluster* a *are then taken to be* $x_{a,i} := r_{a,i} + \gamma_a$*.*

Table 2 summarizes the state of the art for recovery guarantees under the stochastic ball model. In [41], it was shown that the LP relaxation of k-medians will, with high probability, recover clusters drawn from the stochastic ball model provided the smallest distance between ball centers is $\Delta \geq 3.75$. Note that exact recovery only makes sense for $\Delta > 2$ (i.e., when the balls are disjoint). Once $\Delta > 4$, any two points within a particular cluster are closer to each other than any two points from different clusters, and so in this regime, cluster recovery follows from thresholding. For the k-means problem, [22] provides a semidefinite relaxation and demonstrates exact recovery in the regime $\Delta > 2\sqrt{2}(1+1/\sqrt{m})$, where m is the dimension of the Euclidean space. That work also conjectures that the result holds for optimal separation $\Delta > 2$.

In this chapter we study the integrality of (2) under the Stochastic Ball Model and investigate planted cluster recovery of our spectral k-means to the Stochastic Ball Model.

3.2 Integrality of the Relaxation under the Stochastic Ball Model

First, we provide a new dual certificate for the k-means SDP from [22] and show that $\Delta > 2 + k^2/m$ suffices under the Stochastic Ball Model. We compare this bound to that in [22] and demonstrate that our bound is state-of-the-art when $k = o(m^{1/2})$.

We first note that our sufficient condition (9) is implied by

$$\|P_{\Lambda^\perp} M P_{\Lambda^\perp}\|_{2\to 2} + \|P_{\Lambda^\perp} B P_{\Lambda^\perp}\|_{2\to 2} \leq z.$$

By further analyzing the left-hand side above (see Appendix A), we arrive at the following corollary:

Corollary 10. *Take* $X := \sum_{t=1}^{k} \frac{1}{n_t} 1_t 1_t^\top$, *where* n_t *denotes the number of points in cluster* t. *Let* Ψ *denote the* $m \times N$ *matrix whose* (a, i)*th column is* $x_{a,i} - c_a$, *where*

$$c_a := \frac{1}{n_a} \sum_{i \in a} x_{a,i}$$

denotes the empirical center of cluster a. *Consider* M *defined by* (5), *pick* z *so as to satisfy equality in* (7), *and take* $\rho_{(a,b)}$ *defined by* (8). *Then* X *is a solution to the semidefinite relaxation* (2) *if*

$$2\|\Psi\|_{2\to 2}^2 + \sum_{a=1}^{k} \sum_{b=a+1}^{k} \frac{\|P_{1^\perp} M^{(a,b)} 1\|_2 \|P_{1^\perp} M^{(b,a)} 1\|_2}{\rho_{(a,b)}} \leq z. \tag{14}$$

Also in Appendix A, we leverage the stochastic ball model to bound each term in Corollary 10, and in doing so, we identify a regime in which the data points typically satisfy the sufficient condition given in Corollary 10:

Theorem 11. *The k-means semidefinite relaxation* (2) *recovers the planted clusters in the $(\mathcal{D}, \gamma, n)$-stochastic ball model with probability $1 - e^{-\Omega_{\mathcal{D},\gamma}(n)}$ provided $\Delta > 2 + k^2/m$.*

When $k = o(m^{1/2})$, Theorem 11 is near-optimal, and in this sense, it's a significant improvement over the sufficient condition

$$\Delta > 2\sqrt{2}\left(1 + \frac{1}{\sqrt{m}}\right) \tag{15}$$

given in [22]. However, there are regimes (e.g., $k = m$) for which (15) is much better, leaving open the question of what the optimal bound is. Conjecture 4 in [22] suggests that $\Delta > 2$ suffices for the k-means SDP to recover planted clusters under the stochastic ball model, but as we illustrate below, this conjecture is surprisingly false.

Consider the special case where $m = 1$, $\mathcal{D}$ is uniform on $\{\pm 1\}$, and $k = 2$. Centering the two balls on $\pm\Delta/2$, all of the points land in $\{\pm\Delta/2 \pm 1\}$. The k-means-optimal clustering will partition the real line into two semi-infinite intervals; there are three possible ways of clustering these points. Suppose exactly $N/4$ of the points land in each of the 4 positions. Then by symmetry, there are only two ways to cluster: either we select the planted clusters, or we make the left-most location its own cluster. Interestingly, a little algebra reveals that this second alternative is strictly better in the k-means sense provided $\Delta < 1 + \sqrt{3} \approx 2.7320$. Also, in this regime, as N gets large, the proportion of points in each position will be so close to $1/4$ (with high probability) that this clustering will beat the planted clusters.

Overall, when $m = 1$ and $k = 2$, then $\Delta \geq 1 + \sqrt{3}$ is necessary for minimizing the k-means objective to recover planted clusters for an arbitrary $\mathcal{D}$. As a relaxation,

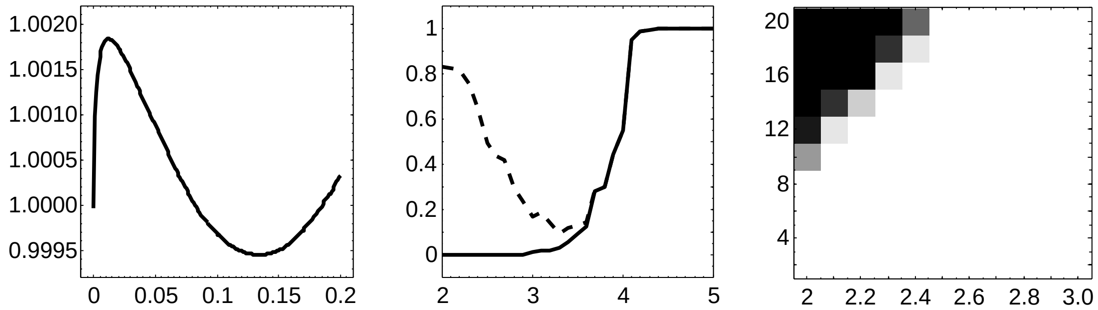


Figure 4. Exploring Bounds on Δ under the Stochastic Ball Model (left) Take two unit disks in $\mathbb{R}^2$ with centers on the x-axis at distance 2.08 apart. Let x_0 denote the smallest possible x-coordinate in the disk on the right. For each disk, draw $N/2 = 50,000$ points uniformly at random from the perimeter. Given θ, cluster the points according to whether the x-coordinate is smaller than $x_0 + \theta$. When $\theta = 0$, this clustering gives the planted clusters, and the k-means objective of the planted clustering (divided by N) is 1. We plot this normalized k-means objective for $\theta \in [0, 0.2]$. Since N is large, this curve is very close to its expected shape, and we see that there are clusters whose k-means value is smaller than that of the planted clustering. (center) Take two intervals of width 2 in $\mathbb{R}$, and let Δ denote the distance between the midpoints of these intervals. For each interval, draw 10 points at random from its endpoints, and then run the k-means SDP. For each $\Delta = 2 : 0.1 : 5$, after running $2,000$ trials of this experiment, we plot the proportion of trials for which the SDP was integral (dashed line) and the proportion of trials for which the SDP recovered the planted clusters (solid line). In this case, cluster recovery appears to exhibit a phase transition at $\Delta = 4$. (right) For each $\Delta = 2 : 0.1 : 3$ and $k = 2 : 2 : 20$, consider the unit balls in $\mathbb{R}^{20}$ centered at $\{\frac{\Delta}{\sqrt{2}} e_i\}_{i=1}^{k}$, where e_i denotes the ith identity basis element. Draw 100 points uniformly from each ball, and test if the resulting data points satisfy (9). After performing 10 trials of this experiment for each (Δ, k), we shade the corresponding pixel according to the proportion of successful trials (white means every trial satisfied (9)). This plot indicates that our certificate (9) and not the proof technique for Theorem 11 is to blame for Theorem 11's dependence on k.

the k-means SDP recovers planted clusters only if minimizing the k-means objective does so as well, and so it inherits this necessary condition, thereby disproving Conjecture 4 in [22]. Furthermore, as Figure 4 (left) illustrates, a similar counterexample is available in higher dimensions.

To study when the SDP recovers the clusters, let's continue with the case where $m = 1$ and $k = 2$. We know that minimizing k-means will recover the clusters with high probability provided $\Delta > 1 + \sqrt{3}$. However, Theorem 11 only guarantees that the SDP recovers the clusters when $\Delta > 6$; in fact, (15) is slightly better here, giving that $\Delta \geq 5.6569$ suffices. To shed light on the disparity, Figure 4 (center) illustrates the performance of the SDP. Observe that the SDP is often tight when Δ is close to

2, but it doesn't reliably recover the planted clusters until $\Delta > 4$. We suspect that $\Delta = 4$ is a phase transition for cluster recovery in this case.

Qualitatively, the biggest difference between Theorem 11 and (15) is the dependence on k that Theorem 11 exhibits. Figure 4 (right) illustrates that this comes from (9), meaning that one would need to use a completely different dual certificate in order to remove this dependence.

3.3 Miscellaneous Extensions

The previous chapter illustrated how to quickly test whether a proposed solution to the k-means problem is optimal. In particular, this section has shown the test will be successful with high probability if the data follow the stochastic ball model with $\Delta > 2 + k^2/m$. Lloyd's algorithm is fast, and we can certify optimality, but we have no guarantees prior to certification that the output of such a solver should be optimal. Here we find a fast k-means solver with guarantees under the Stochastic Ball Model for $k = 2$.

In doing so, we maintain the philosophy that our algorithm should not "see" the stochastic ball model. Indeed, we view the stochastic ball model as a method of evaluating clustering algorithms rather than a realistic data model. For example, Lloyd's algorithm can be viewed as an alternating minimization of the lifted objective function:

$$f(A_1, \ldots, A_k, c_1, \ldots, c_k) := \sum_{t=1}^{k} \sum_{x \in A_t} \|x - c_t\|^2, \qquad A_1 \sqcup \cdots \sqcup A_k = P,\ c_1, \ldots, c_k \in \mathbb{R}^m,$$

and since this function is minimized at the k-means optimizer (regardless of how the data is distributed), such an algorithm is acceptable. On the other hand, one might consider matching the stochastic ball model to the data by maximizing the following function:

$$g(c_1,\ldots,c_k) := \sum_{x\in P}\sum_{t=1}^{k} p_{\mathcal{D}}(x-c_t), \qquad c_1,\ldots,c_k\in\mathbb{R}^m,$$

where $p_{\mathcal{D}}(\cdot)$ denotes the density function of $\mathcal{D}$, which is supported on the unit ball centered at the origin. One could certainly devise a fast greedy method such as matching pursuit [42] to optimize this objective function (especially if $p_{\mathcal{D}}$ is smooth), but doing so violates our philosophy.

Recall in the previous chapter we developed a fast k-means solver called spectral k-means which was designed without a particular distribution of points. As such, we evaluate the performance of spectral k-means under the stochastic ball model. The following results summarize such an evaluation of spectral k-means.

Theorem 12. *Let $\Delta^\star = \Delta^\star(\mathcal{D},k)$ denote the smallest value for which $\Delta > \Delta^\star$ implies that minimizing the k-means objective recovers planted clusters under the $(\mathcal{D},\gamma,n)$-stochastic ball model with probability $1-e^{-\Omega_{\mathcal{D},\gamma}(n)}$. When $k=2$, spectral k-means clustering (Algorithm 2) recovers planted clusters under the stochastic ball model with probability $1-e^{-\Omega_{\mathcal{D},\gamma}(n)}$ provided $\Delta > \Delta^\star$.*

See Appendix A for the proof. The main idea is that the leading eigenvector of $\Phi_0\Phi_0^\top$ is biased towards the difference between the ball centers, and as the following lemma establishes, this bias encourages spectral k-means clustering to separate the planted clusters:

Lemma 13. *Take two clusters contained in unit balls centered at γ and $-\gamma$ with $\|\gamma\|_2 > 1$. If minimizing the k-means objective recovers these clusters, then spectral k-means clustering (Algorithm 2) also recovers them, provided the leading eigenvector z of $\Phi_0\Phi_0^\top$ satisfies $|\gamma^\top z| > \|z\|_2$.*

Proof. Write $\Phi_0 = \Phi - \mu 1^\top$, put $\theta := -\mu^\top z$, and observe that $y = \Phi_0^\top z$ is a leading eigenvector of $\Phi_0^\top \Phi_0$. Then

$$y_i = (x_i - \mu)^\top z = x_i^\top z + \theta \tag{16}$$

for every i. Next, if $|\gamma^\top z| > \|z\|_2$, then a simple trigonometric argument gives that the balls (and therefore the planted clusters) are separated by the hyperplane orthogonal to z. Combined with (16), we then have that the clusters can be identified according to whether $y_i < \theta$ or $y_i > \theta$. It therefore suffices to minimize the k-means objective subject to partitions of this form (for arbitrary thresholds θ), as so spectral k-means clustering succeeds. □

In this chapter we demonstrated our certificate often achieves state-of-the-art performance under the Stochastic Ball Model. In the next chapter we examine a model more closely related to the Air Force health care problem.

IV. Bipartite Stochastic Block Model

In the context of the Air Force health care problem, there is an intrinsic bipartite structure of the data which k-means ignores. That is, when inspecting a database of medical histories, patients are easily distinguishable from medical conditions. As such, we seek a model that exploits this structure. With this model in hand, we seek guarantees and a PCC algorithm to find and certify clusters in the model. In so doing, we realize we may reuse concepts and algorithms in the previous two chapters with noticeable effect.

4.1 Previous Work

A well studied and commonly used generative model for social and biological networks is called the Stochastic Block Model (see [26, 27, 28, 29]). In this model, clusters do not consist of balls of points but are well connected subgraphs of a random graph as shown in Figure 6b. There have been several variants of this model in the literature to include a version we created independently, all of which have been called the Bipartite Stochastic Block Model [30, 32]. Denoting $[i] := \{j \in \mathbb{N} : j \leq i\}$, we provide a definition which encompasses all of these slightly differing models:

Definition 14 (Bipartite Stochastic Block Model (BiSBM))**.** *Let U, V be vertex sets partitioned as $\{U_i\}_{i=1}^{k}$, $\{V_j\}_{j=1}^{l}$, respectively. For every $i \in [k], j \in [l]$, define a probability distribution f_{ij}. Create a graph G where the number of edges (u, v), with $u \in U_i$ and $v \in V_j$ are distributed according to f_{ij}. Edges do not exist between vertices in U nor do they exist between vertices in V.*

In [30], the authors proposed two BiSBMs where the probability distributions were Poisson. The first model has no control over the degree of a vertex, while the second model controls the expected degree of a vertex. The authors further devise an MLE based algorithm recovered planted clusters in synthetic data and detected clusters in several real world datasets given only the graph G. In particular, authors used three real world data sets. The first is the Southern Women Dataset, a common benchmark for bipartite community detection algorithms collected by ethnographers to examine the roles of race and class in dictating social interactions. The second is a dataset on the malaria parasite *P. falciparum* which includes information on how the parasite's genes recombine to create a protein camouflage to evade the human immune system. The third dataset is the Internet Movie Database (IMDb) from which the authors extracted a bipartite network of actors and movies. The authors demonstrate the utility of their novel algorithms which fits bipartite data to a BiSBM by showing how these algorithms outperformed a Stochastic Block Model cluster recovery algorithm presented in [43] on both the synthetic and real word data sets that followed a bipartite structure.

Related work in [31, 32] use a different BiSBM where the probability distributions are Bernoulli random variables where the expected value of this distribution tends to zero as the number of vertices in U and V tend to infinity. They provide multiple spectral algorithms which recover planted clusters with high probability.

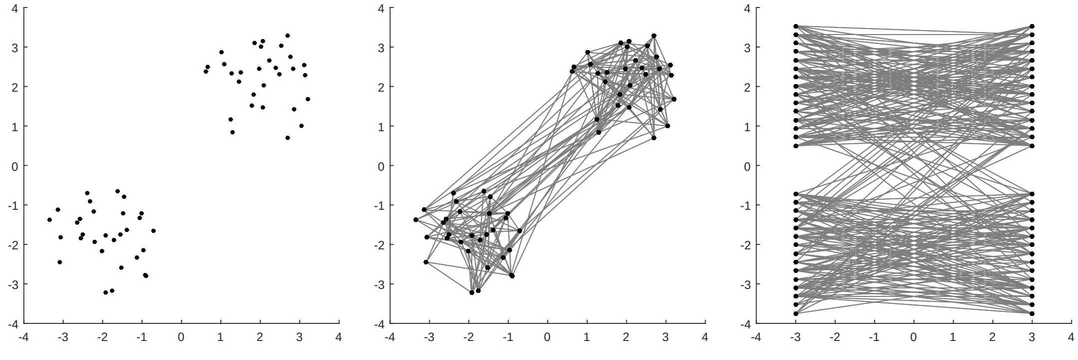
We wish to apply the relax and round paradigm to a BiSBM. In this context, the objective function to be used for the BiSBM will be the likelihood of a proposed partition being the ground truth partition. Notice this objective function is different from the k-means objective function which necessitates a completely new relaxation and dual certificate. Although we could apply the k-means certificate to a clustering under the BiSBM if we somehow interpret the clusters in Euclidean space, we instead choose to consider a different objective function motivated by related work in the Stochastic Block Model. The authors of [26, 44] used a "relax and round" scheme to create a dual certificate that certifies that a user has found the planted clusters given only information of the graph G for the Stochastic Block Model. Using the proof technique in [26, 44] for the Stochastic Block Model, we create a similar dual certificate for our particular form of the BiSBM. We consider vertices $U = \{U_i\}_{i=1}^2, V = \{V_i\}_{i=1}^2$ and let the number of edges (u, v) in our BiSBM be distributed i.i.d. as follows:

$$|(u,v)| \sim \begin{cases} \text{Bernoulli } p, & \text{if } (u,v) \in (U_i, V_i) \\ \text{Bernoulli } q, & \text{else} \end{cases} \tag{17}$$

for fixed p, q. We give a visualization of this BiSBM in Figure 5. Additionally, in this chapter we present guarantees for a thresholding algorithm that recovers planted clusters in one partite in our BiSBM. As such, we present a methodology for quickly performing a one-partite thresholding in the hopes it may be used effectively in creating patient partitions for the Air Force health care problem. We numerically demonstrate that with a simple addition to this thresholding algorithm, we can recover planted clusters in both partites in a large regime of p, q. We conclude with work on creating a PCC algorithm for a particular form of this BiSBM and, as an extension, its cousin the Stochastic Block Model.

4.2 Maximum Likelihood Based Certification

In this section we develop an SDP relaxation of an MLE based partition recovery method for a particular BiSBM given only the graph G. We aim to find a dual certificate and a regime of our probability parameters p, q for which we recover a planted partition. As a special note, the techniques used in this section closely follow Bandeira's work with the Stochastic Block Model in [44].



(a) Stochastic Ball Model **(b)** Stochastic Block Model **(c)** BiSBM

Figure 5. A Comparison of Models The following is performed in MATLAB 2015a. In (a) we generate two Stochastic Balls of radius 1 with centers (-2,-2) and (2,2) using a uniform distribution each having thirty points. In (b) we now consider the points generated in (a) to be vertices in a graph. For each Stochastic Ball, pick 15 vertices and draw edges between them according to a Bernoulli distribution with mean $p = 0.4$. Repeat the same procedure for the other 15 vertices in the Stochastic Ball. Now draw edges between the vertices across the Stochastic Balls according to a Bernoulli distribution with mean $q = 0.1$. Due to the bipartite structure of the graph, we move the vertices to organize them in such a way as to exhibit this structure. Note the Stochastic Block Model need not have its vertices possess a representation in Euclidean space. We merely use such a representation to create an easy to understand visualization.

SDP Formulation and Relaxation.

Assume our BiSBM has $|U| = |V| = n$ with $n/2$ vertices in each partition. Since we could merely relabel the partitions without loss of generality, we enforce $p > q$. Denoting $\mathcal{E}(G)$ to be the edge set of the graph G, let $\mathcal{A} \in \mathbb{R}^{2n\times 2n}$ be the adjacency matrix of G where

$$\mathcal{A}_{ij} = \begin{cases} 1 & \text{if } (i,j) \in \mathcal{E}(G) \\ 0 & \text{else.} \end{cases} \tag{18}$$

For the sake of notation, we group the columns and rows of $\mathcal{A}$ so that the vertices in U and V correspond to the first and last n rows and columns, respectively. Furthermore, we also group the rows and columns to group the partitions of U and V. We illustrate the result of the shuffling of $\mathcal{A}$:

$$\begin{array}{c|cccc|} & U_1 & U_2 & V_1 & V_2 \\ U_1 & 0 & 0 & \text{Bernoulli } p & \text{Bernoulli } q \\ U_2 & 0 & 0 & \text{Bernoulli } q & \text{Bernoulli } p \\ V_1 & \text{Bernoulli } p & \text{Bernoulli } q & 0 & 0 \\ V_2 & \text{Bernoulli } q & \text{Bernoulli } p & 0 & 0 \end{array}.$$

Let $g \in \{\pm 1\}^{2n}$ such that

$$g_i = \begin{cases} 1 & \text{if vertex } i \in U_1 \cup V_1 \\ -1 & \text{if vertex } i \in U_2 \cup V_2. \end{cases} \tag{19}$$

We note the information in $\mathcal{A}$ can be treated as observed data, g is ground truth while $\widetilde{g} \in \{\pm 1\}^{2n}$ represents our guess for the partition of G. Since there are n vertices in both U and V, we should choose $\widetilde{g}$ such that $\widetilde{g}^\top 1 = 0$.

We will now proceed with creating an SDP that optimizes an MLE as a function of $\mathcal{A}$ and $\widetilde{g}$. Similar to the notation used in Chapter II, we use $1_U, 1_V \in \mathbb{Z}_2^{2n}$ to denote an indicator function of the vertices in U and V, respectively. We further denote $\Theta := 1_U 1_U^\top + 1_V 1_V^\top$ and $\mathbf{B} := 2A - 11^\top + \Theta$. Observe that

$$\mathbf{B}_{ij} = \begin{cases} 0 & \text{if } i, j \in U \text{ or } i, j \in V \\ 1 & \text{if } (i, j) \in \mathcal{E}(G) \\ -1 & \text{otherwise.} \end{cases} \tag{20}$$

Let $X = \widetilde{g}\widetilde{g}^\top$ and $\Omega = 1_U 1_V^\top + 1_V 1_U^\top$. We wish to find a partition that maximizes $\Pr(\mathbf{B}|X)$ which is calculated as

$$\Pr(\mathbf{B}|X) = p^{\langle \frac{\Omega+\mathbf{B}}{2}, \frac{\Omega+X}{2} \rangle/2} (1-p)^{\langle \frac{\Omega-\mathbf{B}}{2}, \frac{\Omega+X}{2} \rangle/2} q^{\langle \frac{\Omega+\mathbf{B}}{2}, \frac{\Omega-X}{2} \rangle/2} (1-q)^{\langle \frac{\Omega-\mathbf{B}}{2}, \frac{\Omega-X}{2} \rangle/2}. \tag{21}$$

We reinterpret this probability as the likelihood $\Pr(\mathbf{B}|X)$ and find the MLE which is the partition maximizing (21). The corresponding log likelihood function is

$$\begin{aligned}\mathscr{L}(X|\mathbf{B}) &= \left\langle \frac{\Omega+\mathbf{B}}{2}, \frac{\Omega+X}{2} \right\rangle \frac{\log(p)}{2} + \left\langle \frac{\Omega-\mathbf{B}}{2}, \frac{\Omega+X}{2} \right\rangle \log\frac{(1-p)}{2} + \\ &\quad \left\langle \frac{\Omega+\mathbf{B}}{2}, \frac{\Omega-X}{2} \right\rangle \frac{\log(q)}{2} + \left\langle \frac{\Omega-\mathbf{B}}{2}, \frac{\Omega-X}{2} \right\rangle \log\frac{(1-q)}{2} \\ &= \frac{1}{8}\log\left(\frac{p(1-q)}{q(1-p)}\right)\langle \mathbf{B}, X\rangle + O(1)\end{aligned}$$

which can be seen by expanding the inner products and identifying constants. In order to maximize $\mathscr{L}(X|\mathbf{B})$, it is sufficient to maximize $\langle \mathbf{B}, X\rangle$ as we have enforced $p > q$. We write this maximization problem as

$$\begin{aligned} &\text{maximize} && \langle \widetilde{g}\widetilde{g}^\top, \mathbf{B}\rangle \\ &\text{subject to} && \widetilde{g} \in \{\pm 1\}^{2n}. \end{aligned} \tag{22}$$

Since $X = \widetilde{g}\widetilde{g}^\top$, we can equivalently write

$$\begin{aligned} &\text{maximize} && \langle X, \mathbf{B}\rangle \\ &\text{subject to} && X_{ii} = 1 \\ &&& X \succeq 0 \\ &&& \mathrm{Rank}(X) = 1. \end{aligned} \tag{23}$$

Since the $\mathrm{Rank}(X) = 1$ is a non-convex constraint, we wish to relax (23) to

$$\begin{aligned} &\text{maximize} && \langle X, \mathbf{B}\rangle \\ &\text{subject to} && X_{ii} = 1 \\ &&& X \succeq 0 \end{aligned} \tag{24}$$

Denoting $\mathcal{W}$ to be the dual variable to X, the dual program of (24) is

$$\begin{aligned} &\text{minimize} && \operatorname{Tr}(\mathcal{W}) \\ &\text{subject to} && \mathcal{W} - \mathbf{B} \succeq 0 \\ &&& \mathcal{W} \text{ is diagonal.} \end{aligned} \tag{25}$$

Creating a Dual Certificate.

We now proceed to create a dual certificate for (24). The goal is to find a solution $\mathcal{W}$ to (25) and for it to certify the unique integral solution $X = gg^\top$ in the primal (24).

Lemma 15. *Denote $\mathcal{G}_-$ as the degree matrix of the subgraph generated by the edges occurring with probability p in G and $\mathcal{G}_-$ as the degree matrix of the subgraph generated by the edges occurring with probability q in G. The dual certificate to the semidefinite relaxation* (24) *is*

$$\mathcal{W} = 2(\mathcal{G}_+ - \mathcal{G}_-) \tag{26}$$

which certifies $X = gg^\top$ as a solution to the semidefinite relaxation.

Proof. The dual certificate must be dual feasible so we require $\mathcal{W}$ to be diagonal and $\mathcal{W} - \mathbf{B} \succeq 0$. By complementary slackness of (24) and its dual (25), we desire $gg^\top$ to be complementary to $\mathcal{W} - \mathbf{B}$. That is, $\operatorname{Tr}((\mathcal{W} - \mathbf{B})gg^\top) = 0$. By cycling the trace, we obtain

$$\operatorname{Tr}\left((\mathcal{W} - \mathbf{B})gg^\top\right) = \operatorname{Tr}\left(g^\top(\mathcal{W} - \mathbf{B})g\right) = g^\top(\mathcal{W} - \mathbf{B})g = 0.$$

Since $\mathcal{W} - \mathbf{B} \succeq 0$, we obtain $(\mathcal{W} - \mathbf{B})g = 0$. As $\mathcal{W}g = \mathbf{B}g$, we may write

$$\mathcal{W}g = \mathbf{B}g = (2\mathcal{A} - 11^\top + \Theta)g = (2\mathcal{A} - 11^\top + 1_U 1_U^\top + 1_V 1_V^\top)g = 2\mathcal{A}g$$

where the last step can be seen by observing g is orthogonal to $1, 1_U$, and 1_V. Recalling $\mathcal{W}$ is diagonal and $g_i \in \{\pm 1\}$ denotes community membership, we observe

$$\mathcal{W}_{ii} = \frac{1}{g_i}(\mathcal{W}g)_i = \frac{2}{g_i}(\mathcal{A}g)_i = 2\sum_j \frac{A_{ij}g_j}{g_i} = 2\Big((\mathcal{G}_+)_{ii} - (\mathcal{G}_-)_{ii}\Big). \qquad \square$$

Notice this certifies $X = gg^\top$, but does not guarantee its uniqueness. If, however, the nullspace of $\mathcal{W} - \mathbf{B}$ is $\text{span}(g)$, then complementary slackness guarantees uniqueness. That is, uniqueness is guaranteed if the second smallest eigenvalue of $\mathcal{W} - \mathbf{B}$ is greater than zero. Using this fact and Lemma 15, we have the following main result whose proof is in Appendix A

Theorem 16. *If $p > 1/2$ and $q < 1/2$, then the semidefinite relaxation* (24) *recovers the true partition with probability* $1 - ne^{-\Omega_{p,q}(n)}$.

Notice this theorem is a statement regarding which BiSBMs or equivalently what choices of p, q will allow a guarantee of the relaxation achieving an optimal partition. In particular, the theorem guarantees the relaxation will give the optimal partition by guaranteeing integrality in a certain regime of p, q. See Figure 6 for a simulation which demonstrates the regime of p, q for successful certification of planted clusters.

At this stage, the only way to achieve this optimal partition is through an SDP solver. These algorithms, however, are slow in practice and leave much to be desired in runtime. As such, we pursue a fast algorithm that recovers planted partitions with high probability with the hopes that the partitions will be near optimal. Additionally, we create a thresholding algorithm and apply the power iteration detector to the dual certificate (26) to complete the framework for a PCC algorithm.

4.3 Thresholding

The goals of this section are two-fold: find guarantees that a thresholding algorithm will recover planted clusters in one partite of a generalized BiSBM and develop a fast thresholding algorithm to execute such a task. Furthermore, in this section we consider a slightly more general model where $|U|$ does not necessarily equal $|V|$.

The Case of Equal Size Clusters.

In this case our BiSBM has $|U| = 2n$ and $|V| = 2m$ with $|U_1| = |U_2| = n$. Taking the adjacency matrix of G with the columns and rows shuffled as in Section 4.2, we notice the off-diagonal $m \times n$ blocks are transposes of each other and the diagonal blocks are zeros. For purposes of notation, we denote the biadjacency matrix $\mathcal{Q}$ to be the block in $\mathcal{A}$ with rows corresponding to vertices in V and columns corresponding to vertices in U. We can visualize $\mathcal{Q}$ in block form:

$$\mathcal{Q} = \begin{bmatrix} \mathcal{Q}_{11} & \mathcal{Q}_{12} \\ \mathcal{Q}_{21} & \mathcal{Q}_{22} \end{bmatrix} \tag{27}$$

where each block is an $m \times n$ matrix with each entry in block $\mathcal{Q}_{rs}$ is distributed Bernoulli with probability p if $r = s$ and is distributed Bernoulli with probability q if $r \neq s$. We wish to perform a thresholding algorithm on the columns of $\mathcal{Q}$ where U_r for $r \in [2]$ form the clusters. Let x_i, x_j be columns in the same cluster. Denote $b_l(\cdot), b'_l(\cdot)$ as Bernoulli random variables with index l and with the input as the mean.

The squared Euclidean distance between columns in the same cluster is given by

$$\begin{aligned}
||x_i - x_j||_2^2 &= \sum_{l=1}^{m} \big(b_l(p) - b_l'(p)\big)^2 + \sum_{l=1}^{m} \big(b_l(q) - b_l'(q)\big)^2 \\
&= \sum_{l=1}^{m} b_l\big(2p(1-p)\big) + \sum_{l'=1}^{m} b_{l'}\big(2q(1-q)\big) \\
&= \text{Binomial}\big(m, 2p(1-p)\big) + \text{Binomial}\big(m, 2q(1-q)\big). \qquad (28)
\end{aligned}$$

Now let x_i, x_j be columns in different clusters. The squared Euclidean distance between columns in this case is given by

$$||x_i - x_j||_2^2 = \sum_{l=1}^{2m} \big(b_l(p) - b_l'(p)\big)^2 = \sum_{l=1}^{2m} b_l(p+q-2pq) = \text{Binomial}(2m, p+q-2pq). \qquad (29)$$

Let

$$\mathcal{S} := \max_{\substack{i,j\in U_r \\ r\in[2] \\ i\neq j}} ||x_i - x_j||_2^2 \quad \text{and} \quad \mathcal{T} := \min_{\substack{i\in U_1 \\ j\in U_2}} ||x_i - x_j||_2^2. \qquad (30)$$

Since it is possible to identify the the partition of U if $\mathcal{S} < \mathcal{T}$, we wish to show that $\mathcal{S} < \mathcal{T}$ with high probability. We proceed by bounding $\Pr(\mathcal{S} \geq \mathcal{T})$ from above and showing the bound approaches zeros as m becomes large. Define the random variables S, T by

$$S \sim \text{Binomial}\big(m, 2p(1-p)\big) + \text{Binomial}\big(m, 2q(1-q)\big) \qquad (31)$$

$$T \sim \text{Binomial}\big(2m, p+q-2pq)\big) \qquad (32)$$

so that S, T have the same distribution as any intra-cluster and inter-cluster distance, respectively. Further denotes $\mu_S := \mathbb{E}[S]$ and $\mu_T := \mathbb{E}[T]$. The proof for the following theorem is in Appendix A.

Theorem 17. *Let* $p, q \in [0,1]$ *where* $p \neq q$. *For every* $\epsilon > 0$ *and* $n \in \mathbb{N}$, *there exists* $m \in \mathbb{N}$ *such that* $\Pr(\mathcal{S} < \mathcal{T}) > 1 - \epsilon$. *That is,* $\Pr(\mathcal{S} < \mathcal{T}) \to 1$ *as* $m \to \infty$.

That is, if there are sufficiently many vertices in one partite, thresholding the other partite under this particular BiSBM is possible with high probability. The next subsection achieves a similar result in a slightly more generalized BiSBM.

The Case of Unequal Size Clusters.

We now modify the graph G such that U_r is a set of n_r vertices for $r \in [2]$. The dimension of $\mathcal{Q}$ is now $2m \times (n_1 + n_2)$. We can again visualize $\mathcal{A}$ in block form just as in (27) where block $\mathcal{Q}_{rs}$ is an $m \times n_r$ matrix where where every entry in block $\mathcal{Q}_{rs}$ is distributed Bernoulli with mean p if $r = s$ and is distributed Bernoulli with mean q if $r \neq s$. We again wish to perform a thresholding algorithm on the columns of $\mathcal{Q}$ where U_r for $r \in [2]$ form the clusters.

Notice the distribution of the inter-cluster and intra-cluster distances in (28) and (29) are not dependent on $|U|$. As such the distribution of these distances remain unchanged. We define $\mathcal{S}, \mathcal{T}$ just as before in (30) and S, T as in (31) and (32). We proceed with the main result of this section which is proven in Appendix A.

Theorem 18. *Let* $p, q \in [0, 1]$ *where* $p \neq q$. *For every* $\epsilon > 0$ *and* $n_1, n_2 \in \mathbb{N}$, *there exists* $m \in \mathbb{N}$ *such that* $\Pr(\mathcal{S} < \mathcal{T}) > 1 - \epsilon$. *That is,* $\Pr(\mathcal{S} < \mathcal{T}) \to 1$ *as* $m \to \infty$.

A corollary falls out from the choice of m in the proof of Theorem 18 which presents a perspective on the relationship between m and n_1, n_2.

Corollary 19. *Let* $p, q \in [0, 1]$ *where* $p \neq q$. *If* $n_{max} := \max(n_1, n_2)$, *then* $\Pr(\mathcal{S} < \mathcal{T}) \to 1$ *as* $m \to \infty$ *where* $m = \Omega\big(\log(n_{max})\big)$.

Pursuing a Fast Thresholding Algorithm.

Now we aim to devise a fast algorithm which recovers planted clusters in one partite where thresholding is possible. Certainly, we can threshold if we calculate a distance matrix, but doing so has cost $O(n^2)$ where $|U_r| = n/2$ for $r \in [2]$. It often happens that minimizing the k-means objective will suffice to perform thresholding. More precisely, for sufficiently balanced and separated clusters, the k-means optimal clustering is the threshold clustering. As such, we can use any k-means heuristic to threshold.

Although we may be sufficiently satisfied in partitioning U under the BiSBM for purposes of finding patient partitions, it would be convenient to at least develop some reasonable heuristic to gain insight into the partitioning of V. The rows of $\mathcal{Q}$ under V_1 and V_2 have two distinctly different expected values: $p1_{U_1} + q1_{U_2}$ for rows in V_1 and $q1_{U_1} + p1_{U_2}$ for rows in V_2. As such, we can make an educated guess in which partition a row of $\mathcal{Q}$ lies given the number of ones in columns of U_1 versus the number of ones in the columns of U_2. With this motivation, we propose two options for a heuristic. The first is to re-apply k-means to the rows of $\mathcal{Q}$ as they should have two distinct centroids. This method, however, may not lead to equi-sized partitions of V. The second method is to sort the rows where each row is assigned the quantity

$$\frac{e_i \mathcal{Q} 1_{U_1}}{n_1} - \frac{e_i \mathcal{Q} 1_{U_2}}{n_2}. \tag{33}$$

After this sorting, assign the first m rows to V_1 and the second set of m rows to V_2. We simulate this k-means inspired thresholding (with sorting) and measure the performance of planted cluster recovery and present the results in Figure 6a. Notice this heuristic for thresholding works reasonably well for large regime of p, q.

In the next section, we cover how to take an output of one of these fast thresholding algorithms and certify optimality.

4.4 A PCC Algorithm for the BiSBM

Now that we have a quasilinear time algorithm to find a threshold clustering, we attempt to find a fast method to certify the clustering with the certificate from Lemma 15. Since we intend to use Lemma 15, the BiSBM used in this section is the same as the one in Section 4.2. We attempt to do so by use of the power iteration detector. First, the number of nonzero entries in $\mathcal{A}$ has an expected value of $O(n^2)$. As such, the complexity of the function $x \mapsto \mathbf{B}x$ and the computation of the diagonal of $\mathcal{W}$ are both $O(n^2)$. As demonstrated in the proof for Lemma 15 and Theorem 16, to certify $\widetilde{g}\widetilde{g}^\top$ as the unique optimal solution to the SDP relaxation, the dual certificate required the following:

1. $(\mathcal{W} - \mathbf{B})\widetilde{g} = 0$ by complementary slackness and dual feasibility of $\mathcal{W}$,
2. $\lambda_2(\mathcal{W} - \mathbf{B}) > 0$ to demonstrate uniqueness

where $\lambda_2(\cdot)$ denotes the second smallest eigenvalue of its input. We can check $(\mathcal{W} - \mathbf{B})\widetilde{g} = 0$ in $O(n)$. As the thresholding algorithm may produce slightly unbalanced clusters, it is necessary to check $\widetilde{g}^\top 1_{U_r} = \widetilde{g}^\top 1_{V_r} = n/2$ for $r \in [2]$ which can also be done in $O(n)$. Now we require a method for finding the second smallest eigenvalue. We can upper bound the largest eigenvalue in the following manner:

$$\begin{aligned}
\|\mathcal{W} - \mathbf{B}\|_{2\to 2} &\leq \|\mathcal{W}\|_{2\to 2} + \|2\mathcal{A} - 11^\top + \Theta\|_{2\to 2} \\
&\leq \|\mathcal{W}\|_{2\to 2} + \|2\mathcal{A} - 11^\top\|_{2\to 2} + \|\Theta\|_{2\to 2} \\
&\leq \|\mathcal{W}\|_{2\to 2} + \|2\mathcal{A} - 11^\top\|_{F} + \|\Theta\|_{2\to 2} \\
&= \max_i |\mathcal{W}_{ii}| + 3n.
\end{aligned}$$

Denote $\lambda := \max_i |\mathcal{W}_{ii}| + 3n$ which can be calculated in $O(n)$. Assuming $(\mathcal{W} - \mathbf{B})\widetilde{g} = 0$, then we know $\lambda_2(\mathcal{W} - \mathbf{B}) > 0$ if and only if $\widetilde{g}$ spans the leading eigenspace of

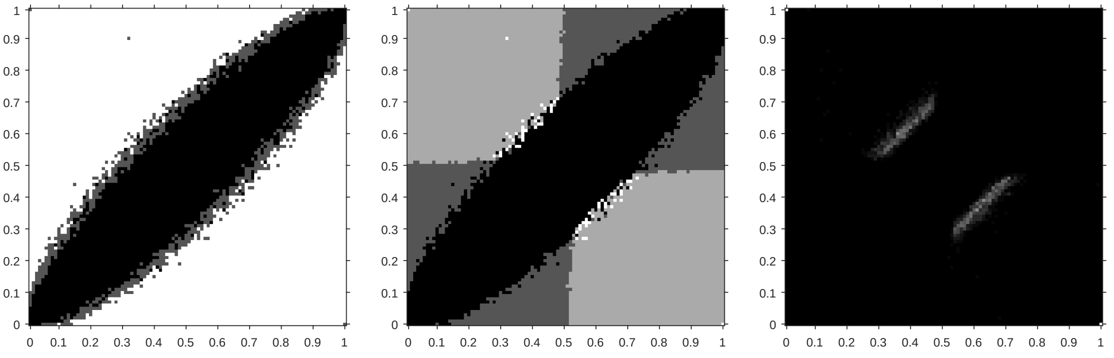
$\lambda I - \mathcal{W} + \mathbf{B}$. As such, we can apply the power iteration detector to test if $\widetilde{g}$ is the leading eigenvector of $\lambda I - \mathcal{W} + \mathbf{B}$. Unfortunately, since $x \mapsto \mathbf{B}x$ takes $O(n^2)$, each iteration of the power iteration detector also costs $O(n^2)$.

Notice the complexity of each iteration of the power iteration detector applied to this BiSBM is a function of the sparsity of the adjacency matrix. In the next section, we demonstrate how the power iteration detector applied to the more general Stochastic Block Model can certify a maximum likelihood partition in quasilinear time.

4.5 Miscellaneous Extensions

The original problem for which a PCC algorithm was developed was community recovery under the stochastic block model [25] (whose notation we borrow for use in this section). For this random graph, there are two communities of vertices, each of size $n/2$, and edges are drawn independently at random with probability p if the pair of vertices belong to the same community, and with probability $q < p$ if they come from different communities. Given the random edges, the maximum likelihood estimator for the communities is given by the vertex partition of two sets of size $n/2$ with the minimum cut. Given a partition of the vertices, let X denote the corresponding $n \times n$ matrix of ± 1s such that $X_{ij} = 1$ precisely when i and j belong to the same community. Given the adjacency matrix A of the random graph, one may express the cut of a partition X in terms of $\operatorname{Tr}(AX)$. Furthermore, X satisfies the convex constraints $X_{ii} = 1$ and $X \succeq 0$, and so one may relax to these constraints to obtain a semidefinite program and hope that the relaxation is typically tight over a large region of (p, q). Amazingly, this relaxation is typically tight precisely over the region of (p, q) for which community recovery is information-theoretically possible [26].

Given A, put $B := 2A - 11^\top + I$, and given a vector $x \in \mathbb{R}^n$, define the corre-



(a) **(b)** **(c)**

Figure 6. Simulated BiSBM Planted Cluster Recovery and Certification In all figures, the axes denote values of p, q. (left) Let $n = 200$ denote the size of U, V, respectively. For values of $p, q = 0 : 0.01 : 1$, randomly generate a biadjacency matrix $\mathcal{Q}$ according to the BiSBM in Section 4.2. Run k-means++ in Matlab 2015a with $k = 2$ on the columns of $\mathcal{Q}$ and sort the rows based on the difference of the sum of entries in each 2-means clustering of the columns. Check if it returned the planted partition. Test if the k-means clustering is k-means optimal with Theorem 3 using matrix norms (instead of the power iteration detector). Determine if the dual certificate in Lemma 15 certifies the integral solution. The shade of a pixel is determined by the k-means++ output being: (a) the planted partition and certified as k-means optimal–white; (b) the planted partition– gray; (c) not the planted partition–black. (center) Using the same data from the figure on the left, the color of a pixel is now determined by if the k-means++ output was: (a) MLE optimal and not k-means optimal–white (b) Both MLE and k-means optimal–light gray (c) Only k-means optimal–dark gray (d) Not optimal–black. (right) Let $n = 200$. For values of $p, q = 0 : 0.01 : 1$, run 30 trials of the above experiment. A pixel is white if the partition was certified as MLE optimal every time and was not certified as being k-means optimal. Conversely, a pixel is black if this event never happened. This shows the region where the MLE certificate outperforms the k-means certificate. (Note: In the figure in the left, there were 2 false positives for the MLE certifier and 17 false positives occurred in the trials for the figure on the right. We attribute these errors either to some numerical instability when $p \approx q$ or due to the fact that our guarantee only holds with high probability as the number of vertices grows large. It may be the case that in this regime of p, q where we observe the false positives that the number of vertices is not sufficiently large. Unfortunately, these false positives seem to occur in the same regime as when the MLE certificate outperforms the k-means certificate.)

sponding $n \times n$ diagonal matrix D_x by $(D_x)_{ii} := x_i \sum_{j=1}^{n} B_{ij} x_j$. In [25], Bandeira observes that, given a partition matrix X by some means (such as the fast algorithm provided in [45]), then $X = xx^\top$ is SDP-optimal if both $x^\top 1 = 0$ and the second smallest eigenvalue of $D_x - B$ is strictly positive, meaning the partition gives the maximum likelihood estimator for the communities. However, as Bandeira notes, the computational bottleneck here is estimating the second smallest eigenvalue of $D_x - B$, and he suggests that a randomized power method–like algorithm might suffice, but leaves the investigation for future research.

Here, we show how the power iteration detector fills this void in the theory. First, we note that in the interesting regime of (p, q), the number of nonzero entries in A is $O(n \log n)$ with high probability [26]. As such, the function $x \mapsto Bx$ can exploit this sparsity to take only $O(n \log n)$ operations. This in turn allows for the computation of the diagonal of D_x to cost $O(n \log n)$ operations. Next, note that

$$\begin{aligned} \|D_x - B\|_{2\to 2} &\leq \|D_x\|_{2\to 2} + \|2A - 11^\top\|_{2\to 2} + \|I\|_{2\to 2} \\ &\leq \|D_x\|_{2\to 2} + \|2A - 11^\top\|_F + 1 = \max_i |(D_x)_{ii}| + n + 1 =: \lambda, \end{aligned}$$

and that λ can be computed in $O(n)$ operations after computing the diagonal of D_x. Also, it takes $O(n)$ operations to verify $x^\top 1 = 0$. Assuming $x^\top 1 = 0$, then the second smallest eigenvalue of $D_x - B$ is strictly positive if and only if x spans the unique leading eigenspace of $\lambda I - D_x + B$. Thus, one may test this condition using the power iteration detector, and furthermore, each iteration will take only $O(n \log n)$ operations, thanks to the sparsity of A.

V. Creating a Robust Clustering for Air Force Health Care

5.1 Motivation and Previous Work

The Department of Defense (DoD) health care costs are growing more than twice as fast as economy-wide medical inflation. This creates a concern that cannot be ignored. That is, the extremely fast-paced health care costs may reduce resource availability for other important defense programs, and as such undermine the overall capability of the United States military. Given this serious concern, the DoD and Department of Veteran Affairs (VA) are currently re-engineering their health information technology systems and infrastructure.

In recent years, the authors of [1], sought to find a fundamentally new strategy that the authors claim will fix health care. They claim that the current system aims to optimize the wrong objective function. Examples of such objectives include maximizing access to care (even if the quality is poor), containing costs, and boosting profits. The right objective function, loosely defined, is to maximize "value for patients, where value is defined as the health outcomes achieved that matter to patients relative to the cost of achieving those outcomes." In [1], the strategy for this value based health care has six components:

- Organize into Integrated Practice Units (IPUs)
- Measure Outcomes and Costs for Every Patient
- Move to Bundled Payments for Care Cycles
- Integrate Care Delivery Systems
- Expand Geographic Reach
- Build an Enabling Information Technology Platform.

The authors of [1] cite multiple case studies where such strategy either fully or partially implemented has given good results and outperforms the current, predominant form of health care. We focus our attention on the first component. The motivation for an IPU is to restructure the way clinicians are organized to deliver care to organize around the patient and their needs. To demonstrate its effectiveness, the authors of [5] documented how a West German migraine headache center was able to lower costs by 20% and achieved a 54% improvement in symptoms in patients by restructuring to create an IPU. As a result, the center was able to expand adding more locations in new cities and develop new programs in conditions such as vertigo, rheumatoid arthritis, and acute back pain.

An IPU as given in [1] is described as fulfilling eleven main requirements:

- An IPU is organized around a medical condition or a set of closely related conditions (or around defined patient segments for primary care).
- Care is delivered by a dedicated, multidisciplinary team of clinicians who devote a significant portion of their time to the medical condition.
- Providers involved are members of or affiliated with a common organizational unit.
- Takes responsibility for the full cycle of care for the condition, encompassing outpatient, inpatient, and rehabilitative care as well as supporting services.
- Incorporates patient education, engagement, and follow-up as integral to care.
- Utilizes a single administrative and scheduling structure.
- Co-located in dedicated facilities.
- Care is led by a physician team captain and a care manager who oversee each patient's care process.

- Measures outcomes, costs, and processes for each patient using a common information platform.
- Providers function as a team, meeting formally and informally on a regular basis to discuss patients, processes and results.
- Accepts joint accountability for outcomes and costs.

We restrict our attention to the first requirement of an IPU. For this requirement, [1] gives the examples of kidney and eye disorders for patients with diabetes and palliative care for those with metastatic cancer. The same researchers have applied the IPU model to primary care in [4]. The first step in creating an IPU model is to identify partitions of a population of patients with similar needs, challenges, and ways to best access care. Certainly, the value of a health care system under this new model is a function of the choice of partition of the patient population. As such, it is natural to ask what patient population constitutes an optimal partition. That is, given a patient population and their medical data, how does one create a partition that maximizes the value of the health care system? In [4] the authors give what seems to be a reasonable set of categories of a patient population that one might use to create a patient partition:

1. Healthy adults
2. Mothers and young children
3. Adults at risk of a chronic disease
4. Adults with a chronic disease
5. Adults with a rare condition
6. The disabled and the frail elderly.

Notice a person could be included in more than one category as listed above. For example, a mother of a young child could also be diagnosed with a chronic disease. As such, we interpret the above set of categories to follow a certain hierarchy. That is, a patient is assigned to the partition requiring the most attention. Given the categories above, we can merely assign a patient to the highest labeled partition for which he or she is a member. Returning to the example, the mother of the young child would be assigned to partition 4 as she has a chronic disease. For reasons that will soon be evident, we call this partition the baseline partition.

The authors do not give much justification for this partition (or its motivating categorization) other than the fact it seems reasonable and state that there is no partition that will be optimal for every possible patient population. Moreover there seems to be little work currently in the literature of a methodology for determining how to create an optimal patient partition. Additionally, to our knowledge there seems to be no statistical work done in determining such a partition given a patient population.

Thus, our aim is to fulfill two goals in this chapter: to create a statistically based methodology for finding good patient partitions for the IPU model and provide a proof of concept of this IPU framework applied to Air Force personnel and their dependents. In particular, we seek to find a clustering of Air Force personnel and their dependents which is more robust to time than the baseline partition.

To this end we first apply the baseline partition to the dataset used in the IRB study. We detail this work in Appendix B as much of the methodology is merely checking if a patient meets certain properties. We do however build a classifier for adults at risk of a chronic disease. Next, we present alternate partitions that outperform the baseline partition in terms of conditional entropy based on a direct application of k-means.

5.2 The Dataset

Our database is taken from the Military Health System Data Mart (M2), the Medical Performance Reporting System (MEPRS), and the Preventive Health Assessment Individual Medical Readiness (PIMR) dataset. Our dataset after cleaning consists of 5,245,379 visits between 2003 and 2012 for 730,296 patients. In each visit information listed in Table 3 is recorded. In our analysis, we use the diagnosis codes instead of the procedural codes as the goal is to find common conditions for which we can create IPUs, not common procedures. Furthermore, there are more diagnoses recorded in the dataset than there are procedures. As the data was collected prior to 1 October, 2015, the diagnoses are all coded in ICD-9-CM. In the cleaning process, visits without a primary diagnosis were not considered. Additionally, patients with recorded ages that exceed ten years in difference were also excluded from consideration. In other words, if there existed two hospital visits with the same patient ID and the ages of the patient recorded for the two visits exceeded ten years, we remove all hospital visits with this patient ID. The reader may notice that SEX and GENDER have the same description, despite the subtle difference in the definition of the words themselves. In the dataset SEX and GENDER agree whenever one of them is not blank. Since entries in SEX have more blanks, we use GENDER instead. We surmise the reason for two redundant elements is an artifact of combining databases.

5.3 Determining Time-Robustness and Optimal k

Here we describe the methodology in how we measured how robust to time a patient partition is and how to select the best number of clusters.

We first formulate our observed data $\mathcal{M}$ as a matrix, where the rows correspond to patients and the columns correspond to ICD codes and where an entry of $\mathcal{M}$ corresponds to how many times a patient was given a certain ICD coding. As motivated

Table 3. Description of the Dataset

Variable Name	Variable description
SPONSOR PAY GRADE	Sponsor's pay grade
APPTYPE	Appointment type
FMP	Family Member Prefix
DX1	Principal Diagnosis
DX2	Diagnosis 2
DX3	Diagnosis 3
DX4	Diagnosis 4
PATCAT	Patient Category
FULL COST	Cost of the visit
PX1	Principal Procedure
PX2	Procedure 2
PX3	Procedure 3
PX4	Procedure 4
GENDER	Sex of the patient
SPONSOR SERVICE	Sponsor's branch of service
BEN CAT	Beneficiary category
PROVIDER SPECIALTY	Provider specialty code
COUNTABLE VISIT	Flag for a countable visit
YEAR	Year of visit
AGE	Age of patient in years
CLINIC SERVICE	Clinic service coding
DUP DATE FLAG	Flag for duplicated dates
MAJCOM	Major Command
PERSONID	Patient ID number
FAMILYID	Patient's family ID number
PROVIDERID	Patient's provider ID number
CLINICID	Patient's clinic ID number
SEX	Sex of the patient
DAFSC	Patient's Duty Air Force Specialty Code

in previous work in [46], we only used the category code of an ICD coding in $\mathcal{M}$. For example, if a patient received multiple concussions with ICD codes of 850.0, 850.11 and 850.12, we would record the patient having been given an ICD coding of 850 three times. Note that for an ICD code using the character 'E' in the first entry, the category code consists of the first four entries. In all other cases, the category coding corresponds to the first three alphanumeric entries.

We will end up running some clustering algorithm on the rows of $\mathcal{M}$ to find clusters of patients. At the same time we wish to evaluate the partition in terms of its predictive power. Intuitively, a patient should stay in the same partition as the conditions he or she has should not vary greatly. Certainly, given enough time a patient will change groups, so we restrict the time period for which our prediction should hold. As such, we train our model on the first five years of data $\mathcal{M}_1$ and test our partition using the second five years of the data $\mathcal{M}_2$. To measure the quality of our clustering we use entropy conditioned on assigned clusters to measure the amount of information in our proposed clustering, that is, a measure of how well our partition summarizes the patient population. This conditional entropy is expressed as

$$H(W|\mathbf{T}) = -\sum_{t\in\mathscr{T}} \Pr(t) \sum_{w\in\mathscr{W}} \Pr(w|t) \log_2 \big(\Pr(w|t)\big) \tag{34}$$

where W is a random variable denoting a patient being given an ICD code w with $\mathscr{W}$ denoting the set of all three digit ICD codings and $\mathbf{T}$ is a random variable denoting the partition to which a patient belongs with $\mathscr{T}$ being the support of $\mathbf{T}$. Denoting n_t to be the number of patients in the tth partition and n to be the number of patients in $\mathcal{M}_2$, we can express the following estimates of the probabilities required to to be calculated in (34);

$$\Pr(t) = \frac{n_t}{n} \qquad \text{and} \qquad \Pr(w|t) = \frac{1_t^\top \mathcal{M}_2 e_w}{1_t^\top \mathcal{M}_2 1}.$$

where 1 and e act as indicator functions of their respective subscripts in the usual manner.

Now a determination must be made in which patients are represented in $\mathcal{M}_1$ and $\mathcal{M}_2$. Naturally, there is a motivation to include all of the patients in $\mathcal{M}$. There are patients, however, who are recorded as exclusively visiting only in the first five years or the second five years of our data set. As such, if we include all of the patients observed over the course of the ten year span in $\mathcal{M}_1$, we will have zero rows in $\mathcal{M}_1$ and $\mathcal{M}_2$. If there is a zero row in $\mathcal{M}_1$ when we train our model, k-means will merely create a cluster of zero rows. A cluster of zero rows is equivalent identifying a cluster of patients with no medical histories. If we have no information on them, we should not be able predict their medical behavior in the future. On the other hand, if there is a zero row in $\mathcal{M}_2$, it will have no effect on entropy. As such, we include only the patients who have medical records in both the first five years and the second five years. We denote $\mathscr{M}_1, \mathscr{M}_2$ to describe medical histories of patients who have medical records in both the first five years and the second five years with rows organized such that the ith row in both $\mathscr{M}_1, \mathscr{M}_2$ corresponds to the same patient.

Now we briefly discuss the motivation behind our methodology behind choosing the best number of clusters k. Unfortunately, in the context of unsupervised learning, there only exist well-motivated heuristics for picking k. We use a popular heuristic called the "elbow method". We choose the k which results in the largest drop in conditional entropy (or the greatest desired change in the desired clustering metric) forming what looks like an elbow when plotted with varying k. Assume in the planted clustering, there are k^* clusters. When $k < k^*$, the pigeonhole principle dictates that at least one of the clusters contains at least two of the planted clusters. As such, moving from identifying this one cluster into the appropriate number of planted clusters should yield a large change in the clustering metric. Once $k \geq k^*$, the improvement in the clustering metric tends to be marginal. Thus, the k which resulted from the largest single step improvement is utilized as k^*. As such, we seek the largest single step decrease in conditional entropy.

5.4 Employing k-means on Air Force Personnel

In this section, we briefly mention why k-means was used to find patient partitions. More importantly, we detail how the best k-means clustering is achieved and give a visualization of this clustering.

Although the objective functions for k-means and conditional entropy look dissimilar, it was shown in [18] that the k-means objective function is approximately derived from the minimum entropy principle when Rényi quadratic entropy is used. Furthermore, under a certain distribution of data, the two objective functions are closely related. Consider data with low variance in entry-wise differences between points in the same planted cluster. A k-means optimal clustering will have small entry-wise differences. That is, the data from one planted cluster viewed one dimension at a time will be tightly packed around the mean. As such, the conditional probability of data being near to its assigned cluster centroid will be close to 1 (or 0 if assigned to an incorrect cluster). This near deterministic nature leads to the conditional entropy to be almost optimal. As such, it is reasonable to think minimizing k-means may lead to high performing clusterings under conditional entropy. Furthermore, the k-means problem and its heuristics are commonly used in data science and are available in many statistical packages. As such, any difficulty arising from replicating this analysis on another dataset of patient medical histories will not come from the availability of software implementations of this algorithm. Additionally, we require a clustering algorithm to determine the best number of clusters. It happens that when we examine $k = 2$, we are essentially performing the thresholding algorithm omitting row-sorting described in Chapter IV. Having established the utility of k-means, we now describe the methodology of obtaining our k-means based partition.

We first would like to transform $\mathscr{M}_1$ such that k-means will obtain a clustering which approximates a minimal entropy clustering. As we have demonstrated the intuition for minimizing the entry-wise differences between points in the same cluster, we transform $\mathscr{M}_1$ to $\widetilde{\mathscr{M}_1}$ such that any positive entry in $\mathscr{M}_1$ is a 1 in $\widetilde{\mathscr{M}_1}$ and zero elsewhere. For each value of k ranging from $k = 2$ to 10, we perform 10 trials of k-means++ in MATLAB 2015a on $\widetilde{\mathscr{M}_1}$ and retain the partition with the lowest entropy. Since we will vary k, we desire to find the best number of clusters in the data. As summarized in Table 4 and Figure 9, we notice every k-means partition for $k > 1$ outperforms the baseline model. Notice the largest drop off in entropy occurs between $k = 1$ and $k = 2$. As such, we conclude that under the k-means model, the best partitioning uses two clusters. An illustration of the data matrix $\mathscr{M}_2$ with the patients organized by cluster for our 2-means model is presented in Figure 7. Notice the two cluster patient partition has an interpretation in the context of a BiSBM due to our results on thresholding. To complete the thresholding heuristic as described in Section 4.3 (i.e. perform row sorting on $\mathscr{Q}$), we sort the columns of $\mathscr{M}_2$ according to a sorting of the quantity (33). We obtain another illustration of $\mathscr{M}_2$ from this sorting in Figure 8 which provides a starker contrast between the two clusters.

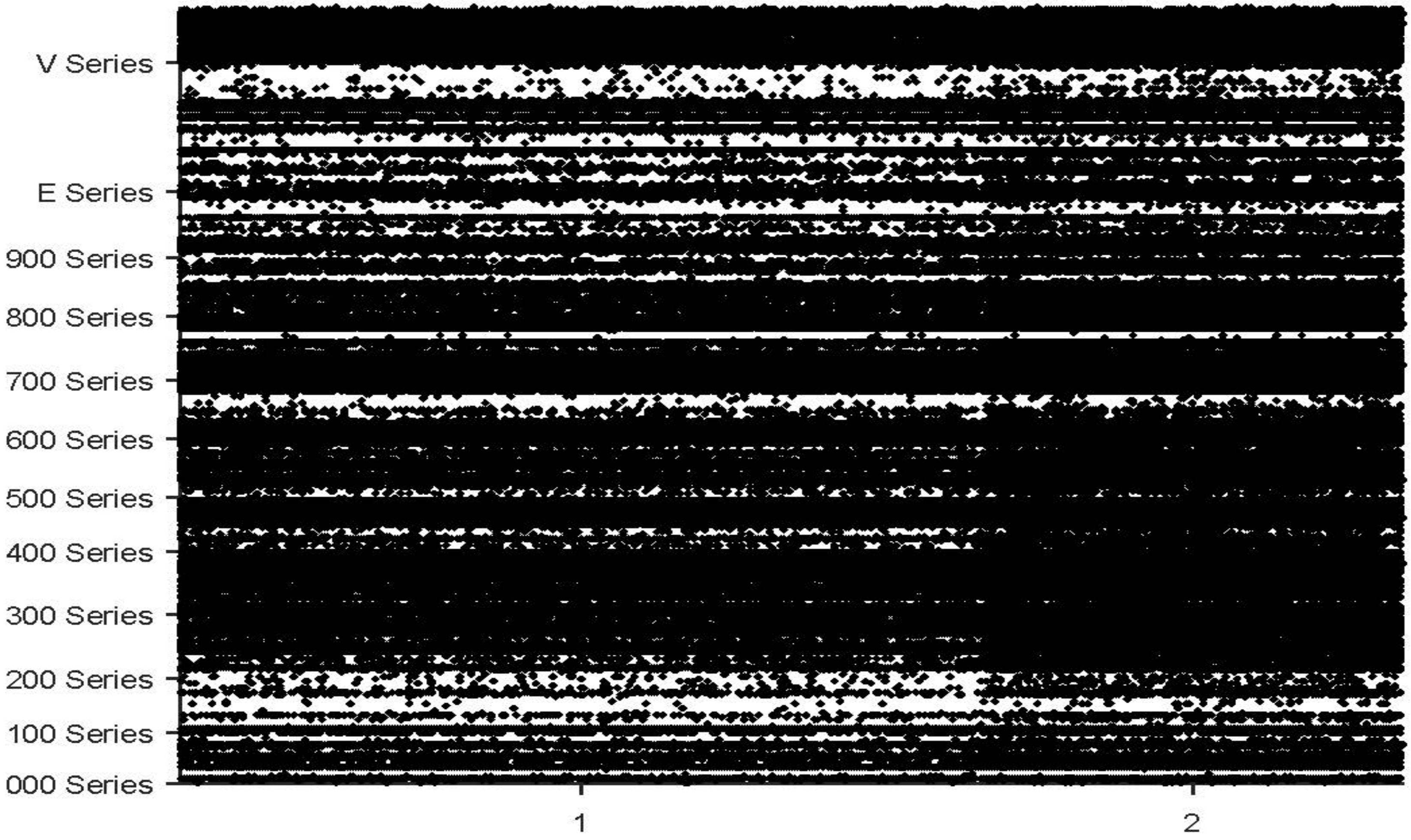


Figure 7. A Visualization of the Data Under the Best 2-means Clustering A black dot denotes a patient being given a certain ICD coding at least once between the years 2008-2012. The patients are sorted by cluster on the horizontal axis and the ICD codes are listed in order on the vertical axis.

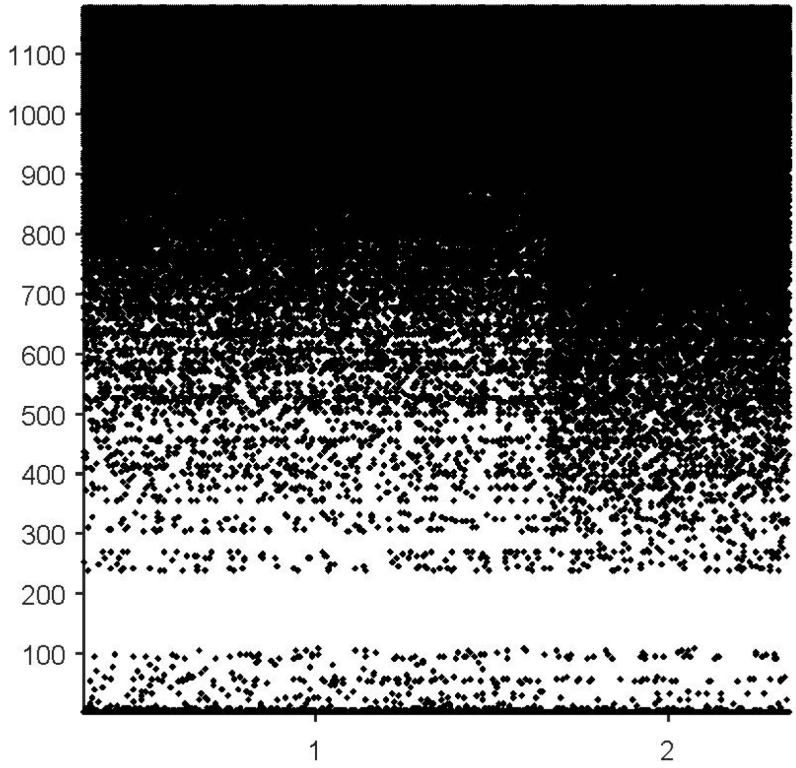


Figure 8. A Visualization of the Data Under the Best 2-means Clustering with Sorted ICD codings A black dot denotes a patient being given a certain ICD coding at least once between the years 2008-2012. The patients are sorted by cluster on the horizontal axis and the vertical axis denotes ICD codings. We sort the ICD codings using the quantity of the expected value of an ICD coding being given to a patient in cluster 2 minus the expected value of a patient in cluster 1 receiving the same coding. The ICD codings are sorted with the least values of this difference at the bottom and the greatest at the top. The labels on the vertical axis merely indicate the *i*th ICD code from the bottom.

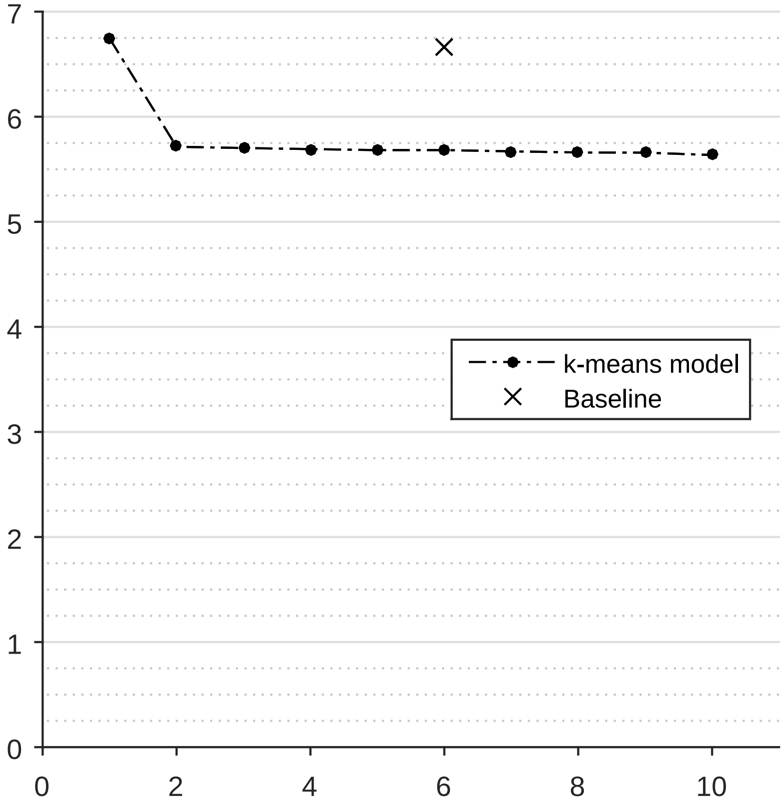


Figure 9. A Comparison of the Patient Partitions with Conditional Entropy The vertical axis denotes the conditional entropy measured in bits of information. The horizontal axis denotes the number of clusters k. For each value of k, the minimum entropy obtained over 10 trials is displayed for the k-means model.

Table 4. Summary of Clustering Entropies

Method	Minimum Entropy	Average Entropy	Std Dev of Entropy
Degenerate	6.7447	-	-
Baseline	6.6557	-	-
k-means with k=	-	-	-
2	5.7144	5.7276	0.0418
3	5.7020	5.7104	0.0036
4	5.6921	5.7037	0.0096
5	5.6822	5.6880	0.0056
6	5.6824	5.6924	0.0093
7	5.6707	5.6841	0.0102
8	5.6597	5.6735	0.0090
9	5.6584	5.6763	0.0121
10	5.6354	5.6595	0.0126

VI. Conclusions and Future Work

In summary, this thesis demonstrates the utility of clustering in Air Force health care reform and provides advances in clustering theory relevant to applying the IPU framework to the Air Force. We created a fast certifier for k-means optimality and a fast 2-means solver called spectral k-means which achieves state-of-the-art performance. Additionally, we created a PCC algorithm for a BiSBM and found guarantees for single-partite planted cluster recovery under a class of BiSBMs. Finally, we demonstrated a patient partition derived from k-means with an interpretation in the BiSBM that is more robust to time than the currently existing partition in the literature for the IPU framework. We conclude with areas for future work:

Future Work in the Theory

- Let $\Delta^\star(\mathcal{D}, k)$ denote the smallest value for which $\Delta > \Delta^\star$ implies that minimizing the k-means objective recovers planted clusters under the $(\mathcal{D}, \gamma, n)$-stochastic ball model with probability $1 - e^{-\Omega_{\mathcal{D},\gamma}(n)}$. What is $\Delta^\star$? It was conjectured in [22] that $\Delta^\star = 2$, but as we demonstrated in Section 3.2, this is not the case.

- Let $\Delta^\star_{\mathrm{SDP}}(\mathcal{D}, k)$ denote the smallest value for which $\Delta > \Delta^\star_{\mathrm{SDP}}$ implies that solving the k-means SDP recovers planted clusters under the $(\mathcal{D}, \gamma, n)$-stochastic ball model with probability $1 - e^{-\Omega_{\mathcal{D},\gamma}(n)}$. What is $\Delta^\star_{\mathrm{SDP}}$? Considering Section 3.2 and Figure 4(center), we suspect the SDP exhibits a performance gap: $\Delta^\star_{\mathrm{SDP}} > \Delta^\star$.

- Is there a single dual certificate for the k-means SDP that typically certifies planted clusters under the stochastic ball model whenever $\Delta > \Delta^\star_{\mathrm{SDP}}$? Does this certification have a quasilinear-time implementation similar to Section 2.3?

- Is there a quasilinear-time k-means solver that typically solves k-means under the stochastic ball model whenever $\Delta > \Delta^\star$? In particular, is there a quasilinear-time initialization of Lloyd's algorithm that meets this specification? Following the philosophy of Section 3.3, such algorithms should be designed so as to not "see" the stochastic ball model.

- Work in [31, 32, 30] had $p \to 0$ as $n \to \infty$ in their BiSBMs. The advantage of this constraint is that the model better represents real world data and as such a clustering algorithm devised under this model should better fit real world bipartite data. For this reason, finding a probably certifiably correct algorithm for a BiBSM with this constraint should prove to be more applicable to real world data. As such, does the BiSBM in [30] create a more robust model for IPU organization than k-means?

- Currently, our MLE based certifier on our BiSBM certifies an optimal solution only if $p > 1/2$ and $q < 1/2$. Using perhaps a different "relax and round" scheme, is it possible to increase the regime of p, q in certifying our BiSBM if we enforce the constraint of $p \to 0$ as $n \to \infty$?

- Recall the k-means optimal clustering is the threshold clustering for reasonably balance clusters and sufficient inter-cluster seperation. Given this relationship, what are the bounds on cluster imbalance and inter-cluster separation for which this relationship holds?

Future Work in the Application

- Is there another way to visualize the set of patient medical histories which allows us to cluster the patients? Additionally, t-distributed stochastic neighbor embedding described in [47] may also provide a nice way to perform a dimensionality reduction on our data to which we may cluster afterwards.

- This thesis provided a proof of concept for this IPU framework using statistically based methods which prove to give more time-robust clusterings than the only existing partition in the literature. As this work provides a proof of concept, much more work needs to be done in identifying meaningful, time-robust patient segmentations that can be employed in the Air Force.

Appendix A. Miscellaneous Proofs

A.A Deriving the Dual Program of the k-means Relaxation

To derive the dual of (2), we consider the general setting of cone programming (e.g., see [24]). Given closed convex cones K and L in real finite-dimensional vector spaces and a linear operator A, the dual of

$$\begin{aligned} \text{maximize} \quad & \langle c, x\rangle \\ \text{subject to} \quad & b - Ax \in L \\ & x \in K \end{aligned} \tag{35}$$

is given by

$$\begin{aligned} \text{minimize} \quad & \langle b, y\rangle \\ \text{subject to} \quad & A^* y - c \in K^* \\ & y \in L^* \end{aligned} \tag{36}$$

where A^* denotes the adjoint of A, while K^* and L^* denote the dual cones of K and L, respectively. In our case, $c = -D, x = X$, and K is simply the cone of positive semidefinite matrices (as is K^*). Before we determine L, we need to interpret the remaining constraints in (2). To this end, we note that $\operatorname{Tr}(X) = k$ is equivalent to $\langle X, I\rangle = k, X1 = 1$ is equivalent to having

$$\left\langle X, \frac{1}{2}(e_i 1^\top + 1 e_i^\top)\right\rangle = 1 \quad \forall i \in \{1, \ldots, N\},$$

and $X \geq 0$ is equivalent to having

$$\left\langle X, \frac{1}{2}(e_i e_j^\top + e_j e_i^\top) \right\rangle \geq 0 \quad \forall i, j \in \{1, \dots, N\}, i \leq j.$$

(These last two equivalences exploit the fact that X is symmetric.) We can express the remaining constraints in (2) as using a linear operator A that sends any matrix X to its inner products with $\{\langle X, \frac{1}{2}(e_i 1^\top + 1 e_i^\top)\rangle\}_{i=1}^N, \{\langle X, \frac{1}{2}(e_i e_j^\top + e_j e_i^\top)\rangle\}_{i,j=1,i\leq j}^N$, and I. The remaining constraints in (2) are equivalent to having $b - Ax \in L$ where $b = k \oplus 1 \oplus 0$ and $L = 0 \oplus 0 \oplus \mathbb{R}_{\geq 0}^{N(N+1)/2}$. Writing $y = z \oplus \alpha \oplus (-\beta)$, the dual of (2) is then given by

$$\begin{aligned} \text{minimize} \quad & kz + \sum_{i=1}^N \alpha_i && (37) \\ \text{subject to} \quad & zI + \sum_{i=1}^N \alpha_i \cdot \frac{1}{2}(e_i 1^\top + 1 e_i^\top) - \sum_{i=1}^N \sum_{j=i}^N \beta_{ij} \cdot \frac{1}{2}(e_i e_j^\top + e_j e_i^\top) + D \succeq 0 \\ & \beta \geq 0 \end{aligned}$$

A.B Proof of Proposition 1

The proof of Proposition 1 relies on the following standard result:

Proposition 20 (e.g., see [24]). *Suppose the primal program* (35) *and dual program* (36) *are feasible and bounded.*

(a) **Strong duality.** *The primal program* (35) *has optimal value* val *if and only if the dual program* (36) *has bounded optimal value* val.

(b) **Complementary slackness.** *The decision variables x and y are optimal in* (35) *and* (36), *respectively, if and only if*

$$\langle A^* y - c, x\rangle = \langle y, b - Ax\rangle.$$

Proof of Proposition 1. (a)⇔(b): By complementary slackness, (a) is equivalent to having both

$$\langle A^* y - c, X\rangle = 0$$

and

$$\langle y, b - A(X)\rangle = 0. \tag{38}$$

Since $Q \succeq 0$, we have

$$\langle A^* - c, X\rangle = \langle Q, X\rangle = \left\langle Q, \sum_{t=1}^{k} \frac{1}{n_t} 1_t 1_t^\top \right\rangle = \sum_{t=1}^{k} \frac{1}{n_t} 1_t^\top Q 1_t \geq 0,$$

with equality if and only if $Q1_a = 0$ for every $a \in \{1, \ldots, k\}$. Next, we recall that $y = z \oplus \alpha \oplus (-\beta), b - A(X) \in L = 0 \oplus 0 \oplus \mathbb{R}_{\geq 0}^{N(N+1)/2}$, and $b = k \oplus 1 \oplus 0$. As such, (38) is equivalent to β having disjoint support with $\{\langle X, \frac{1}{2}(e_i e_j^\top + e_j e_i^\top)\rangle\}_{i,j=1, i \leq j}^{N}$, i.e., $\beta^{(a,a)} = 0$ for every cluster a.

(b)⇔(c): Take any solution to the dual program (3), and note that

$$\begin{aligned} Q^{(a,a)} &= zI + \left(\sum_{t=1}^{k} \sum_{i \in t} \alpha_{t,i} \cdot \frac{1}{2}(e_i 1^\top + 1 e_i^\top) \right)^{(a,a)} - \beta^{(a,a)} + D^{(a,a)} \\ &= zI + \left(\sum_{i \in a} \alpha_{a,i} \cdot \frac{1}{2}(e_i 1^\top + 1 e_i^\top) \right)^{(a,a)} + D^{(a,a)}, \end{aligned}$$

where the 1 vectors in the second line are n_a-dimensional (instead of N-dimensional, as in the first line), and similarly for e_i (instead of $e_{t,i}$). We now consider each entry of $Q^{(a,a)}1$, which is zero by assumption:

$$\begin{aligned}
0 &= e_r^\top Q^{(a,a)} 1 \\
&= e_r^\top \left(zI + \sum_{i \in a} \alpha_{a,i} \cdot \frac{1}{2}(e_i 1^\top + 1 e_i^\top) + D^{(a,a)} \right) 1 \\
&= z + \sum_{i \in a} \alpha_{a,i} \cdot \frac{1}{2}(e_r^\top e_i 1^\top + e_r^\top 1 e_i^\top) + e_r^\top D^{(a,a)} 1 \\
&= z + \sum_{i \in a} \alpha_{a,i} \cdot \frac{1}{2}(n_a \delta_{ir} + 1) + e_r^\top D^{(a,a)} 1. \qquad (39)
\end{aligned}$$

As one might expect, the n_a linear equations determine the variables $\{\alpha_{a,i}\}_{i \in a}$. To solve this system, we first observe

$$\begin{aligned}
0 &= 1^\top Q^{(a,a)} 1 \\
&= 1^\top \left(zI + \sum_{i \in a} \alpha_{a,i} \cdot \frac{1}{2}(e_i 1^\top + 1 e_i^\top) + D^{(a,a)} \right) 1 \\
&= n_a z + \sum_{i \in a} \alpha_{a,i} \cdot \frac{1}{2}(1^\top e_i 1^\top 1 + 1^\top 1 e_i^\top 1) + 1^\top D^{(a,a)} 1 \\
&= n_a z + n_a \sum_{i \in a} \alpha_{a,i} + 1^\top D^{(a,a)} 1,
\end{aligned}$$

and so rearranging gives

$$\sum_{i \in a} \alpha_{a,i} = -z - \frac{1}{n_a} 1^\top D^{(a,a)} 1.$$

We use this identity to continue (39):

$$
\begin{aligned}
0 &= z + \sum_{i\in a} \alpha_{a,i} \cdot (n_a \delta_{ir} + 1) + e_r^\top D^{(a,a)} 1 \\
&= z + \frac{n_a}{2}\alpha_{a,r} + \frac{1}{2}\sum_{i\in a} \alpha_{a,i} + e_r^\top D^{(a,a)} 1 \\
&= z + \frac{n_a}{2}\alpha_{a,r} + \frac{1}{2}\left(-z - \frac{1}{n_a} 1^\top D^{(a,a)} 1\right) + e_r^\top D^{(a,a)} 1
\end{aligned}
$$

and rearranging yields the desired formula for $\alpha_{a,r}$.

(c)⇔(a): Take any solution to the dual program (3). Then by assumption, the dual objective at this point is given by

$$
\begin{aligned}
kz + \sum_{t=1}^{k}\sum_{i\in t} \alpha_{t,i} &= kz + \sum_{t=1}^{k}\sum_{i\in t}\left(-\frac{1}{n_t} + \frac{1}{n_t^2} 1^\top D^{(t,t)} 1 - \frac{2}{n_t} e_i^\top D^{(t,t)} 1\right) \\
&= -\sum_{t=1}^{k} \frac{1}{n_t} 1^\top D^{(t,t)} 1 \\
&= -\operatorname{Tr}(DX),
\end{aligned}
$$

i.e., the primal objective (2) evaluated at X. Since X is feasible in the primal program, we conclude that X is optimal by strong duality. □

A.C Proof of Corollary 10

It suffices to have

$$
\|P_{\Lambda^\perp} M P_{\Lambda^\perp}\|_{2\to 2} + \|P_{\Lambda^\perp} B P_{\Lambda^\perp}\|_{2\to 2} \le z. \tag{40}
$$

We will bound the terms in (40) separately and then combine the bounds to derive a sufficient condition for Theorem 3. To bound the first term in (40), let ν be the $N \times 1$ vector whose (a,i)th entry is $\|x_{a,i}\|_2^2$, and let Φ be the $m \times N$ matrix whose

(a,i)th column is $x_{a,i}$. Then

$$D_{(a,i),(b,j)} = \|x_{a,i}-x_{b,j}\|_2^2 = \|x_{a,i}\|_2^2 - 2x_{a,i}^\top x_{b,j} + \|x_{b,j}\|_2^2 = (\nu 1^\top - 2\Phi^\top\Phi + 1\nu^\top)_{(a,i),(b,j)},$$

meaning $D = \nu 1^\top - 2\Phi^\top\Phi + 1\nu^\top$. With this, we appeal to the blockwise definition of M (5):

$$\begin{aligned}\|P_{\Lambda^\perp}MP_{\Lambda^\perp}\|_{2\to 2} = \|P_{\Lambda^\perp}DP_{\Lambda^\perp}\|_{2\to 2} &= \|P_{\Lambda^\perp}(\nu 1^\top - 2\Phi^\top\Phi + 1\nu^\top)P_{\Lambda^\perp}\|_{2\to 2}\\ &= 2\|P_{\Lambda^\perp}\Phi^\top\Phi P_{\Lambda^\perp}\|_{2\to 2} = 2\|\Phi P_{\Lambda^\perp}\|_{2\to 2}^2 = 2\|\Psi\|_{2\to 2}^2.\end{aligned}$$

For the second term in (40), we first write the decomposition

$$B = \sum_{a=1}^{k}\sum_{b=a+1}^{k}\Big(H_{(a,b)}(B^{(a,b)}) + H_{(b,a)}(B^{(b,a)})\Big),$$

where $H_{(a,b)}\colon \mathbb{R}^{n_a\times n_b} \to \mathbb{R}^{N\times N}$ produces a matrix whose (a,b)th block is the input matrix, and is otherwise zero. Then

$$\begin{aligned}P_{\Lambda^\perp}BP_{\Lambda^\perp} &= \sum_{a=1}^{k}\sum_{b=a+1}^{k} P_{\Lambda^\perp}\Big(H_{(a,b)}(B^{(a,b)}) + H_{(b,a)}(B^{(b,a)})\Big)P_{\Lambda^\perp}\\ &= \sum_{a=1}^{k}\sum_{b=a+1}^{k}\Big(H_{(a,b)}(P_{1^\perp}B^{(a,b)}P_{1^\perp}) + H_{(b,a)}(P_{1^\perp}B^{(b,a)}P_{1^\perp})\Big),\end{aligned}$$

and so the triangle inequality gives

$$\begin{aligned}\|P_{\Lambda^\perp}BP_{\Lambda^\perp}\|_{2\to 2} &\le \sum_{a=1}^{k}\sum_{b=a+1}^{k}\|H_{(a,b)}(P_{1^\perp}B^{(a,b)}P_{1^\perp}) + H_{(b,a)}(P_{1^\perp}B^{(b,a)}P_{1^\perp})\|_{2\to 2}\\ &= \sum_{a=1}^{k}\sum_{b=a+1}^{k}\|P_{1^\perp}B^{(a,b)}P_{1^\perp}\|_{2\to 2},\end{aligned}$$

where the last equality can be verified by considering the spectrum of the square:

$$\begin{aligned}&\Big(H_{(a,b)}(P_{1^\perp}B^{(a,b)}P_{1^\perp}) + H_{(b,a)}(P_{1^\perp}B^{(b,a)}P_{1^\perp})\Big)^2\\ &\quad = H_{(a,a)}\Big((P_{1^\perp}B^{(a,b)}P_{1^\perp})(P_{1^\perp}B^{(a,b)}P_{1^\perp})^\top\Big) + H_{(b,b)}\Big((P_{1^\perp}B^{(a,b)}P_{1^\perp})^\top(P_{1^\perp}B^{(a,b)}P_{1^\perp})\Big).\end{aligned}$$

At this point, we use the definition of B (8) to get

$$\|P_{1^\perp}B^{(a,b)}P_{1^\perp}\|_{2\to 2} = \frac{\|P_{1^\perp}u_{(a,b)}\|_2\|P_{1^\perp}u_{(b,a)}\|_2}{\rho_{(a,b)}}.$$

Recalling the definition of $u_{(a,b)}$ (8) and combining these estimates then produces the result.

A.D Proof of Theorem 11

In this section, we apply the certificate from Corollary 10 to the $(\mathcal{D}, \gamma, n)$-stochastic ball model (see Definition 9) to prove our main result. We will prove Theorem 11 with the help of several lemmas. Figure 10 provides a graphical guide to how the Lemmas are used in proving this Theorem.

Lemma 21. *Denote*

$$c_a := \frac{1}{n}\sum_{i=1}^{n} x_{a,i}, \qquad \Delta_{ab} := \|\gamma_a - \gamma_b\|_2, \qquad O_{ab} := \frac{\gamma_a + \gamma_b}{2}.$$

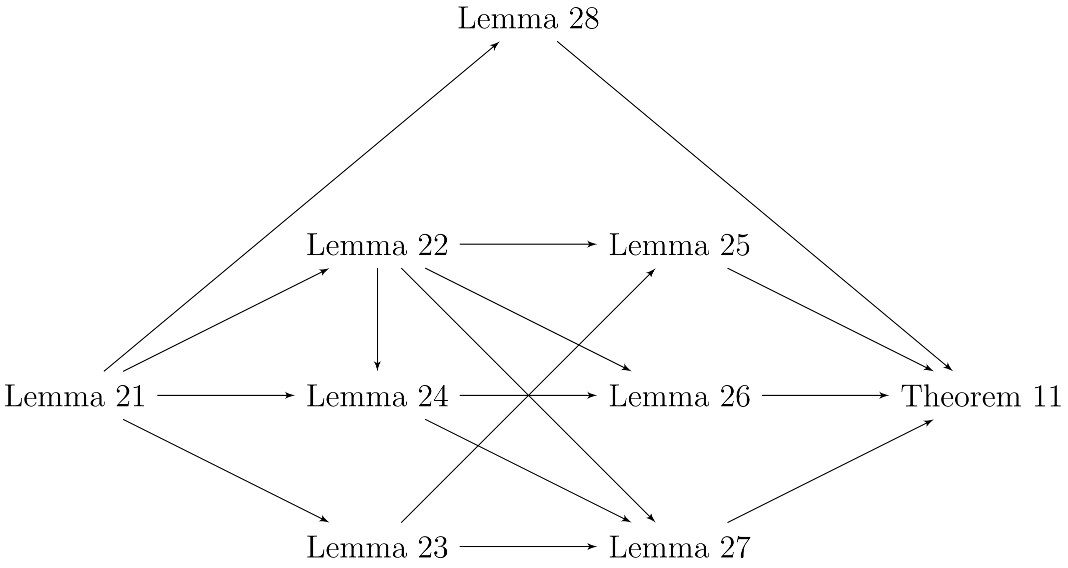


Figure 10. A directed graph to illustrate the development of the Lemmas in proving Theorem 11 where the predecessor of a statement is included in the statement's proof.

Then the $(\mathcal{D}, \gamma, n)$-stochastic ball model satisfies the following estimates:

$$\|c_a - \gamma_a\|_2 < \epsilon \qquad w.p. \qquad 1 - e^{-\Omega_{m,\epsilon}(n)} \tag{41}$$

$$\left| \frac{1}{n} \sum_{i=1}^{n} \|r_{a,i}\|_2^2 - \mathbb{E}\|r\|_2^2 \right| < \epsilon \qquad w.p. \qquad 1 - e^{-\Omega_{\epsilon}(n)} \tag{42}$$

$$\left| \frac{1}{n} \sum_{i=1}^{n} \|x_{a,i} - O_{ab}\|_2^2 - \mathbb{E}\|r + \gamma_a - O_{ab}\|_2^2 \right| < \epsilon \qquad w.p. \qquad 1 - e^{-\Omega_{\Delta_{ab},\epsilon}(n)} \tag{43}$$

Proof. Since $\mathbb{E}r = 0$ and $\|r\|_2^2 \le 1$ almost surely, one may lift

$$X_{a,i} := \begin{bmatrix} 0 & r_{a,i}^{\top} \\ r_{a,i} & 0 \end{bmatrix}$$

and apply the Matrix Hoeffding inequality [48] to conclude that

$$\Pr\left(\left\| \sum_{i=1}^{n} r_{a,i} \right\|_2 \ge t \right) \le m e^{-t^2/8n}.$$

Taking $t := \epsilon n$ then gives (41). For (42) and (43), notice that the random variables in each sum are iid and confined to an interval almost surely, and so the result follows from Hoeffding's inequality. $\square$

Lemma 22. *Under the* $(\mathcal{D},\gamma,n)$*-stochastic ball model, we have* $D^{(a,b)}1 - D^{(a,a)}1 = 4np + q$*, where*

$$\begin{aligned} p_i &:= r_{a,i}^\top(\gamma_a - O_{ab}) + \frac{\Delta_{ab}^2}{4} \\ q_i &:= 2n(x_{a,i} - O_{ab})^\top\Big((c_a - c_b) - (\gamma_a - \gamma_b)\Big) \\ &\qquad + \bigg(\sum_{j=1}^n \|x_{b,j} - O_{ab}\|_2^2 - \sum_{j=1}^n \|x_{a,j} - O_{ab}\|_2^2\bigg) \end{aligned}$$

and $|q_i| \leq (6 + 2\Delta_{ab})n\epsilon$ *with probability* $1 - e^{-\Omega_{m,\Delta_{ab},\epsilon}(n)}$.

Proof. Add and subtract O_{ab} and then expand the squares to get

$$\begin{aligned} e_i^\top(D^{(a,b)}1 - D^{(a,a)}1) &= \sum_{j=1}^n \|x_{a,i} - x_{b,j}\|_2^2 - \sum_{j=1}^n \|x_{a,i} - x_{a,j}\|_2^2 \\ &= n\bigg(-2(x_{a,i} - O_{ab})^\top(c_b - O_{ab}) + \frac{1}{n}\sum_{j=1}^n \|x_{b,j} - O_{ab}\|_2^2\bigg) \\ &\qquad - n\bigg(-2(x_{a,i} - O_{ab})^\top(c_a - O_{ab}) + \frac{1}{n}\sum_{j=1}^n \|x_{a,j} - O_{ab}\|_2^2\bigg) \\ &= 2n(x_{a,i} - O_{ab})^\top(c_a - c_b) \\ &\qquad + \bigg(\sum_{j=1}^n \|x_{b,j} - O_{ab}\|_2^2 - \sum_{j=1}^n \|x_{a,j} - O_{ab}\|_2^2\bigg). \end{aligned}$$

Add and subtract $\gamma_a - \gamma_b$ to $c_a - c_b$ and distribute over the resulting sum to obtain

$$\begin{aligned} e_i^\top(D^{(a,b)}1 - D^{(a,a)}1) &= 2n(x_{a,i} - O_{ab})^\top(\gamma_a - \gamma_b) + q \\ &= 4n\Big(r_{a,i} + (\gamma_a - O_{ab})\Big)^\top(\gamma_a - O_{ab}) + q. \end{aligned}$$

Distributing and identifying $\|\gamma_a - O_{ab}\|_2^2 = \Delta_{ab}^2/4$ explains the definition of p. To show $|q_i| \leq (6 + 2\Delta_{ab})n\epsilon$, apply the triangle and Cauchy–Schwarz inequalities to obtain

$$\begin{aligned} |q_i| &\leq \Big| 2n(x_{a,i} - O_{ab})^\top \Big((c_a - c_b) - (\gamma_a - \gamma_b) \Big) \Big| \\ &\quad + \Big| \sum_{j=1}^n \|x_{b,j} - O_{ab}\|_2^2 - \sum_{j=1}^n \|x_{a,j} - O_{ab}\|_2^2 \Big| \\ &\leq 2n \Big(\|r_{a,i}\|_2 + \|\gamma_a - O_{a,b}\|_2 \Big) \Big(\|c_a - \gamma_a\|_2 + \|c_b - \gamma_b\|_2 \Big) \\ &\quad + \Big| \sum_{j=1}^n \|x_{b,j} - O_{ab}\|_2^2 - \sum_{j=1}^n \|x_{a,j} - O_{ab}\|_2^2 \Big| \\ &\leq 2n \Big(1 + \frac{\Delta_{ab}}{2} \Big) \Big(\|c_a - \gamma_a\|_2 + \|c_b - \gamma_b\|_2 \Big) \\ &\quad + \Big| \sum_{j=1}^n \|x_{b,j} - O_{ab}\|_2^2 - \sum_{j=1}^n \|x_{a,j} - O_{ab}\|_2^2 \Big|. \end{aligned}$$

To finish the argument, apply (41) to the first term while adding and subtracting

$$\mathbb{E}\|r + \gamma_a - O_{ab}\|_2^2 = \mathbb{E}\|r + \gamma_b - O_{ab}\|_2^2,$$

from the second and apply (43). □

Lemma 23. *Under the $(\mathcal{D}, \gamma, n)$-stochastic ball model, we have*

$$\left| \frac{1}{n} 1^\top D^{(a,a)} 1 - 2n\mathbb{E}\|r\|_2^2 \right| \leq 4n\epsilon \qquad w.p. \qquad 1 - e^{-\Omega_{\Delta_{ab},\epsilon}(n)}.$$

Proof. Add and subtract γ_a and expand the square to get

$$\frac{1}{n} e_i^\top D^{(a,a)} 1 = \frac{1}{n} \sum_{j=1}^n \|x_{a,i} - x_{a,j}\|_2^2 = \|r_{a,i}\|_2^2 - 2r_{a,i}^\top (c_a - \gamma_a) + \frac{1}{n} \sum_{j=1}^n \|r_{a,j}\|_2^2.$$

The triangle and Cauchy–Schwarz inequalities then give

$$\begin{aligned}
&\bigg|\frac{1}{n}1^\top D^{(a,a)}1 - 2n\mathbb{E}\|r\|_2^2\bigg| \\
&\qquad= \bigg|\sum_{i=1}^{n}\bigg(\|r_{a,i}\|_2^2 - 2r_{a,i}^\top(c_a-\gamma_a) + \frac{1}{n}\sum_{j=1}^{n}\|r_{a,j}\|_2^2\bigg) - 2n\mathbb{E}\|r\|_2^2\bigg| \\
&\qquad\le n\bigg|\frac{1}{n}\sum_{i=1}^{n}\|r_{a,i}\|_2^2 - \mathbb{E}\|r\|_2^2\bigg| + 2\sum_{i=1}^{n}|r_{a,i}^\top(c_a-\gamma_a)| + n\bigg|\frac{1}{n}\sum_{j=1}^{n}\|r_{a,j}\|_2^2 - \mathbb{E}\|r\|_2^2\bigg| \\
&\qquad\le n\bigg|\frac{1}{n}\sum_{i=1}^{n}\|r_{a,i}\|_2^2 - \mathbb{E}\|r\|_2^2\bigg| + 2\sum_{i=1}^{n}\|c_a-\gamma_a\|_2 + n\bigg|\frac{1}{n}\sum_{j=1}^{n}\|r_{a,j}\|_2^2 - \mathbb{E}\|r\|_2^2\bigg| \\
&\qquad\le 4n\epsilon,
\end{aligned}$$

where the last step occurs with probability $1-e^{-\Omega_{\Delta_{ab},\epsilon}(n)}$ by a union bound over (42) and (41). □

Lemma 24. *Under the $(\mathcal{D},\gamma,n)$-stochastic ball model, we have*

$$1^\top D^{(a,b)}1 - 1^\top D^{(a,a)}1 \ge n^2\Delta_{ab}^2 - (6+4\Delta_{ab})n^2\epsilon \qquad w.p. \qquad 1-e^{-\Omega_{m,\Delta_{ab},\epsilon}(n)}.$$

Proof. Lemma 22 gives

$$\begin{aligned}
1^\top D^{(a,b)}1 - 1^\top D^{(a,a)}1 &= 1^\top(4np+q) \\
&\ge 4n\sum_{i=1}^{n}\bigg(r_{a,i}^\top(\gamma_a - O_{ab}) + \frac{\Delta_{ab}^2}{4}\bigg) - (6+2\Delta_{ab})n^2\epsilon \\
&\ge 4n\bigg(n(c_a-\gamma_a)^\top(\gamma_a-O_{ab}) + \frac{n\Delta_{ab}^2}{4}\bigg) - (6+2\Delta_{ab})n^2\epsilon.
\end{aligned}$$

Cauchy–Schwarz along with (41) then gives the result. □

Lemma 25. *Under the $(\mathcal{D},\gamma,n)$-stochastic ball model, there exists $C = C(\gamma)$ such* *that*

$$\min_{\substack{a,b\in\{1,\ldots,k\}\\ a\neq b}} \min(M^{(a,b)}1) \geq n\Delta(\Delta-2)+Cn\epsilon \qquad w.p. \qquad 1-e^{-\Omega_{m,\gamma,\epsilon}(n)},$$

where $\Delta := \min_{\substack{a,b\in\{1,\ldots,k\}\\ a\neq b}} \Delta_{ab}$.

Proof. Fix a and b. Then by Lemma 22, the following holds with probability $1-e^{-\Omega_{m,\Delta_{ab},\epsilon}(n)}$:

$$\begin{aligned}\min\Big(D^{(a,b)}1-D^{(a,a)}1\Big) &\geq 4n\min_{i\in\{1,\ldots,n\}}\bigg(r_{a,i}^\top(\gamma_a-O_{ab})+\frac{\Delta_{ab}^2}{4}\bigg)-(6+2\Delta_{ab})n\epsilon\\ &\geq n\Delta_{ab}^2-2n\Delta_{ab}-(6+2\Delta_{ab})n\epsilon,\end{aligned}$$

where the last step is by Cauchy–Schwarz. Taking a union bound with Lemma 23 then gives

$$\begin{aligned}&\min(M^{(a,b)}1)\\ &\quad=\min\Big(D^{(a,b)}1-D^{(a,a)}1\Big)+\frac{1}{2}\bigg(\frac{1}{n}1^\top D^{(a,a)}1-\frac{1}{n}1^\top D^{(b,b)}1\bigg)\\ &\quad\geq\min\Big(D^{(a,b)}1-D^{(a,a)}1\Big)\\ &\quad\quad-\frac{1}{2}\bigg(\bigg|\frac{1}{n}1^\top D^{(a,a)}1-2n\mathbb{E}\|r\|_2^2\bigg|+\bigg|\frac{1}{n}1^\top D^{(b,b)}1-2n\mathbb{E}\|r\|_2^2\bigg|\bigg)\\ &\quad\geq n\Delta_{ab}(\Delta_{ab}-2)-(10+2\Delta_{ab})n\epsilon\end{aligned}$$

with probability $1-e^{-\Omega_{\Delta_{ab},\epsilon}(n)}$. The result then follows from a union bound over a and b. □

Lemma 26. *Suppose* $\epsilon\leq 1$. *Then there exists* $C=C(\Delta_{ab},m)$ *such that under the*

$(\mathcal{D}, \gamma, n)$-stochastic ball model, we have

$$\|P_{1^\perp} M^{(a,b)} 1\|_2^2 \le \frac{4n^3 \Delta_{ab}^2}{m} + Cn^3\epsilon$$

with probability $1 - e^{-\Omega_{m,\Delta_{ab},\epsilon}(n)}$.

Proof. First, a quick calculation reveals

$$\begin{aligned} e_i^\top M^{(a,b)} 1 &= e_i^\top D^{(a,b)} 1 - e_i^\top D^{(a,a)} 1 + \frac{1}{2}\bigg(\frac{1}{n} 1^\top D^{(a,a)} 1 - \frac{1}{n} 1^\top D^{(b,b)} 1\bigg), \\ \frac{1}{n} 1^\top M^{(a,b)} 1 &= \frac{1}{n} 1^\top D^{(a,b)} 1 - \frac{1}{2}\bigg(\frac{1}{n} 1^\top D^{(a,a)} 1 + \frac{1}{n} 1^\top D^{(b,b)} 1\bigg), \end{aligned}$$

from which it follows that

$$\begin{aligned} e_i^\top P_{1^\perp} M^{(a,b)} 1 &= e_i^\top M^{(a,b)} 1 - \frac{1}{n} 1^\top M^{(a,b)} 1 \\ &= \bigg(e_i^\top D^{(a,b)} 1 - \frac{1}{n} 1^\top D^{(a,b)} 1\bigg) - \bigg(e_i^\top D^{(a,a)} 1 - \frac{1}{n} 1^\top D^{(a,a)} 1\bigg) \\ &= e_i^\top P_{1^\perp} (D^{(a,b)} 1 - D^{(a,a)} 1). \end{aligned}$$

As such, we have

$$\begin{aligned} \|P_{1^\perp} M^{(a,b)} 1\|_2^2 &= \|P_{1^\perp} (D^{(a,b)} 1 - D^{(a,a)} 1)\|_2^2 \\ &= \|D^{(a,b)} 1 - D^{(a,a)} 1\|_2^2 - \|P_1 (D^{(a,b)} 1 - D^{(a,a)} 1)\|_2^2. \end{aligned} \tag{44}$$

To bound the first term, we apply the triangle inequality over Lemma 22:

$$\|D^{(a,b)} 1 - D^{(a,a)} 1\|_2 \le 4n\|p\|_2 + \|q\|_2 \le 4n\|p\|_2 + (6 + 2\Delta_{ab}) n^{3/2} \epsilon. \tag{45}$$

We proceed by bounding $\|p\|_2$. To this end, note that the p_i's are iid random variables whose outcomes lie in a finite interval (of width determined by Δ_{ab}) with probability

1. As such, Hoeffding's inequality gives

$$\left|\frac{1}{n}\sum_{i=1}^{n} p_i^2 - \mathbb{E}p_1^2\right| \le \epsilon \qquad \text{w.p.} \qquad 1 - e^{-\Omega_{\Delta_{ab},\epsilon}(n)}.$$

With this, we then have

$$\|p\|_2^2 = n\left(\frac{1}{n}\sum_{i=1}^{n} p_i^2 - \mathbb{E}p_1^2 + \mathbb{E}p_1^2\right) \le n\mathbb{E}p_1^2 + n\epsilon \tag{46}$$

in the same event. To determine $\mathbb{E}p_1^2$, first take $r_1 := e_1^\top r$. Then since the distribution of r is rotation invariant, we may write

$$p_1 = r_{a,1}^\top(\gamma_a - O_{ab}) + \|\gamma_a - O_{ab}\|_2^2 = \frac{\Delta_{ab}}{2} r_1 + \frac{\Delta_{ab}^2}{4},$$

where the second equality above is equality in distribution. We then have

$$\mathbb{E}p_1^2 = \mathbb{E}\left(\frac{\Delta_{ab}}{2} r_1 + \frac{\Delta_{ab}^2}{4}\right)^2 = \frac{\Delta_{ab}^2}{4}\mathbb{E}r_1^2 + \frac{\Delta_{ab}^4}{16}. \tag{47}$$

We also note that $1 \ge \mathbb{E}\|r\|_2^2 = m\mathbb{E}r_1^2$ by linearity of expectation, and so

$$\mathbb{E}r_1^2 \le \frac{1}{m}. \tag{48}$$

Combining (45), (46), (47) and (48) then gives

$$\|D^{(a,b)}1 - D^{(a,a)}1\|_2 \le \left(\frac{4n^3\Delta_{ab}^2}{m} + n^3\Delta_{ab}^4 + 16n^3\epsilon\right)^{1/2} + (6 + 2\Delta_{ab})n^{3/2}\epsilon. \tag{49}$$

To bound the second term of (44), first note that

$$\|P_1(D^{(a,b)}1 - D^{(a,a)}1)\|_2 = \frac{1}{\sqrt{n}}\left|1^\top D^{(a,b)}1 - 1^\top D^{(a,a)}1\right|. \tag{50}$$

Lemma 24 then gives

$$\left|1^\top D^{(a,b)}1 - 1^\top D^{(a,a)}1\right| \geq 1^\top D^{(a,b)}1 - 1^\top D^{(a,a)}1 \geq n^2\Delta_{ab}^2 - (6+4\Delta_{ab})n^2\epsilon \tag{51}$$

with probability $1-e^{-\Omega_{m,\Delta_{ab},\epsilon}(n)}$. Using (44) to combine (49) with (50) and (51) then gives the result. □

Lemma 27. *There exists $C = C(\gamma)$ such that under the $(\mathcal{D},\gamma,n)$-stochastic ball model, we have*

$$\rho_{(a,b)} \geq n^2\big(\Delta_{ab}^2 - \Delta(\Delta-2)\big) - Cn^2\epsilon \qquad w.p. \qquad 1-e^{-\Omega_{\mathcal{D},\gamma,\epsilon}(n)}.$$

Proof. Recall from (8) that

$$\rho_{(a,b)} = u_{(a,b)}^\top 1 = 1^\top M^{(a,b)}1 - nz = 1^\top M^{(a,b)}1 - n \min_{\substack{a,b\in\{1,\ldots,k\}\\ a\neq b}} \min(M^{(a,b)}1). \tag{52}$$

To bound the first term, we leverage Lemma 24:

$$\begin{aligned}1^\top M^{(a,b)}1 &= 1^\top D^{(a,b)}1 - \frac{1}{2}(1^\top D^{(a,a)}1 + 1^\top D^{(b,b)}1)\\ &= \frac{1}{2}\Big(1^\top D^{(a,b)}1 - 1^\top D^{(a,a)}1\Big) + \frac{1}{2}\Big(1^\top D^{(b,a)}1 - 1^\top D^{(b,b)}1\Big)\\ &\geq n^2\Delta_{ab}^2 - (6+4\Delta_{ab})n^2\epsilon\end{aligned}$$

with probability $1-e^{-\Omega_{m,\Delta_{ab},\epsilon}(n)}$. To bound the second term in (52), note from

Lemma 23 that

$$\begin{aligned}
&\min(M^{(a,b)}1) \\
&\quad = \min\left(D^{(a,b)}1 - D^{(a,a)}1\right) + \frac{1}{2}\left(\frac{1}{n}1^\top D^{(a,a)}1 - \frac{1}{n}1^\top D^{(b,b)}1\right) \\
&\quad \leq \min\left(D^{(a,b)}1 - D^{(a,a)}1\right) \\
&\qquad\quad + \frac{1}{2}\left(\left|\frac{1}{n}1^\top D^{(a,a)}1 - 2n\mathbb{E}\|r\|_2^2\right| + \left|\frac{1}{n}1^\top D^{(b,b)}1 - 2n\mathbb{E}\|r\|_2^2\right|\right) \\
&\quad \leq \min\left(D^{(a,b)}1 - D^{(a,a)}1\right) + 4n\epsilon
\end{aligned}$$

with probability $1 - e^{-\Omega_{\Delta_{ab},\epsilon}(n)}$. Next, Lemma 22 gives

$$\min\left(D^{(a,b)}1 - D^{(a,a)}1\right) \leq n\Delta_{ab}^2 + (6 + 2\Delta_{ab})n\epsilon + 4n \min_{i\in\{1,\ldots,n\}} r_{a,i}^\top(\gamma_a - O_{ab}).$$

By assumption, we know $\|r\|_2 \geq 1 - \epsilon$ with positive probability regardless of $\epsilon > 0$. It then follows that

$$r^\top(\gamma_a - O_{ab}) \leq -\frac{\Delta_{ab}}{2} + \epsilon$$

with some (ϵ-dependent) positive probability. As such, we may conclude that

$$\min_{i\in\{1,\ldots,n\}} r_{a,i}^\top(\gamma_a - O_{ab}) \leq -\frac{\Delta_{ab}}{2} + \epsilon \qquad \text{w.p.} \qquad 1 - e^{-\Omega_{\mathcal{D},\epsilon}(n)}.$$

Combining these estimates then gives

$$\min(M^{(a,b)}1) \leq n\Delta_{ab}^2 - 2n\Delta_{ab} + (10 + 2\Delta_{ab})n\epsilon \qquad \text{w.p.} \qquad 1 - e^{-\Omega_{\mathcal{D},\Delta_{ab},\epsilon}(n)}.$$

Performing a union bound over a and b then gives

$$\min_{\substack{a,b\in\{1,\ldots,k\}\\ a\neq b}} \min(M^{(a,b)}1) \leq n\Delta^2 - 2n\Delta + (10 + 2\Delta)n\epsilon \qquad \text{w.p.} \qquad 1 - e^{-\Omega_{\mathcal{D},\gamma,\epsilon}(n)}.$$

Combining these estimates then gives the result. $\square$

Lemma 28. *Under the $(\mathcal{D}, \gamma, n)$-stochastic ball model, we have*

$$\|\Psi\|_{2\to 2} \leq \left(\frac{(1+\epsilon)\sigma}{\sqrt{m}} + \epsilon\right)\sqrt{N} \qquad w.p. \qquad 1 - e^{-\Omega_{m,k,\sigma,\epsilon}(n)},$$

where $\sigma^2 := \mathbb{E}\|r\|_2^2$ for $r \sim \mathcal{D}$.

Proof. Let R denote the matrix whose (a,i)th column is $r_{a,i}$. Then

$$\Psi = R - \Big[(c_1 - \gamma_1)1^\top \ \cdots \ (c_k - \gamma_k)1^\top\Big],$$

and so the triangle inequality gives

$$\begin{aligned} \|\Psi\|_{2\to 2} &\leq \|R\|_{2\to 2} + \Big\|\Big[(c_1 - \gamma_1)1^\top \ \cdots \ (c_k - \gamma_k)1^\top\Big]\Big\|_{2\to 2} \\ &\leq \|R\|_{2\to 2} + \left(n\sum_{a=1}^{k}\|c_a - \gamma_a\|_2^2\right)^{1/2}, \end{aligned}$$

where the last estimate passes to the Frobenius norm. For the first term, since $\mathcal{D}$ is rotation invariant, we may apply Theorem 5.41 in [49]:

$$\|R\|_{2\to 2} \leq (1+\epsilon)\sigma\sqrt{\frac{N}{m}} \qquad \text{w.p.} \qquad 1 - e^{-\Omega_{m,\sigma,\epsilon}(n)}.$$

For the second term, apply (41). The union bound then gives the result. $\square$

Proof of Theorem 11. First, we combine Lemmas 26, 27 and 28: For every $\delta > 0$, there exists an $\epsilon > 0$ such that

$$\begin{aligned}2\|\Psi\|_{2\to 2}^2 + \sum_{a=1}^{k}\sum_{b=a+1}^{k} \frac{\|P_{1^\perp}M^{(a,b)}1\|_2\|P_{1^\perp}M^{(b,a)}1\|_2}{\rho_{(a,b)}} \\ \leq 2\bigg(\frac{1+\epsilon}{\sqrt{m}}+\epsilon\bigg)^2 nk + \sum_{a=1}^{k}\sum_{b=a+1}^{k}\frac{4n^3\Delta_{ab}^2/m + Cn^3\epsilon}{n^2(\Delta_{ab}^2-\Delta(\Delta-2))-Cn^2\epsilon} \\ \leq n\bigg(\frac{2k}{m}+\frac{4}{m}\sum_{a=1}^{k}\sum_{b=a+1}^{k}\frac{\Delta_{ab}^2}{\Delta_{ab}^2-\Delta(\Delta-2)}+\delta\bigg) \qquad (53)\end{aligned}$$

with probability $1-e^{-\Omega_{\mathcal{D},\gamma,\epsilon}(n)}$. Next, the uniform bound $\Delta_{ab}\geq\Delta$ implies

$$\frac{\Delta_{ab}^2}{\Delta_{ab}^2-\Delta(\Delta-2)} = \frac{1}{1-\Delta(\Delta-2)/\Delta_{ab}^2} \leq \frac{1}{1-\Delta(\Delta-2)/\Delta^2} = \frac{\Delta}{2}.$$

Combining this with (53) and considering Lemma 25, it then suffices to have

$$\frac{2k}{m}+\frac{4}{m}\cdot\binom{k}{2}\cdot\frac{\Delta}{2} < \Delta(\Delta-2).$$

Rearranging then gives

$$\Delta > 2 + \frac{2k}{m\Delta} + \frac{k(k-1)}{m},$$

which is implied by the hypothesis since $\Delta \geq 2$. □

A.E Proof of Theorem 12

Put $g = \gamma/\|\gamma\|_2$ and let z have unit 2-norm. Since $\|\Phi_0^\top z\|_2 \geq \|\Phi_0^\top g\|_2$, then considering Lemma 13, it suffices to show that the containment

$$S_1 := \bigg\{v\in\mathbb{S}^{m-1} : |\langle g^\top v\rangle| \leq \frac{2}{\Delta}\bigg\} \subseteq \bigg\{v\in\mathbb{S}^{m-1} : \|\Phi_0^\top v\|_2 < \|\Phi_0^\top g\|_2\bigg\} =: S_2$$

holds with probability $1-e^{-\Omega_{m,\Delta}(N)}$. To this end, we will first show that each $v \in S_1$ is also a member of S_2 with high probability, and then we will perform a union bound over an ϵ-net of S_1.

We start by considering $\|\Phi^\top v\|_2$ and $\|\Phi^\top g\|_2$. Decompose x_i as either $\gamma + r_i$ or $-\gamma + r_i$ depending on whether x_i belongs to the ball centered at γ or $-\gamma$. Let w with $\|w\|_2 = 1$ be arbitrary. Then

$$(x_i^\top w)^2 = ((\pm\gamma + r_i)^\top w)^2 = (\pm\gamma^\top w + r_i^\top w)^2 = (\gamma^\top w)^2 \pm 2(\gamma^\top w)(r_i^\top w) + (r_i^\top w)^2,$$

and so $\mathbb{E}(x_i^\top w)^2 = (\gamma^\top w)^2 + \mathbb{E}(e_1^\top r)^2$. Linearity of expectation then gives

$$\mathbb{E}\big[(x_i^\top g)^2 - (x_i^\top v)^2\big] = (\gamma^\top g)^2 - (\gamma^\top v)^2 = \|\gamma\|^2(1-(g^\top v)^2) \geq 1 - \frac{4}{\Delta^2}.$$

Since $|(x_i^\top g)^2 - (x_i^\top v)^2| \leq 2(1+\Delta/2)^2$ almost surely, we may apply Hoeffding's inequality to get

$$\|\Phi^\top g\|_2^2 - \|\Phi^\top v\|_2^2 = \sum_{i=1}^N \Big((x_i^\top g)^2 - (x_i^\top v)^2\Big) \geq N\bigg(1-\frac{4}{\Delta^2}\bigg) - s \quad \text{w.p.} \quad 1-e^{-\Omega_\Delta(s^2/N)}. \tag{54}$$

For a properly chosen t, rearranging gives that $\|\Phi^\top v\|_2 < \|\Phi^\top g\|_2$. Instead, we will use (54) to prove the closely related inequality $\|\Phi_0^\top v\|_2 < \|\Phi_0^\top g\|_2$. Letting μ denote the centroid of the columns of Φ, we know by (41) that $\|\mu\|_2 \leq \delta$ with probability $1-e^{-\Omega_{m,\delta}(N)}$. In this event, every w with $\|w\|_2 = 1$ satisfies

$$\begin{aligned}\big|\|\Phi_0^\top w\|_2 - \|\Phi^\top w\|_2\big| &= \big|\|(\Phi + \mu 1^\top)^\top w\|_2 - \|\Phi^\top w\|_2\big| \\ &= \big|\|\Phi^\top w + 1\mu^\top w\|_2 - \|\Phi^\top w\|_2\big| \leq \|1\mu^\top w\|_2 \leq \sqrt{N}\delta. \end{aligned} \tag{55}$$

Furthermore,

$$\|\Phi_0^\top w\|_2 = \|(\Phi - \mu 1^\top)^\top w\|_2 \leq \|\Phi w\|_2 + \|1\mu^\top w\|_2 \leq \sqrt{N}\bigg(\frac{\Delta}{2} + 1 + \|\mu\|_2\bigg),$$

where the last inequality follows from Cauchy–Schwarz along with the fact that $\|x_i\|_2 \leq \Delta/2 + 1$ for every i. Taking a supremum over w then gives

$$\|\Phi_0^\top\|_{2\to 2} \leq \sqrt{N}\bigg(\frac{\Delta}{2} + 1 + \|\mu\|_2\bigg) \leq \sqrt{N}\bigg(\frac{\Delta}{2} + 1 + \delta\bigg) \quad \text{w.p.} \quad 1 - e^{-\Omega_{m,\delta}(N)}. \quad (56)$$

In (54), pick $s = (N/2)(1 - 4/\Delta^2) =: c_1(\Delta)N$. Then taking a union bound with (55) gives

$$\big(\|\Phi_0^\top v\|_2 - \sqrt{N}\delta\big)^2 \leq \|\Phi^\top v\|_2^2 \leq \|\Phi^\top g\|_2^2 c_1(\Delta)N \leq \big(\|\Phi_0^\top g\|_2 + \sqrt{N}\delta\big)^2 - c_1(\Delta)N$$

with probability $1 - e^{-\Omega_{m,\Delta,\delta}(N)}$. Expanding both sides and rearranging then gives

$$\begin{aligned} \|\Phi_0^\top v\|_2^2 &\leq \|\Phi_0^\top g\|_2^2 + 2\sqrt{N}\delta\big(\|\Phi_0^\top v\|_2 + \|\Phi_0^\top g\|_2\big) - c_1(\Delta)N \\ &\leq \|\Phi_0^\top g\|_2^2 - \underbrace{\bigg(c_1(\Delta) - 4\delta\bigg(\frac{\Delta}{2} + 1 + \delta\bigg)\bigg)}_{c_2(\Delta)} N, \end{aligned}$$

where the last step follows from (56). Thus, picking $\delta = \delta(\Delta)$ sufficiently small ensures $c_2(\Delta) > 0$. Since $c_2(\Delta)N \leq \|\Phi_0^\top g\|_2^2 - \|\Phi_0^\top v\|_2^2 = (\|\Phi_0^\top g\|_2 + \|\Phi_0^\top v\|_2)(\|\Phi_0^\top g\|_2 - \|\Phi_0^\top v\|_2)$, we further have

$$\|\Phi_0^\top g\|_2 - \|\Phi_0^\top v\|_2 \geq \frac{c_2(\Delta)N}{\|\Phi_0^\top g\|_2 + \|\Phi_0^\top v\|_2} \geq c_3(\Delta)\sqrt{N},$$

where the last inequality takes $c_3(\Delta) := c_2(\Delta)/(\Delta/2 + 1 + \delta)$, following (56).

At this point, we know that if $v \in S_1$, then $v \in S_2$ with probability $1 - e^{-\Omega_{m,\Delta}(N)}$.

It remains to perform a union bound over an ϵ-net of S_1 to conclude that $S_1 \subseteq S_2$ with high probability. To this end, pick $\epsilon < c_3(\Delta)/(\Delta/2+1+\delta)$, consider an ϵ-net $\mathcal{N}_\epsilon$ of S_1, and suppose

$$\|\Phi_0^\top v\|_2 \leq \|\Phi_0^\top g\|_2 - c_3(\Delta)\sqrt{N} \qquad \forall v \in \mathcal{N}_\epsilon. \tag{57}$$

Then for every $x \in S_1$, there exists $v \in \mathcal{N}_\epsilon$ such that $\|x-v\|_2 \leq \epsilon$, and so (56) gives

$$\begin{aligned} \|\Phi_0^\top x\|_2 &\leq \|\Phi_0^\top\|_{2\to 2}\|x-v\|_2 + \|\Phi_0^\top v\|_2 \\ &\leq \sqrt{N}\bigg(\frac{\Delta}{2}+1+\delta\bigg)\epsilon + \|\Phi_0^\top g\|_2 - c_3(\Delta)\sqrt{N} \\ &< \|\Phi_0^\top g\|_2, \end{aligned}$$

as desired. To measure the probability of the success event (57), a standard volume comparison argument establishes the existence of an ϵ-net of size $|\mathcal{N}_\epsilon| \leq (1+2/\epsilon)^m$; see Lemma 5.2 in [49]. As such, the union bound gives that (57) occurs with probability $1-e^{-\Omega_{m,\Delta}(N)}$.

A.F Proof of Theorem 16

Recall Lemma 15 demonstrated $\mathcal{W}$ serves as a dual certificate which may yield the integral solution. More precisely, we have yet to prove the dual certificate uniquely yields the integral solution $gg^\top$. Since g is in the nullspace of $\mathcal{W}-\mathbf{B}$ as a result of satisfying complementary slackness, it is sufficient to show the second smallest eigenvalue of this matrix denoted $\lambda_2(\mathcal{W}-\mathbf{B})$ is positive. Since $\mathcal{W}-\mathbf{B}$ is a random matrix, we instead demonstrate $\lambda_2(\mathcal{W}-\mathbf{B}) > 0$ with high probability. The strategy is to use Matrix Bernstein presented as Theorem 1.4 in [48] in our concentration argument of $\lambda_2(\mathbb{E}[\mathcal{W}-\mathbf{B}])$. Figure 11 is a graphical representation of how the proceeding lemmas

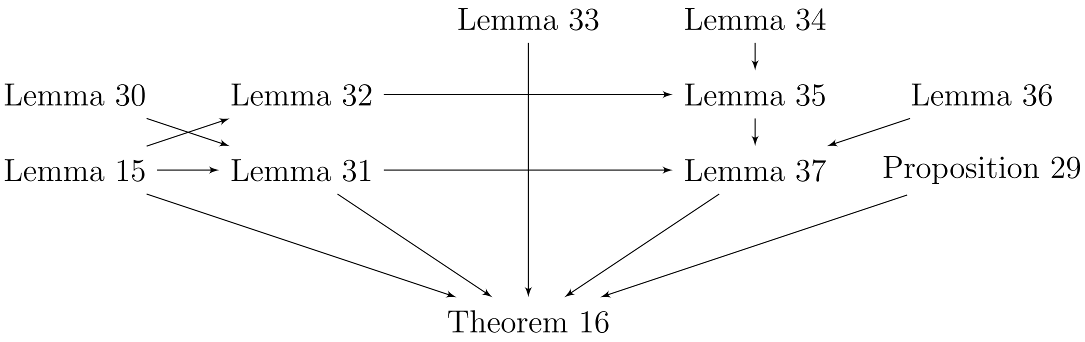


Figure 11. A directed graph to illustrate the development of the Lemmas in proving Theorem 16 where the predecessor of a statement is included in the statement's proof.

build to the theorem statement.

Proposition 29 ((Matrix Bernstein) Theorem 1.4 in [48]). *Consider a finite sequence* $\{\mathbf{X}_i\}$ *of independent, random, self-adjoint matrices with dimension* d*. Assume that each random matrix satisfies*

$$\mathbb{E}[\mathbf{X}_i] \quad \textit{and} \quad \lambda_{max}(\mathbf{X}_i) \leq \mathbf{R} \quad \textit{almost surely.}$$

Then, for all $t \geq 0$*,*

$$\Pr\left(\lambda_{max}\left(\sum_i \mathbf{X}_i\right) < t\right) \geq 1 - d \cdot \exp\left(\frac{-t^2/2}{\zeta + \mathbf{R}t/3}\right) \quad \textit{where} \quad \zeta := \left\|\sum_i \mathbb{E}(\mathbf{X}_i^2)\right\|.$$

Lemma 30. *Under the BiSBM used in Section 4.2, with* $p, q \in [0,1]$ *and* $p > q$*,*

$$\mathbb{E}[\mathcal{A}] = \frac{p-q}{2}(gg^\top - \Xi) + \frac{p+q}{2}(11^\top - \Theta)$$

where $\Xi := \sum_{i=1}^2 \sum_{j=1}^2 1_{U_i}1_{U_j}^\top + 1_{V_i}1_{V_j}^\top - \sum_{\substack{i,j=1 \\ i\neq j}}^2 1_{U_i}1_{U_j}^\top + 1_{V_i}1_{V_j}^\top$.

Proof. Per the definition of $\mathcal{A}$, we can write

$$\begin{aligned}
2\mathbb{E}[\mathcal{A}] &= 2p\sum_{i=1}^{2} 1_{U_i}1_{V_i}^\top + 1_{V_i}1_{U_i}^\top + 2q\sum_{\substack{i,j=1\\ i\neq j}}^{2} 1_{U_i}1_{V_j}^\top + 1_{V_i}1_{U_j}^\top \\
&= (2p-q+q)\sum_{i=1}^{2} 1_{U_i}1_{V_i}^\top + 1_{V_i}1_{U_i}^\top + (2q-p+p)\sum_{\substack{i,j=1\\ i\neq j}}^{2} 1_{U_i}1_{V_j}^\top + 1_{V_i}1_{U_j}^\top \\
&= (p-q)\left(\sum_{i=1}^{2} 1_{U_i}1_{V_i}^\top + 1_{V_i}1_{U_i}^\top - \sum_{\substack{i,j=1\\ i\neq j}}^{2} 1_{U_i}1_{V_j}^\top + 1_{V_i}1_{U_j}^\top\right) \\
&\quad + (p+q)\sum_{i,j=1}^{2} 1_{U_i}1_{V_i}^\top + 1_{V_i}1_{U_i}^\top. \qquad (58)
\end{aligned}$$

Notice we can express $gg^\top$ and $11^\top$ as

$$\begin{aligned}
gg^\top &= \sum_{i=1}^{2} 1_{U_i}1_{U_i}^\top + 1_{V_i}1_{V_i}^\top + 1_{U_i}1_{V_i}^\top + 1_{V_i}1_{U_i}^\top - \sum_{\substack{i,j=1\\ i\neq j}}^{2} 1_{U_i}1_{U_j}^\top + 1_{V_i}1_{V_j}^\top + 1_{U_i}1_{V_j}^\top + 1_{V_i}1_{U_j}^\top \\
&= \Xi + \sum_{i=1}^{2} 1_{U_i}1_{V_i}^\top + 1_{V_i}1_{U_i}^\top - \sum_{\substack{i,j=1\\ i\neq j}}^{2} 1_{U_i}1_{V_j}^\top + 1_{V_i}1_{U_j}^\top
\end{aligned}$$

and

$$11^\top = \sum_{i,j=1}^{2} 1_{U_i}1_{U_j}^\top + 1_{V_i}1_{V_j}^\top + 1_{U_i}1_{V_j}^\top + 1_{V_i}1_{U_j}^\top = \Theta + \sum_{i,j=1}^{2} 1_{U_i}1_{V_i}^\top + 1_{V_i}1_{U_i}^\top,$$

respectively. Substitution of $gg^\top - \Xi$ and $11^\top - \Theta$ into (58) and division by 2 gives the result. □

Lemma 31. *Under the BiSBM used in Section 4.2 with $\mathcal{W}$ being the dual certificate defined in Lemma 15, we can write*

$$\lambda_{min}(\mathbb{E}[\mathcal{W} - \mathbf{B}]_{span(g)^\perp}) = n\min(2p-1, 1-2q, p-q).$$

Proof. Employing Lemma 58, we calculate $\mathbb{E}[\mathcal{W} - \mathbf{B}]$:

$$\begin{aligned}\mathbb{E}[\mathcal{W} - \mathbf{B}] &= \mathbb{E}[2(\mathcal{G}_+ - \mathcal{G}_-)] - \mathbb{E}[2\mathcal{A} - 11^\top + \Theta] \\ &= n(p-q)I - \big((p+q)\mathbb{E}[11^\top - \Theta] + (p-q)\mathbb{E}[gg^\top - \Xi]\big) + 11^\top - \Theta \\ &= n(p-q)I - (1-p-q)(11^\top - \Theta) + (p-q)\Xi - (p-1)gg^\top.\end{aligned}$$

Notice the eigenvectors in $\text{span}(g)^\perp$ are spanned by $\{1, 1_U - 1_V, 1_{U_1} - 1_{U_2}, 1_{V_1} - 1_{V_2}\}$. Calculating the corresponding eigenvalues and taking the minimum concludes the proof. □

Lemma 32. *Define $b_{ij}(\cdot)$ to be a Bernoulli random variable with indices i, j and the input as the mean. Under the BiSBM used in Section 4.2 with $\mathcal{W}$ being the dual certificate defined in Lemma 15, we can write*

$$\mathcal{W} - \mathbb{E}[\mathcal{W}] = \sum_{i=1}^{2n} \sum_{j=1}^{n/2} \mathbf{X}_1(i,j) + \mathbf{X}_2(i,j)$$

where $\mathbf{X}_1(i,j) := 2e_ie_i^\top \Big(b_{ij}(p) - p\Big)$ *and* $\mathbf{X}_2(i,j) := 2e_ie_i^\top \Big(q - b_{ij}(q)\Big)$.

Proof. Consider $\mathcal{W} - \mathbb{E}[\mathcal{W}]$. By the definition of $\mathcal{W}$ and the linearity of expectation,

$$\begin{aligned}\mathcal{W} - \mathbb{E}[\mathcal{W}] &= 2(\mathcal{G}_+ - \mathcal{G}_-) - 2\mathbb{E}[\mathcal{G}_+ - \mathcal{G}_-] \\ &= 2(\mathcal{G}_+ - \mathbb{E}[\mathcal{G}_+]) - 2(\mathcal{G}_- - \mathbb{E}[\mathcal{G}_-]). \qquad (59)\end{aligned}$$

Observe

$$\mathbb{E}[\mathcal{G}_+] = \sum_{j=1}^{2n} \frac{np}{2} e_j e_j^\top \quad \text{and} \quad \mathbb{E}[\mathcal{G}_-] = \sum_{j=1}^{2n} \frac{nq}{2} e_j e_j^\top.$$

Using this observation and the definition of $\mathcal{G}_+$ and $\mathcal{G}_-$, we continue (59):

$$\begin{aligned}
\mathcal{W} - \mathbb{E}[\mathcal{W}] &= 2\sum_{i=1}^{2n} e_i e_i^\top \left(-\frac{np}{2} + \sum_{j=1}^{n/2} b_{ij}(p) \right) + 2\sum_{i=1}^{2n} e_i e_i^\top \left(\frac{nq}{2} - \sum_{j=1}^{n/2} b_{ij}(q) \right) \\
&= 2\sum_{i=1}^{2n}\sum_{j=1}^{n/2} e_i e_i^\top \Big(b_{ij}(p) - p \Big) + 2\sum_{i=1}^{2n}\sum_{j=1}^{n/2} e_i e_i^\top \Big(q - b_{ij}(q) \Big) \\
&= \sum_{i=1}^{2n}\sum_{j=1}^{n/2} 2 e_i e_i^\top \Big(b_{ij}(p) - p + q - b_{ij}(q) \Big) \\
&= \sum_{i=1}^{2n}\sum_{j=1}^{n/2} \mathbf{X}_1(i,j) + \mathbf{X}_2(i,j).
\end{aligned}$$

□

Lemma 33. *Under the BiSBM used in Section 4.2 with $\mathcal{W}$ being the dual certificate defined in Lemma 15, the following inequality holds:*

$$|\lambda_2(\mathcal{W} - \mathbf{B}) - \lambda_2(\mathbb{E}[\mathcal{W} - \mathbf{B}])| \leq ||\mathcal{W} - \mathbf{B} - \mathbb{E}[\mathcal{W} - \mathbf{B}]||_{2\to 2}.$$

Proof. Denote

$$\begin{aligned}
\mathcal{J} &:= (\mathcal{W} - \mathbf{B})_{span(g)^\perp} \\
\mathcal{J}' &:= \mathbb{E}[\mathcal{W} - \mathbf{B}]_{span(g)^\perp} \\
\eta &:= \max(||\mathcal{J}||_{2\to 2}^2, ||\mathcal{J}'||_{2\to 2}^2).
\end{aligned}$$

Now,

$$\begin{aligned}|\lambda_2(\mathcal{W}-\mathbf{B})-\lambda_2(\mathbb{E}[\mathcal{W}-\mathbf{B}])| &= |\lambda_{min}(\mathcal{J})-\lambda_{min}(\mathcal{J}')| \\ &= |\lambda_{min}(\mathcal{J})-\eta+\eta-\lambda_{min}(\mathcal{J}')| \\ &= \left| ||\eta I-\mathcal{J}'||_{2\to 2}^2 - ||\eta I-\mathcal{J}||_{2\to 2}^2 \right| \\ &\leq ||\eta I-\mathcal{J}'-(\eta I-\mathcal{J})||_{2\to 2}^2 \\ &= ||\mathcal{J}-\mathcal{J}'||_{2\to 2}^2.\end{aligned}$$

By Lemma 32, g is in the nullspace of $\mathbb{E}[\mathcal{W}-\mathbf{B}]$ and g is in the nullspace of $\mathcal{W}$ by construction. Since the spectral norm outputs the largest eigenvalue in magnitude, the inclusion of a zero eigenvalue will not change the spectral norm. Thus, $||\mathcal{J}-\mathcal{J}'||_{2\to 2}^2 = ||\mathcal{W}-\mathbf{B}-\mathbb{E}[\mathcal{W}-\mathbf{B}]||_{2\to 2}$. □

Lemma 34. *Define $b'_{ij}(x)$ to be a Bernoulli random variable with mean x and indices i, j. Under the BiSBM used in Section 4.2, we can write*

$$\begin{aligned}\mathbb{E}[\mathbf{B}]-\mathbf{B} = &\sum_{i=1}^{n/2}\sum_{j=n+1}^{3n/2}\mathbf{X}_3(i,j) + \sum_{i=n/2+1}^{n}\sum_{j=3n/2+1}^{2n}\mathbf{X}_3(i,j) \\ &+ \sum_{i=3n/2+1}^{2n}\sum_{j=n+1}^{3n/2}\mathbf{X}_4(i,j) + \sum_{i=n/2+1}^{n}\sum_{j=n+1}^{3n/2}\mathbf{X}_4(i,j)\end{aligned}$$

where $\mathbf{X}_3(i,j) := 2(e_i e_j^\top + e_j e_i^\top)(p - b'_{ij}(p))$ *and* $\mathbf{X}_4(i,j) := 2(e_i e_j^\top + e_j e_i^\top)(q - b'_{ij}(q))$

Proof. By the definition of $\mathbf{B}$,

$$\begin{aligned}\mathbb{E}[\mathbf{B}]-\mathbf{B} &= \mathbb{E}[2\mathcal{A} - 11^\top + \Theta] - 2\mathcal{A} + 11^\top - \Theta \\ &= 2(\mathbb{E}[\mathcal{A}] - \mathcal{A}).\end{aligned}$$

Per the definition of $\mathcal{A}$ and the way we shuffled the columns and rows by which partition a vertex lies, we can write

$$\mathcal{A} = \sum_{i=1}^{n/2} \sum_{j=n+1}^{3n/2} (e_i e_j^\top + e_j e_i \top) b'_{ij}(p) + \sum_{i=n+1}^{n} \sum_{j=3n/2+1}^{2n} (e_i e_j^\top + e_j e_i \top) b'_{ij}(p)$$
$$+ \sum_{i=3n/2+1}^{2n} \sum_{j=1}^{n/2} (e_i e_j^\top + e_j e_i^\top) b'_{ij}(q) + \sum_{i=n/2+1}^{n} \sum_{j=n+1}^{3n/2} (e_i e_j^\top + e_j e_i^\top) b'_{ij}(q).$$

Similarly, we can write

$$\mathbb{E}[\mathcal{A}] = \sum_{i=1}^{n/2} \sum_{j=n+1}^{3n/2} (e_i e_j^\top + e_j e_i^\top) p + \sum_{i=n+1}^{n} \sum_{j=3n/2+1}^{2n} (e_i e_j^\top + e_j e_i^\top) p$$
$$+ \sum_{i=3n/2+1}^{2n} \sum_{j=1}^{n/2} (e_i e_j^\top + e_j e_i^\top) q + \sum_{i=n/2+1}^{n} \sum_{j=n+1}^{3n/2} (e_i e_j^\top + e_j e_i^\top) q.$$

Combining these expressions gives the result. □

Lemma 35. *Consider the matrices $\mathbf{X}_l(i,j)$ defined in Lemmas 32 and 34. For all appropriate i, j, l*

$$\lambda_{max}\Big(\mathbf{X}_l(i,j)\Big) \leq 2p$$

Proof. By linearity of expectation, $\mathcal{W} - \mathbf{B} - \mathbb{E}[\mathcal{W} - \mathbf{B}] = \mathcal{W} - \mathbb{E}[\mathcal{W}] + \mathbb{E}[\mathbf{B}] - \mathbf{B}$. By Lemma 32, we have $\mathcal{W} - \mathbb{E}[\mathcal{W}]$ written as a sum of $\mathbf{X}_1(i,j)$ and $\mathbf{X}_2(i,j)$. Observe for $i \in [2n], j \in [n/2]$

$$\lambda_{max}(\mathbf{X}_1(i,j)) \leq 2(1-p) \quad \text{and} \quad \lambda_{max}(\mathbf{X}_2(i,j)) \leq 2q.$$

Similarly in 34, we have $\mathbb{E}[\mathbf{B}] - \mathbf{B}$ written as a sum of $\mathbf{X}_3(i,j)$ and $\mathbf{X}_4(i,j)$. Notice for $i, j \in [2n]$

$$\lambda_{max}(\mathbf{X}_3(i,j)) \leq 2p \quad \text{and} \quad \lambda_{max}(\mathbf{X}_4(i,j)) \leq 2q.$$

Taking the maximum of these upper bounds gives the result. □

Lemma 36. *Under the BisBM used in Section 4.2,*

$$\left\|\sum_{l}\sum_{ij}\mathbb{E}[\mathbf{X}_l^2(i,j)]\right\| = 2(n+2)\Big(p(1-p)+q(1-q)\Big).$$

Proof. We begin by calculating $\mathbb{E}[\mathbf{X}_l^2(i,j)]$ for $l \in [3]$. In the case of $l = 1$,

$$\begin{aligned}
\mathbb{E}[\mathbf{X}_1^2(i,j)] &= \mathbb{E}\left[\left(2e_ie_i^\top(b_{ij}(p)-p\right)^2\right] \\
&= 4\left(e_ie_i^\top\right)^2\mathbb{E}[(b_{ij}(p))^2 - 2pb_{ij}(p) + p^2] \\
&= 4e_ie_i^\top(p - 2p^2 + p^2) \\
&= 4e_ie_i^\top p(1-p).
\end{aligned}$$

A similar calculation shows $\mathbb{E}[\mathbf{X}_2^2(i,j)] = 4e_ie_i^\top q(1-q)$. Consider the case of $l = 3$.

$$\begin{aligned}
\mathbb{E}[\mathbf{X}_3^2(i,j)] &= \mathbb{E}\left[\left(2(e_ie_j^\top + e_je_i^\top)(p - b'_{ij}(p)\right)^2\right] \\
&= 4(e_ie_j^\top + e_je_i^\top)^2\mathbb{E}[(b'_{ij}(p))^2 - 2pb'_{ij}(p) + p^2] \\
&= 4(e_ie_i^\top + e_je_j^\top)(p - 2p^2 + p^2) \\
&= 4(e_ie_i^\top + e_je_j^\top)p(1-p).
\end{aligned}$$

A similar calculation shows $\mathbb{E}[\mathbf{X}_4^2(i,j)] = 4(e_ie_i^\top + e_je_j^\top)q(1-q)$. Notice $\sum \mathbb{E}[\mathbf{X}_l^2(i,j)]$

is a sum of diagonal matrices. Moreover,

$$\begin{aligned}
&\sum_{l}\sum_{ij}\mathbb{E}[\mathbf{X}_l^2(i,j)] \\
&\quad= \sum_{i=1}^{2n}\sum_{j=1}^{n/2}\mathbb{E}[\mathbf{X}_1^2(i,j)] + \mathbb{E}[\mathbf{X}_2^2(i,j)] + \sum_{i=1}^{n/2}\sum_{j=n+1}^{3n/2}\mathbb{E}[\mathbf{X}_3^2(i,j)] \\
&\qquad+ \sum_{i=n+1}^{n}\sum_{j=3n/2+1}^{2n}\mathbb{E}[\mathbf{X}_3^2(i,j)] + \sum_{i=3n/2+1}^{2n}\sum_{j=1}^{n/2}\mathbb{E}[\mathbf{X}_4^2(i,j)] + \sum_{i=n/2+1}^{n}\sum_{j=n+1}^{3n/2}\mathbb{E}[\mathbf{X}_4^2(i,j)] \\
&\quad= \sum_{i=1}^{2n}\sum_{j=1}^{n/2}4e_ie_i^\top\Big(p(1-p)+q(1-q)\Big) + \sum_{i=1}^{2n}e_ie_i^\top p(1-p) + \sum_{i=1}^{2n}e_ie_i^\top q(1-q) \\
&\quad= 2(n+2)\Big(p(1-p)+q(1-q)\Big)I.
\end{aligned}$$

The result follows from the sum being merely a multiple of the identity. ☐

Lemma 37. *Under the BisBM used in Section 4.2, with* $p > 1/2$ *and* $q < 1/2$*,*

$$\frac{t^2/2}{\zeta + \mathbf{R}t/3} = \Omega(n)$$

where

$$\begin{aligned}
t &:= \lambda_2(\mathbb{E}[\mathcal{W} - \mathbf{B}]) \\
\zeta &:= \left\|\sum_{l}\sum_{i,j}\mathbb{E}[\mathbf{X}_l^2(i,j)]\right\|.
\end{aligned}$$

and where $\mathbf{R}$ *is chosen to be a function solely of* p *satisfying* $\mathbf{R} \geq \lambda_{max}(\mathbf{X}_l(i,j))$ *for all appropriate* i, j, l.

Proof. First denote

$$\alpha := \min(2p-1, 1-2q, 2(p-q)) \quad \text{and} \quad \beta := 2\Big(p(1-p)+q(1-q)\Big).$$

By Lemma 31, we know $t = n\alpha$. Additionally, we may choose $\mathbf{R} := 2p$ by Lemma 35. Applying Lemma 36, we know

$$\frac{t^2/2}{\zeta + \mathbf{R}t/3} = \frac{n^2\alpha^2/2}{(n+2)\beta + 2pn\alpha/3} = n\frac{\alpha^2}{2(1+2/n)\beta + 4p\alpha/3}.$$

Since $n \geq 1$,

$$\frac{t^2/2}{\zeta + \mathbf{R}t/3} \geq n\frac{\alpha^2}{6\beta + 4p\alpha/3}$$

which is the result. $\square$

Proof of Theorem 16. By Lemma 15, we know using $\mathcal{W} = 2(\mathcal{G}_+ - \mathcal{G}_-)$ will certify $gg^\top$ as being optimal in the semidefinite relaxation. All that is left to show is $gg^\top$ as being the unique solution to (24) by demonstrating $\mathcal{W} - \mathbf{B}$ has a nullspace of dimension one. Denoting λ_2 to represent the second smallest eigenvalue, we must demonstrate $\lambda_2(\mathcal{W} - \mathbf{B}) > 0$. Since $\mathcal{W} - \mathbf{B}$ is random, we will utilize $\lambda_2(\mathbb{E}[\mathcal{W} - \mathbf{B}])$. By Lemma 31, we know $\lambda_2(\mathbb{E}[\mathcal{W} - \mathbf{B}]) > 0$ when $p > 1/2$ and $q < 1/2$. To demonstrate $\lambda_2(\mathcal{W} - \mathbf{B}) > 0$, it suffices to show

$$|\lambda_2(\mathcal{W} - \mathbf{B}) - \lambda_2(\mathbb{E}[\mathcal{W} - \mathbf{B}])| < \lambda_2(\mathcal{W} - \mathbf{B}).$$

By Lemma 33,

$$|\lambda_2(\mathcal{W} - \mathbf{B}) - \lambda_2(\mathbb{E}[\mathcal{W} - \mathbf{B}])| \leq \|\mathcal{W} - \mathbf{B} - \mathbb{E}[\mathcal{W} - \mathbf{B}]\|.$$

As such, it suffices to show

$$\|\mathcal{W} - \mathbf{B} - \mathbb{E}[\mathcal{W} - \mathbf{B}]\| < \lambda_2(\mathbb{E}[\mathcal{W} - \mathbf{B}]). \tag{60}$$

Using Proposition 29 and Lemma 37, we bound the probability of 60 occurring:

$$\Pr\Bigg(\|\mathcal{W} - \mathbf{B} - \mathbb{E}[\mathcal{W} - \mathbf{B}]\,\| < \lambda_2\left(\mathbb{E}[\mathcal{W} - \mathbf{B}]\right) \Bigg) \geq 1 - 2ne^{-\Omega(n)}. \qquad \square$$

A.G Proof of Theorem 17

Here we prove Theorem 17 using two versions of the Chernoff Bound.

Proposition 38 (Chernoff Bound)**.** *Let* $W_1, \ldots, W_n$ *be random variables such that* $0 \leq W_i \leq 1$ *for every* $i \in [n]$*. Let* $W := \sum_{i=1}^n W_i$ *and* $\mu := \mathbb{E}W$*. For any* $\epsilon \geq 0$*,*

$$\Pr\big(W \geq (1+\epsilon)\mu\big) \leq \exp\left(-\frac{\epsilon^2}{2+\epsilon}\mu\right)$$

and

$$\Pr\big(W \leq (1-\epsilon)\mu\big) \leq \exp\left(-\frac{\epsilon^2}{2}\mu\right).$$

For the proof of the first bound see [50], and for the proof of the second bound see [51].

Lemma 39. *If* $p, q \in [0,1]$*, then* $\mu_S \leq \mu_T$*. Moreover, if* $p \neq q$*, then* $\mu_S < \mu_T$*.*

Proof. We begin by calculating μ_S using linearity of expectation and properties of a Bernoulli random variable.

$$\begin{aligned}
\mu_S &= \mathbb{E}\left(\sum_{l=1}^m b_l(2p(1-p)) + \sum_{l'=1}^m b_{l'}(2q(1-q))\right) \\
&= \sum_{l=1}^m \mathbb{E}\left[b_l(2p(1-p))\right] + \sum_{l'=1}^m \mathbb{E}\left[b_{l'}(2q(1-q))\right] \\
&= 2mp(1-p) + 2mq(1-q) \\
&= 2m(p+q-p^2-q^2).
\end{aligned}$$

Now calculate μ_T.

$$\mu_T = \mathbb{E}\sum_{l=1}^{2m} b_l(p+q-2pq) = 2m(p+q-2pq).$$

Consider $\mu_T - \mu_S$.

$$\mu_T - \mu_S = 2m(p+q-2pq) - 2m(p+q-p^2-q^2) = 2m(p-q)^2 \geq 0$$

with equality when $p = q$. Thus, when $p \neq q$, $\mu_S < \mu_T$. □

Proof of Theorem 17. Let $p, q \in [0,1]$ such that $p \neq q$. Let $\epsilon > 0$ and $n \in \mathbb{N}$. Define $\alpha := p(1-p) + q(1-q)$ and $\beta := p + q - 2pq$. Let $m \in \mathbb{N}$ such that

$$m > \frac{2(\beta + 3\alpha)}{(\beta - \alpha)^2} \log\left(\frac{2n}{\epsilon}\right). \tag{61}$$

Notice we cannot allow $p = q$ as that would imply $\beta - \alpha = 0$. By an application of DeMorgan's law, for all $t > 0$, we have $\Pr(\mathcal{S} < \mathcal{T}) = 1 - \Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T})$. As such, showing $\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) < \epsilon$ for all $t > 0$ is sufficient to prove the result. Let $t > 0$ and consider the quantity $\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T})$. Taking the union bound, we obtain

$$\begin{aligned}\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) &\leq \sum_{\substack{i,j \in U_r \\ r \in [2] \\ i \neq j}} \Pr(\|x_i - x_j\|_2^2 \geq t) + \sum_{\substack{i \in U_1 \\ j \in U_2}} \Pr(\|x_i - x_j\|_2^2 \leq t) \\ &= \sum_{i=1}^{n(n-1)} \Pr(S \geq t) + \sum_{j=1}^{n^2} \Pr(T \leq t)\end{aligned}$$

as the maximum number of distinct intracluster distances per cluster is given by $\binom{n}{2}$ and the maximum number of distinct intercluster distances are given by n^2. Define

$\psi := t - \mu_S$ and $\phi := \mu_T - t$. As such,

$$\begin{aligned}
\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) &\leq \sum_{i=1}^{n(n-1)} \Pr(S \geq t) + \sum_{j=1}^{n^2} \Pr(T \leq t) \\
&= \sum_{i=1}^{n(n-1)} \Pr(S \geq \mu_S + \psi) + \sum_{j=1}^{n^2} \Pr(T \leq \mu_T - \phi) \\
&= n(n-1) \Pr(S \geq \mu_S + \psi) + n^2 \Pr(T \leq \mu_T - \phi) \\
&\leq n^2 \Pr(S \geq \mu_S + \psi) + n^2 \Pr(T \leq \mu_T - \phi) \\
&= n^2 \Pr\left(S \geq \left(1 + \frac{\psi}{\mu_S}\right) \mu_S\right) + n^2 \Pr\left(T \leq \left(1 - \frac{\phi}{\mu_T}\right) \mu_T\right). \qquad (62)
\end{aligned}$$

Since (62) holds for arbitrary t, we can redefine ψ and ϕ in terms of $\tau := \frac{\mu_S + \mu_T}{2}$ so that $\psi = \tau - \mu_S$ and $\phi = \mu_T - \tau$. Notice $\frac{\psi}{\mu_S} = \frac{\beta - \alpha}{2\alpha} > 0$ and $\frac{\phi}{\mu_T} = \frac{\beta - \alpha}{2\beta} > 0$ as Lemma 39 implies $\beta > \alpha > 0$. By Proposition 38,

$$\begin{aligned}
\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) &\leq n^2 \exp\left(-\left(\frac{\psi}{\mu_S}\right)^2 \frac{\mu_S}{2 + \psi/\mu_S}\right) + n^2 \exp\left(-\left(\frac{\phi}{\mu_T}\right)^2 \frac{\mu_T}{2}\right) \\
&\leq n^2 \exp\left(-\frac{\psi^2}{2\mu_S + \psi}\right) + n^2 \exp\left(-\frac{\phi^2}{2\mu_T}\right).
\end{aligned}$$

By Lemma 39, μ_S, μ_T, ψ, ϕ are expressible in terms of m, α, β. We make the bound in terms of m, α, and β.

$$\begin{aligned}
\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) &\leq n^2 \exp\left(-\frac{m^2(\beta - \alpha)^2}{4m\alpha + m(\beta - \alpha)}\right) + n^2 \exp\left(-\frac{m^2(\beta - \alpha)^2}{4m\beta}\right) \\
&\leq n^2 \exp\left(-m\frac{(\beta - \alpha)^2}{\beta + 3\alpha}\right) + n^2 \exp\left(-m\frac{(\beta - \alpha)^2}{4\beta}\right)
\end{aligned}$$

A quick calculation using the fact that $\beta > \alpha > 0$ will show

$$\exp\left(-m\frac{(\beta - \alpha)^2}{\beta + 3\alpha}\right) > \exp\left(-m\frac{(\beta - \alpha)^2}{4\beta}\right). \qquad (63)$$

Employing this inequality, we obtain

$$\begin{aligned}\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) &< 2n^2 \exp\left(-m\frac{(\beta-\alpha)^2}{\beta+3\alpha}\right)\\ &= 2\exp\left(2\log(n) - m\frac{(\beta-\alpha)^2}{\beta+3\alpha}\right).\end{aligned}$$

As such, it follows from (61) that

$$2\exp\left(2\log(n) - m\frac{(\beta-\alpha)^2}{\beta+3\alpha}\right) < \epsilon.$$

That is,

$$\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) < \epsilon.$$

Thus, $\Pr(\mathcal{S} < \mathcal{T}) = 1$ with high probability. □

A.H Proof of Theorem 18

of Theorem 18. Let $p, q \in [0,1]$ such that $p \neq q$. Let $\epsilon > 0$ and $n_1, n_2 \in \mathbb{N}$. Define $\alpha := p(1-p) + q(1-q)$ and $\beta := p + q - 2pq$ and $n_{max} = \max(n_1, n_2)$. Let $m \in \mathbb{N}$ such that

$$m > \frac{2(\beta+3\alpha)}{(\beta-\alpha)^2}\log\left(\frac{3n_{max}}{\epsilon}\right). \tag{64}$$

Notice we cannot allow $p = q$ as that would imply $\beta - \alpha = 0$. Just as in Theorem 17 showing $\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) < \epsilon$ for all $t > 0$ is sufficient to prove the result. Let $t > 0$ and consider the quantity $\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T})$. Taking the union bound, we obtain

$$\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) \leq \sum_{\substack{i,j \in U_r \\ r \in [2] \\ i \neq j}} \Pr(\|x_i - x_j\|_2^2 \geq t) + \sum_{\substack{i \in U_1 \\ j \in U_2}} \Pr(\|x_i - x_j\|_2^2 \leq t)$$
$$= \sum_{i=1}^{\binom{n_1}{2}+\binom{n_1}{2}} \Pr(S \geq t) + \sum_{j=1}^{n_1 n_2} \Pr(T \leq t)$$

as the maximum number of distinct intracluster distances per cluster is given by $\binom{n_1}{2} + \binom{n_1}{2}$ and the maximum number of distinct intercluster distances are given by $n_1 n_2$. Define $\psi := t - \mu_S$ and $\phi := \mu_T - t$. As such,

$$\begin{aligned}
\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) &\leq \sum_{i=1}^{\binom{n_1}{2}+\binom{n_1}{2}} \Pr(S \geq t) + \sum_{j=1}^{n_1 n_2} \Pr(T \leq t) \\
&= \sum_{i=1}^{\binom{n_1}{2}+\binom{n_1}{2}} \Pr(S \geq \mu_S + \psi) + \sum_{j=1}^{n^2} \Pr(T \leq \mu_T - \phi) \\
&= \left(\binom{n_1}{2} + \binom{n_1}{2}\right) \Pr(S \geq \mu_S + \psi) + n_1 n_2 \Pr(T \leq \mu_T - \phi) \\
&\leq \left(\frac{n_1(n_1-1)}{2} + \frac{n_2(n_2-1)}{2}\right) \Pr(S \geq \mu_S + \psi) \\
&\qquad + n_1 n_2 \Pr(T \leq \mu_T - \phi) \\
&\leq (n_1^2 + n_2^2) \Pr\left(S \geq \left(1 + \frac{\psi}{\mu_S}\right) \mu_S\right) \\
&\qquad + n_1 n_2 \Pr\left(T \leq \left(1 - \frac{\phi}{\mu_T}\right) \mu_T\right) \\
&\leq 2n_{max}^2 \Pr\left(S \geq \left(1 + \frac{\psi}{\mu_S}\right) \mu_S\right) \qquad (65) \\
&\qquad + n_{max}^2 \Pr\left(T \leq \left(1 - \frac{\phi}{\mu_T}\right) \mu_T\right).
\end{aligned}$$

Since (65) holds for arbitrary t, we can redefine ψ and ϕ in terms of $\tau := \frac{\mu_S + \mu_T}{2}$ so that $\psi = \tau - \mu_S$ and $\phi = \mu_T - \tau$. Observe the proof for Lemma (64) does not depend on $|U|$. Notice $\frac{\psi}{\mu_S} = \frac{\beta-\alpha}{2\alpha} > 0$ and $\frac{\phi}{\mu_T} = \frac{\beta-\alpha}{2\beta} > 0$ as Lemma 39 implies $\beta > \alpha > 0$. By Proposition 38,

$$\begin{aligned}\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) &\leq 2n_{max}^2 \exp\left(-\left(\frac{\psi}{\mu_S}\right)^2 \frac{\mu_S}{2+\psi/\mu_S}\right) + n_{max}^2 \exp\left(-\left(\frac{\phi}{\mu_T}\right)^2 \frac{\mu_T}{2}\right) \\ &\leq 2n_{max}^2 \exp\left(-\frac{\psi^2}{2\mu_S+\psi}\right) + n_{max}^2 \exp\left(-\frac{\phi^2}{2\mu_T}\right).\end{aligned}$$

By Lemma 39, μ_S, μ_T, ψ, ϕ are expressible in terms of m, α, β. We make the bound in terms of m, α, and β.

$$\begin{aligned}\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) &\leq 2n_{max}^2 \exp\left(-\frac{m^2(\beta-\alpha)^2}{4m\alpha + m(\beta-\alpha)}\right) + n_{max}^2 \exp\left(-\frac{m^2(\beta-\alpha)^2}{4m\beta}\right) \\ &\leq 2n_{max}^2 \exp\left(-m\frac{(\beta-\alpha)^2}{\beta+3\alpha}\right) + n_{max}^2 \exp\left(-m\frac{(\beta-\alpha)^2}{4\beta}\right)\end{aligned}$$

By using (63), we obtain

$$\begin{aligned}\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) &< 3n_{max}^2 \exp\left(-m\frac{(\beta-\alpha)^2}{\beta+3\alpha}\right) \\ &= 3\exp\left(2\log(n_{max}) - m\frac{(\beta-\alpha)^2}{\beta+3\alpha}\right).\end{aligned}$$

As such, it follows from (64) that

$$3\exp\left(2\log(n_{max}) - m\frac{(\beta-\alpha)^2}{\beta+3\alpha}\right) < \epsilon.$$

That is,

$$\Pr(\mathcal{S} \geq t \text{ or } t \geq \mathcal{T}) < \epsilon.$$

Thus, $\Pr(\mathcal{S} < \mathcal{T}) = 1$ with high probability. □

Appendix B. Modeling Methodology

B.A The Baseline Model

In this subsection, we describe how we create the baseline model given a set of patients and their medical histories. In evaluating the model we perform the procedure described in Section 5.3.

Following Section 5.3, we formulate our data as an adjacency matrix with the rows corresponding to patients and the columns corresponding to compressed three digit ICD-9 codings a patient was given in their medical history. Recall the patient partition given in [4] is given by

1. Healthy adults
2. Mothers and young children
3. Adults at risk of a chronic disease
4. Adults with a chronic disease
5. Adults with a rare condition
6. The disabled and the frail elderly

where we assign a patient to a the partition with the highest label number for which he or she is a member. As the partition follows a certain hierarchy, we identify a patient as being healthy if he or she is not in any of the other partitions. We identified the disabled by merely checking if they had any of the corresponding ICD-9-CM codes for any of the severe or chronic disabilities for which a person could seek Medicaid or Medicare coverage as described by the CFR Title 42, 435, 1010. We used the list of ICD-9 codes provided by the Texas Department of Aging and Disability Services for approved diagnostic codes following this federal regulation. Similarly, we identified the chronically ill as those having a chronic illnesses associated with a high probability of death as used and described in [52, 53]. The list of ICD-9 codes used in [52] are given by the Dartmouth Atlas of Health Care. Since there does not exist a comprehensive list of ICD-9 codes for all rare diseases, we restricted our attention to those with an ICD coding as given by Orpha.net. We used a web scraper to pull the ICD-10 codings of the rare diseases in their database to generate our list, which we present in Table 5. Since we require ICD-9 codes, we map our list of rare conditions in ICD-10 notation to ICD-9 notation. We use the 2014 general equivalency mapping provided by the Centers for Medicare and Medicaid Services, and present the translated ICD-9 codes in Table 6. In this case, as with all the other partitions, if there is a list of ICD codes which describe medical conditions required for membership in a partition, we compress the ICD code to three digits to allow for a direct comparison between the patient histories and the test for inclusion in a partition.

According to [54], frailty is a state of increased vulnerability to poor resolution of homeostasis following a stress, which increases the risk of adverse outcomes including falls, delirium and disability. The authors of [55] created a well cited model called the Rockwood Index that performed well in predicting frailty in older adults partially based on medical history. The Rockwood index uses a weighted sum of 70 variables, each being a possible condition that may reasonably occur to a frail elderly person. Only 50 of these variables have ICD codes associated with them, and as we only have access to a patients' medical history labeled in ICD coding, we restrict our modified Rockwood index using only these fifty variables. Those variables and their associated ICD codings that were used in the baseline model are listed in Tables 7 and 8. Motivated by the methodology of [55], we identify a patient as being frail elderly if the patient is at least 65 years old and has at least 36 of the physical deficits in our subset of the conditions in the Rockwood index.

Table 5. ICD-10 Codes for Rare Conditions

A05.1	A06.0	A77.1	A93.2	A96.0	B22.2	B34.0	B44.0	B44.1	B56.0
B57.0	B60.1	B60.1	B67.5	B76.0	B78.0	B81.0	B81.3	C15	C18.1
C22.0	C24.1	C38.0	C49	C53	C54	C56	C60.0	C63	C64
C73	C74	C74.0	C78.6	C83.3	C84.4	C88.3	C91.0	C91.5	C91.7
C92.0	C92.2	C92.4	C92.5	C92.6	C92.7	C94.0	C94.2	C94.3	C94.5
C94.7	C95.0	C96.0	C96.2	C96.5	D01.4	D12.6	D13.6	D27	D35.0
D36.0	D45	D46.7	D47.4	D48.1	D55.3	D56.0	D58.8	D59.0	D59.1
D60.0	D61.0	D64.0	D64.3	D65	D68.4	D69.0	D69.1	D69.2	D69.3
D69.4	D69.8	D70	D72.8	D74.0	D75.1	D80.0	D80.5	D81.0	D81.3
D82.2	D82.4	D82.8	D84.8	E03.1	E16.1	E20.1	E20.8	E22.0	E23.0
E23.2	E23.6	E24.0	E24.3	E24.8	E25.0	E25.8	E26.1	E26.8	E27.1
E27.2	E27.4	E28.8	E30.1	E31.0	E31.8	E34.5	E34.8	E70.2	E70.3
E71.1	E71.3	E72.0	E72.1	E72.2	E72.3	E72.5	E72.8	E74.0	E74.4
E75.0	E75.1	E75.2	E75.4	E76.0	E76.2	E77.1	E77.8	E78.6	E79.8
E80.2	E80.3	E80.5	E83.1	E83.2	E83.3	E83.4	E83.5	E85	E85.0
E85.1	E85.3	E85.4	E85.8	E88.0	E88.1	E88.2	E88.8	F84.1	G04.0
G11	G11.0	G11.1	G11.2	G11.3	G11.4	G11.8	G12.1	G12.2	G20
G23.0	G23.1	G24.1	G24.8	G31.0	G31.8	G37.3	G37.4	G40.3	G40.4
G40.5	G40.8	G54.0	G54.5	G58.0	G60.0	G60.1	G60.2	G60.8	G61.0
G61.8	G70.0	G70.8	G71.0	G71.1	G71.2	G71.3	G71.8	G72.3	G72.4
G72.8	G93.0	G95.0	G98	H16.2	H16.8	H18.5	H20.0	H26.0	H31.2
H33.0	H35.0	H35.5	H44.0	H47.2	H49.4	H53.5	H83.3	H90.3	H90.5
I00	I15.1	I42.8	I45.5	I45.6	I60.9	I71.0	I73.8	I74.4	J61
J80	J84.0	J84.1	J98.0	K00.0	K00.2	K00.5	K03.5	K06.1	K07.1
K22.0	K52.8	K60.3	K72.0	K75.4	K76.8	K86.1	L12.3	L40.2	L43.8
L56.4	L60.3	L63.0	L63.1	L65.8	L66.4	L67.8	L68.1	L81.4	L81.7
L81.8	L85.0	L93.2	L98.2	L98.5	L98.8	M02.8	M06.1	M31.0	M31.3
M31.8	M32.8	M35.8	M60.8	M62.8	M79.2	M87.0	M89.0	M89.5	M91.1
M92.4	M93.2	M93.9	N10	N25.8	N85.6	O26.6	P28.4	P35.8	Q01.0
Q02	Q04.0	Q04.2	Q04.3	Q07.0	Q10.4	Q11.0	Q11.2	Q12.0	Q13.1
Q13.8	Q16.0	Q20.8	Q21.1	Q21.2	Q21.4	Q22.2	Q22.8	Q23	Q23.0
Q23.8	Q24.5	Q25.1	Q25.4	Q25.7	Q25.8	Q26.8	Q27.3	Q30.1	Q30.8
Q39.5	Q41.0	Q43.1	Q44.2	Q44.6	Q44.7	Q45.1	Q45.3	Q51.0	Q55.5
Q61.1	Q61.5	Q64.3	Q68.8	Q71.0	Q71.3	Q72.0	Q72.3	Q72.5	Q73.0
Q73.8	Q74.0	Q74.3	Q74.8	Q75.0	Q75.1	Q75.4	Q75.8	Q76.3	Q76.4
Q76.8	Q77.0	Q77.2	Q77.3	Q77.4	Q77.5	Q77.7	Q77.8	Q78.0	Q78.2
Q78.8	Q79.4	Q79.6	Q79.8	Q80.2	Q80.3	Q80.8	Q81.2	Q82.0	Q82.4
Q82.5	Q82.8	Q83.1	Q84.2	Q84.3	Q84.6	Q84.8	Q86.0	Q86.8	Q87.0
Q87.1	Q87.2	Q87.3	Q87.4	Q87.5	Q87.8	Q89.7	Q93.0	Q93.5	Q99.8
R82.1	T37.5	T40.0	T43.0	T50.4	T56.8	T62.2	T78.3	T86.0	Y43.2

Table 6. ICD-9 Codes for Rare Conditions

005.1	006.0	082.1	066.1	078.7	079.0	117.3	484.6	117.3	086.3
086.0	122.5	126.0	126.2	126.3	126.8	127.2	127.1	127.7	153.5
155.0	156.2	164.1	187.1	193	197.6	203.80	202.50	202.60	277.89
211.3	211.6	229.0	207.10	207.11	207.12	238.4	289.89	238.1	282.3
282.43	282.8	283.0	283.0	28.481	285.0	285.0	286.6	286.7	287.0
287.1	287.2	287.31	287.39	287.8	289.7	289.0	279.04	279.05	279.2
277.2	279.8	279.8	279.8	279.3	243	251.1	275.49	252.1	253.0
253.2	253.3	628.1	253.5	253.8	255.0	255.0	255.0	255.2	255.2
255.14	255.41	255.41	256.8	259.1	258.1	258.8	259.8	270.7	270.7
270.8	271.0	271.8	330.1	271.8	272.7	272.7	272.5	277.2	277.89
277.4	686.8	277.31	277.39	277.39	277.39	272.6	272.8	334.2	334.0
334.3	334.2	334.8	334.1	334.8	335.11	332.0	333.0	333.0	333.6
337.9	341.20	341.21	323.9	341.8	353.0	353.5	354.8	356.0	356.1
356.2	356.3	356.0	356.8	357.0	359.1	359.0	359.89	359.89	359.3
348.0	336.0	370.8	389.11	389.12	389.14	389.18	389.10	389.14	389.16
390	405.91	425.2	425.4	426.6	426.7	426.81	430	444.22	501
518.82	520.0	520.2	520.5	521.6	523.8	530.0	565.1	571.42	577.1
577.8	696.1	697.0	692.82	703.8	704.09	704.09	704.09	701.8	704.2
704.1	709.09	709.09	709.09	701.1	695.4	695.89	701.8	709.8	714.2
447.5	710.0	710.5	710.8	729.2	590.10	590.11	621.5	770.82	771.2
742.0	742.1	742.2	742.2	742.2	743.65	743.03	743.06	743.10	743.30
743.45	744.01	745.19	745.7	745.8	745.5	745.60	745.61	745.69	745.8
746.09	746.1	746.3	746.89	746.85	747.10	747.20	747.21	747.29	747.29
747.49	748.1	748.1	750.4	750.7	751.1	751.3	751.61	751.62	751.60
751.69	751.7	751.7	752.31	752.49	752.69	753.16	753.17	755.51	755.52
756.9	755.4	755.4	755.53	755.54	755.56	755.57	755.59	754.89	755.8
756.0	756.0	756.0	756.0	754.2	756.3	756.4	756.3	756.59	756.4
756.4	756.4	756.4	756.51	756.52	756.53	756.59	75.671	756.83	756.81
756.82	756.89	756.9	757.1	757.1	757.1	757.39	757.0	757.31	757.32
757.2	757.39	757.6	757.4	757.5	757.5	757.8	760.71	760.79	755.55
756.0	759.81	759.89	759.89	759.89	759.7	758.5	758.39	758.81	758.89
791.3									

Table 7. The Items in the Modified Rockwood Index with Corresponding ICD-9 codes, Part I

Description of physical deficit used in modified Rockwood model	ICD-9 Coding Used
History of thyroid disease	246
Thyroid problems	246
History of diabetes mellitus	250
Mood problems	296
Depression (clinical impression)	296.82 296.20-296.26 296.30-296.36
Paranoid features	297.9
Restlessness	307.9
Feeling sad, blue, depressed	311
History relevant to cognitive impairment or loss	331
Family history relevant to cognitive impairment or loss	331
History of Parkinsons disease	332
Tremor at rest	333.1
Postural tremor	333.1
Intention tremor	333.1
Seizures, generalized	345
Seizures, partial complex	345.4 345.5
Arterial hypertension	401.9
Myocardial infarction	410
Arrhythmia	427
Congestive heart failure	428
Cardiac problems	429
History of stroke	434.91
Cerebrovascular problems	437
Lung Problems	518
Gastrointestinal problems	520-579
Breast problems	610-612
Skin problems	680-686
Skin problems	690-698
Skin problems	700-709
Musculoskeletal problems	710-739
Clouding or delirium	780.09
Syncope or blackouts	780.2
Sleep changes	780.5
Tiredness all the time	780.7 780.71 780.79

Table 8. The Items in the Modified Rockwood Index with Corresponding ICD-9 codes, Part I

Description of physical deficit used in modified Rockwood model	ICD-9 Coding Used
Memory changes	780.93
Short-term memory impairment	780.93
Long-term memory impairment	780.93
Bradykinesia	781
Irregular gait pattern	781.2
Poor limb coordination	781.3
Poor coordination, trunk	781.3
Poor muscle tone in limbs	781.99
Poor standing posture	781.99
Impaired vibration	782
Head and Neck Problems	784
Headache	784
Peripheral pulses	785.9
Respiratory problems	786
Urinary incontinence	788.3
Abdominal problems	789
Presence of snout reflex	796.4
Falls	E800-E888
Family history of degenerative disease	V17.2
Impaired mobility	V49.89

Due to the nature of the dataset, we cannot directly identify if a female is a mother. As such we create a reasonable set of properties a mother would satisfy that are accessible in the data. The properties we used are as follows:

- The person is female.
- There is a child with the same family ID
- The person is at least 10 years older than the eldest child in the family.

The reader may notice the odd minimum age for a mother. According to [37], there exist live births for mothers between the ages of 10 and 15 at a rate of 0.6 per 1000 births in 2005. Due to the large size of the dataset, it may be possible that a teen pregnancy within this age range may occur and so we account for this in the minimum age of a mother. To identify a person as a mother, there necessarily has to be a child in the database. As such, we identify any person under the age of 18 as being a child as that is the age range for a pediatricians' practice. Furthermore, for purposes of simplicity we label a person a child even if he or she becomes an adult during the five year span.

Lastly, we are left with identifying people at risk of chronic conditions, and to create such a classifier is not trivial. We can easily identify those with chronic conditions and those without such conditions. Given the data, we cannot identify if a person is at risk of a chronic condition. For example, we know if a person engages in behaviors such as heavy smoking and severe lack of physical activity, a person is more likely to suffer a chronic condition. We need some sort of machinery to predict if a person is likely to get a chronic condition based on their medical history. Work has been done to this extent in [56, 46, 57] where different recommender systems are devised that output a list of conditions a person may get in the future based on their medical histories. These recommender systems, however, are too computationally expensive for our purposes. As such, we devise our own Principal Component based classifier.

The idea behind our method is to find a direction where the variance of the chronically diseased is small and the variance of the non-chronically diseased is high. In so doing we hope a clustering algorithm will identify all of the chronically diseased and some of the non-chronically diseased in one cluster and all other patients in another cluster. In other words, the first cluster will be the patients who either have a chronic condition or will likely get chronic condition, and the second cluster consists of people who are not at risk of a chronic condition. As such, we wish to find a direction $\widetilde{v}$ that maximizes $\widetilde{v}^\top \mathcal{M}_0^\top \mathcal{M}_0 \widetilde{v}$ and minimize $\widetilde{v}^\top \mathcal{D}_0^\top \mathcal{D}_0 \widetilde{v}$ where $\mathcal{M}_0$ denotes the centered matrix of patients and their medical histories and $\mathcal{D}_0$ is a centered matrix of chronically ill patients and their medical histories. As such, we can formulate our problem as a regularization:

$$\begin{aligned} &\text{maximize} && \widetilde{v}^\top(\mathcal{M}_0^\top \mathcal{M}_0 - \xi \mathcal{D}_0^\top \mathcal{D}_0)\widetilde{v} \\ &\text{subject to} && \|\widetilde{v}\|_2 = 1 \end{aligned}$$

with ξ being a regularization parameter. Assuming a ξ is picked, our classifier is as listed in Algorithm 3.

We briefly describe how we selected the value of ξ for our model. We considered values of $\xi \in \{2^{-10}, 2^{-9}, \ldots, 2^{20}\}$ and for each value of ξ, we attempt to create a classifier with good cross-validation results in identifying the chronically ill as being in the at-risk category. As we wish to perform a cross-validation, we randomly choose our hold out set of chronically diseased patients. As such, roughly ten classifiers were created for each value of ξ where each classifier was trained under a different randomly selected training set. We select a value of $\xi = 2^6$ as it gave the most promising classification accuracy (in identifying the chronically ill as being in the at-risk category) of 60% or higher. In selecting a classifier for the model, we additionally

Algorithm 3: A Classifier for being at risk of a chronic condition

```
Input: Centered matrices $\mathcal{M}_0, \mathcal{D}_0$ as defined in the text with patients in the
       training set,
        a regularization parameter $\xi$,
        and an unclassified patient's medical history $h$ which is centered on
        the mean of $\mathcal{M}_0$
Output: Classification f the patient as "at risk" or "not at risk" of a chronic
        condition
Compute the leading eigenvector $v$ of $\mathcal{M}_0^\top \mathcal{M}_0 - \xi \mathcal{D}_0^\top \mathcal{D}_0$
Apply $k$-means to the rows of $\mathcal{M}_0 v$ to find two clusters and their centroids
Identify the cluster with the majority of the chronically diseased as the
    "at risk" cluster
Denote the cluster centroids $c_{\text{at risk}}$ and $c_{\text{not at risk}}$, apropriately
Find the vector $\hat{\Delta} := c_{\text{at risk}} - c_{\text{notat risk}}$
if $\hat{\Delta} h > 0$ then
    Classify patient as "at risk"
else
    Classify patient as "not at risk"
end
```

impose another constraint: the classifier should place more non-chronically ill people in the not at risk category than in the at risk category. We admit that we are imposing a bias here, but we wish our classifier to be slightly more discerning than a classifier that says more than half of the population that does not have a chronic condition is at risk for one.

We select the best classifier for our particular choice of ξ in the following manner. We run 30 trials of cross-validation where in each trial we randomly select a third of the chronically ill and a third of the non-chronically ill for the hold out set. Furthermore, we enforce each trial to satisfy the mentioned condition on the non-chronically ill. So if a trial does not satisfy this condition, we do not retain the results of the trial. We take the classifier that performed the best under cross validation (classification accuracy of 70.61%) to identify the people at risk.

B.B Important Code Used in Modeling

Here we present the code that was used in Chapter V. The reader may notice that in certain instances a line ends prematurely such that if the code were to run “as is” an error would result. Due to some of the lines exceeding margins, we merely continue the line of code on a new line here in thesis. Where these breaks occur should be fairly evident (every line break of this type is not followed by a semi-colon).

Code for creating the Baseline model.

In this subsection we present the code used in creating the Baseline model.

create_baseline.

The following MATLAB code creates the baseline model using the first five years.

```
filename4adjmatrix= 'I:\My Documents\MATLAB\thesis\Create Baseline
    \WORKSPACE DATAFIRSTCOLLAPSEDsparse_Adjmatrix.mat';
% 'I:\My Documents\MATLAB\thesis\Create Baseline\WORKSPACE
    DATASECONDCOLLAPSEDsparse_Adjmatrix.mat';
% function [IDX,entropy,crossvalidationresults]=create_baseline(
    filename4adjmatrix)
%initialize      -------------------------
loaddata=load('I:\My Documents\MATLAB\thesis\Create Baseline\
    WORKSPACE DATA\cleaned_data_4baseline_CA029SEP.mat');
AGE=loaddata.AGE;
PERSONID=loaddata.PERSONID;
FAMILYID=loaddata.FAMILYID;
N=max(size(AGE));
IDX=zeros(N,8);
```

```
[All_patients,ia_P,ic_P]    =unique(PERSONID,'stable');
                    %Note:  all_patients =  [patient(ia_p)
                    %Note:  patient      =  all_patients(ic_p)


loadtemp=load('I:\My Documents\MATLAB\thesis\Create Baseline
    \WORKSPACE DATA\unique_CLPSDDX.mat');
unique_CLPSDDX=loadtemp.unique_CLPSDDX;
loadtemp=load(filename4adjmatrix);
sparse_Adjmatrix=loadtemp.FIRST_COLLAPSEDsparse_Adjmatrix;
% adjmatrix=loadadj.SECOND_COLLAPSEDsparse_Adjmatrix;


filename4results='I:\My Documents\MATLAB\thesis\Create Baseline\
    WORKSPACE DATA\Baseline_results.mat';
%get frail elderly                                 --------------
[frail,frail_ill_in_adj]=get_frail(unique_CLPSDDX,ic_P,
    sparse_Adjmatrix,AGE);   %created&works
%label as frail elderly/disabled (col: 8, val 6)
IDX(frail,8)=6;
save(filename4results,...
    'IDX','-append')
clear frail
%get disabled                                      --------------
[disabled,disabled_ill_in_adj]=get_disable(unique_CLPSDDX,ic_P,
    sparse_Adjmatrix);  %created&works
%label as frail elderly/disabled (col: 7, val 6)
IDX(disabled,7)=6;
```

```
save(filename4results,...
    'IDX','-append')
clear disabled

%get adults with rare conditions                       --------------
DX1=loaddata.DX1;
DX2=loaddata.DX2;
DX3=loaddata.DX3;
DX4=loaddata.DX4;
[rare_ppl,rare_ill_in_adj]=get_rare_ppl_new(All_patients,DX1,DX2,DX3,
    DX4,PERSONID); %created
%label adults with rare conditions (col: 6, val 5)
IDX(rare_ppl,6)=5;
save(filename4results,...
    'IDX','-append')
clear rare_ppl DX1 DX2 DX3 DX4

%get chronically ill adults                            --------------
[chronic_ill,chronic_ill_in_adj]=get_chronic_adults(unique_CLPSDDX,
    ic_P,sparse_Adjmatrix); %created&works
%label chronically ill adults (col: 5, val 4)
IDX(chronic_ill,5)=4;
save(filename4results,...
    'IDX','-append')
clear chronic_ill
```

```
%get adults at risk of devoloping chronic diseases --------------
trials4Xvalidation=30;
rowsnot2include=union(frail_ill_in_adj,disabled_ill_in_adj);
rowsnot2include=union(rowsnot2include,rare_ill_in_adj);
rowsnot2include=union(rowsnot2include,chronic_ill_in_adj);
rows2include=setdiff(1:size(sparse_Adjmatrix,1),rowsnot2include);
clear rowsnot2include
[cols4chronic,whochronic,whoNOTchronic,Ac_PSD,Dc_PSD,Ac,Dc]=
    setup4getrisk(rows2include,unique_CLPSDDX,sparse_Adjmatrix);
fprintf('chronic_risk: setup complete\n')
[chronic_risk,crossvalidationresults,common_sensecheck,where_cmn_sns]=
    get_chronic_risk_PCAmotivatedwcrossvalidation(whochronic,
        whoNOTchronic,Ac,Dc,trials4Xvalidation);%created
fprintf('chronic_risk: cross validation complete\n')

save(filename4results,...
    'IDX',...
    'crossvalidationresults',...
    'common_sensecheck',...
    'where_cmn_sns','-append');
fprintf('chronic_risk: get chronic risk complete\n')
%label as being at risk of developing chronic diseases (col: 4, val 3)
IDX(chronic_risk,4)=3;
clear chronic_risk
fprintf('chronic_risk complete\n')
```

```
%get children and adults                              ---------------
[children,~]=get_children(PERSONID,AGE);              %created&works
%label children (col: 3, val 2)
IDX(children,3)=2;
save(filename4results,...
    'IDX','-append')

fprintf('children complete\n')


%get mothers                                          --------------
GENDER=loaddata.GENDER;
YEAR=loaddata.YEAR;
[eldest_child_ages]=get_mothers_precompute(children,AGE,PERSONID,
    FAMILYID,YEAR);
mothers=get_mothers_new(children,AGE,PERSONID,FAMILYID,GENDER,YEAR,
    eldest_child_ages);
%label mothers (col: 2, val 2)
IDX(mothers,2)=2;
save(filename4results,...
    'IDX','-append')
clear mothers
clear children
clear GENDER YEAR FAMILYID
fprintf('mothers complete\n')

%get all other adults not already labeled
```

```
%see logical indexing in labeling of IDX                %created&works
%label as healthy adults (col: 1, val 1)
IDX(~any(IDX,2),1)=1;
save(filename4results,...
    'IDX','-append')
clear healthy
fprintf('healthy complete\n')

%calculate entropy from labels/IDX                  --------------
    %change format from hospital visits to unique patients
clear AGE
clear eldest_child_ages

maxedIDX=max(IDX,[],2);
save(filename4results,...
    'maxedIDX','-append')

clear IDX

maxedIDX4Adj=zeros(size(All_patients,1),1);
parfor i=1:size(All_patients,1)
    maxedIDX4Adj(i)=max(maxedIDX(All_patients(i)==PERSONID));
    disp(i)
end
save(filename4results,...
    'maxedIDX4Adj','-append')
```

```
fprintf('maxedIDX4Adj complete\n')



clear sparse_Adjmatrix


YEAR=loaddata.YEAR;
secondhalfpersonid=zeros(max(size(PERSONID)),1);
secondhalfpersonid(YEAR>=2008)=PERSONID(YEAR>=2008);
num_visits=zeros(size(All_patients,1),1);
parfor i=1:size(All_patients,1)
    num_visits(i)=sum(All_patients(i)==secondhalfpersonid);
end
save(filename4results,...
    'num_visits','-append')
fprintf('num_visits complete\n')
```

get_frail.

The following MATLAB code is designed to find the frail elderly in the population.

```
%--------------
%Author: Takayuki Iguchi
%Last Modified: 14 Dec 2015
%Program Description:
%     Finds the frail elderly in a population.
%Input:
%     -all_diagnosis : A list of unique 3 digit ICD codings that occur
```

```
%     in the dataset.
%     -ic_p: An output from the unique command applied to PERSONID.
%     This array orders a unique list of patients according to hospital
%     visits as ordered out in the dataset.
%     -sparse_Adjmatrix : An N x m sparse array where rows correspond
%     to patients and columns correspond to a 3 digit ICD coding. An
%     entry in this array corresponds to the number of times a patient
%     has been given an ICD coding.
%     -AGE : A list of patient ages organized by hospital visit.
%Output:
%     -frail: A list of the frail elderly organized by hospital visit.
%     -frail_ill_in_adj: A list of those identified as being frail
%     (not taking into account age) per the modified Rockwood index.
%--------------
function [frail,frail_ill_in_adj]=get_frail(all_diagnosis,ic_p,
    sparse_Adjmatrix,AGE)

%load 3 digit ICD codings from modified Rockwood index
stuff=load(’I:\My Documents\MATLAB\thesis\Create Baseline\
    Frail Elderly\CLPSDhierarchical_subset_of_Rockwood_index1.mat’);
list=stuff.hierarchicalsubsetofRockwoodindex1;
[˜,L]=size(list);
list(end+1,:)={’’};

patients_score=zeros(size(sparse_Adjmatrix,1),1);
percent=50/70;
```

```
min_age=65;
for i=1:L

    %initialize while loop
    cont=true;
    codes4indexhit=find(strcmp(list(:,i),''),1,'first')-1;
    cols4frail=zeros(L,1);
    patients_hit=zeros(size(sparse_Adjmatrix,1),1);
    j=1;
    while cont
        temp=list{j,i};

        %get column of a Rockwood index condition in adj matrix
        if isnumeric(temp)
            temp=num2str(temp);
            temp=find(strcmp(temp,all_diagnosis));
        else
            temp=find(strcmp(temp,all_diagnosis));
        end



        if numel(temp)~=0
            cols4frail=temp(1);
            cols4frail=cols4frail(cols4frail~=0);
            patients_hit=patients_hit+any(sparse_Adjmatrix(:,
                cols4frail),2);
```

```
        end

        if j>=codes4indexhit
            cont=false;
        else
            j=j+1;
        end
    end
    patients_score=patients_score+any(patients_hit,2);
end

%find the frail people in the adj matrix
frail_ill_in_adj=(patients_score>=percent*L);
frail_ill_in_adj=find(frail_ill_in_adj==1);

%find the frail elderly organized by hospital visits
frail=frail_ill_in_adj(ic_p);
age_requirement=(AGE>=min_age);
frail=age_requirement.*frail;
frail=find(frail==1);

end
```

get_disable.

The following MATLAB code is designed to find the patients with disabilities in a population.

```
%--------------
%Author: Takayuki Iguchi
%Last Modified: 14 Dec 2015
%Program Description:
%     Finds the disabled people in a population.
%Input:
%     -all_diagnosis : A list of unique 3 digit ICD codings that occur
%     in the dataset.
%     -ic_p: An output from the unique command applied to PERSONID.
%     This array orders a unique list of patients according to hospital
%     visits as ordered out in the dataset.
%     -sparse_Adjmatrix : An N x m sparse array where rows correspond
%     to patients and columns correspond to a 3 digit ICD coding. An
%     entry in this array corresponds to the number of times a patient
%     has beengiven an ICD coding.
%Output:
%     -disabled: A list of the frail elderly organized by hospital
%     visit.
%     -disabled_ill_in_adj: A list of those identified as being
%     disabled in the adjacency matrix.
%--------------


function [disabled,disabled_ill_in_adj]=get_disable(all_diagnosis,ic_p,
```

```
    sparse_Adjmatrix)

%load ICD codings for disabilites
stuff=load('I:\My Documents\MATLAB\thesis\Create Baseline\
    Disability codes\DiagnosticCode4Disabilities.mat');
list=stuff.DiagnosticCode4Disabilities;

%get columns for the disabilities
L=max(size(list));
cols4disabled=zeros(L,1);
for i=1:L
    temp=list(i);
    if isnumeric(temp)
        temp=num2str(temp);
        temp=find(strcmp(temp,all_diagnosis));
    else
        temp=find(strcmp(temp,all_diagnosis));
    end
    if numel(temp)~=0
        cols4disabled(i)=temp(1);
    end
end
cols4disabled=cols4disabled(cols4disabled~=0);

%pull columns from adj matrix corresponding to diabilities
disabled_sparse_Adjmatrix=sparse_Adjmatrix(:,cols4disabled);
```

```
%find disabled patients in adj matrix
[N,m]=size(disabled_sparse_Adjmatrix);
disabled_ill_in_adj=zeros(N,1);
for i=1:N
   if ~isequal(disabled_sparse_Adjmatrix(i,:),zeros(1,m))
       disabled_ill_in_adj(i)=1;
   end
end

%find disabled patients organized by hospital visit
disabled=disabled_ill_in_adj(ic_p);
disabled=find(disabled==1);


end
```

get_rare_ppl_new.

The following MATLAB code is used to find patients with a rare condition in the Baseline model.

```
%---------------
%Author: Takayuki Iguchi
%Last Modified: 14 Dec 2015
%Program Description:
%     Finds the patients with a rare condition in a population.
%Input:
%     -All_patients : A list of unique patient ID numbers.
```

```
%     -DX1: A list of primary diagnoses organized by hospital visit.
%     -DX2: A list of secondary diagnoses organized by hospital visit.
%     -DX3: A list of tertiary diagnoses organized by hospital visit.
%     -DX4: A list of quatrinary diagnoses organized by hospital visit.
%     -PERSONID: A list of patient ID numbers organized by hospital
%     visit.
%Output:
%     -rare_ppl: A list of the patients with a rare condition
%     organized by hospital visit.
%     -rare_ill_in_adj: A list of those identified as having a rare
%     condition organized by appearance in the adjacency matrix.
%--------------

function [rare_ppl,rare_ill_in_adj]=get_rare_ppl_new(All_patients,DX1,
    DX2,DX3,DX4,PERSONID)

% load ICD codings for rare diseases
stuff=load('I:\My Documents\MATLAB\thesis\Create Baseline\
    pull_rare_diseases\orphanet_rarediseases_translated2_ICD9.mat');
list=stuff.unique_rare_list_in_ICD9;
list=unique(list);
list=list(2:end);

%find the hospital visits corresponding to a rare condition
temp=ismember(DX1,list);
temp=temp+ismember(DX2,list);
```

```
temp=temp+ismember(DX3,list);
temp=temp+ismember(DX4,list);
rare_ppl=find(temp>0);

%find the patients with a rare condition as organized in the
%adj matrix
rare_ill_in_adj=zeros(size(All_patients,1),1);
parfor i=1:size(All_patients,1)
    rare_ill_in_adj(i)=max(temp(All_patients(i)==PERSONID));
    disp(i)
end
rare_ill_in_adj=rare_ill_in_adj(rare_ill_in_adj~=0);

end
```

get_chronic_adults.

The following MATLAB code is designed to find the chronically ill in a population.

```
%--------------
% Author: Takayuki Iguchi
% Last modified: 14 Dec 2015
% Program description:
%   -This program is is designed to identify the chronically ill in a
%   patient poplulation.
%   -We use the list of chronic diseases coded in ICD-9-CM provided by
%   the Dartmouth Atlas:
```

```
%   <http://www.dartmouthatlas.org/downloads/methods/
%   Chronic_Disease_Codes.pdf>
%
% Inputs:
%
%   -all_diagnosis : A list of the unique ICD codes that occur in
%   the database.
%   -ic_p : A permutation of the list of unique person ID numbers to
%    the list of person ID numbers arranged by visit.
%   -sparse_Adjmatrix : A sparse adjacency matrix with rows
%    corresponding to a patient and columns corresponding to 3 digit
%   ICD codings. An entry in this array coresponds to how many times a
%   patient was given a certain ICD coding.
%
% Outputs:
%
%   -chronic_ill : A list of indices corresponding to which hospital
%   visits were from people with a chronic condition somewhere in their
%   medical history.
%   -chronic_ill_in_adj: A list of indices corresponding to which
%   patient (i.e. which row in the adjacency matrix) has a chronic
%   condition in their medical history.
%--------------
function [chronic_ill,chronic_ill_in_adj]=get_chronic_adults(
    all_diagnosis,ic_p,sparse_Adjmatrix)
```

```
% Load list of chronic conditions
stuff=load('I:\My Documents\MATLAB\thesis\Create Baseline\
    Dartmouth list of Chronic diseases\list_of_chronic_diseases.mat');
list=stuff.list_of_chronic_diseases;

% Find columns corresponding to chronic conditions
L=max(size(list));
cols4chronic=zeros(L,1);
for i=1:L
    temp=list{i};
    if isnumeric(temp)
        temp=num2str(temp);
        temp=find(strcmp(temp,all_diagnosis));
    else
        temp=find(strcmp(temp,all_diagnosis));
    end
    if numel(temp)~=0
        cols4chronic(i)=temp(1);
    end
end
cols4chronic=cols4chronic(cols4chronic~=0);

% Pull columns from Adj matrix corresponding to chronic conditions
chronic_sparse_Adjmatrix=sparse_Adjmatrix(:,cols4chronic);

% Find patients with a chronic condition in Adj matrix
```

```
[N,m]=size(chronic_sparse_Adjmatrix);
chronic_ill_in_adj=zeros(N,1);
for i=1:N
   if ~isequal(chronic_sparse_Adjmatrix(i,:),zeros(1,m))
        chronic_ill_in_adj(i)=1;
   end
end

% Find patients with a chronic condition organized by hospital visit
chronic_ill=chronic_ill_in_adj(ic_p);
chronic_ill=find(chronic_ill==1);

end
```

setup4getrisk.

This program details the preliminary computations performed in building a classifier to detect if a patient is at risk of a chronic disease.

```
%--------------
% Author: Takayuki Iguchi
% Last modified: 14 Dec 2015
% Program description:
%   -This program performs preliminary calculations to create a
%   classifier that detects if a person is at risk of a chronic
%   condition.
%   -We use the list of chronic diseases coded in ICD-9-CM provided by
%   the Dartmouth Atlas:
```

```
%   <http://www.dartmouthatlas.org/downloads/methods/
%    Chronic_Disease_Codes.pdf>
%
% Inputs:
%
%   -rows2include: A list of rows (i.e. patients) in the adjacency
%   matrix that are not disabled, frail elderly, nor have a chronic
%   condition.
%   -All_diagnosis : A list of the unique ICD codes that occur in the
%   database.
%   -sparse_Adjmatrix : A sparse adjacency matrix with rows
%   corresponding to a patient and columns corresponding to 3 digit
%   ICD codings. An entry in this array corresponds to how many times
%   a patient was given a certain ICD coding.
%
% Outputs:
%
%   -cols4chronic: A list of columns corresponding to chronic
%   conditions.
%   whochronic: A list of patients with chronic conditions.
%   whoNOTchronic: A list of patients without chronic conditions.
%   -Ac_PSD: The mean corrected sums of squares and cross products
%   matrix for the adjacency matrix.
%   -Dc_PSD: The mean corrected sums of squares and cross products
%   matrix for the adjacency matrix of chronically ill patients.
%   -Ac: A centered version of the adjacency matrix without the
```

```
%   columns corresponding to chronic conditions.
%   -Dc: A centered version of the adjacency matrix of chronically
%   ill patients without columns corresponding to chronic conditions.
%--------------

function [cols4chronic,whochronic,whoNOTchronic,Ac_PSD,Dc_PSD,Ac,Dc]=
    setup4getrisk(rows2include,All_diagnosis,sparse_Adjmatrix)
% only consider patients not already classified as disabled,
% frail elderly,
sparse_Adjmatrix=sparse_Adjmatrix(rows2include,:);
sparse_Adjmatrix=sparse_Adjmatrix(any(sparse_Adjmatrix,2),:);

% load ICD codings for chronic conditions
stuff=load('I:\My Documents\MATLAB\thesis\Create Baseline\
    Dartmouth list of Chronic diseases\list_of_chronic_diseases.mat');
list=stuff.list_of_chronic_diseases;
list=collapseICD(list);

% Find columns in Adj matrix corresponding to chronic conditions
L=max(size(list));
cols4chronic=zeros(L,1);
for i=1:L
    temp=list{i};
    if isnumeric(temp)
        temp=num2str(temp);
        temp=find(strcmp(temp,All_diagnosis));
```

```
    else
        temp=find(strcmp(temp,All_diagnosis));
    end
    if numel(temp)~=0
        cols4chronic(i)=temp(1);
    end
end
cols4chronic=cols4chronic(cols4chronic~=0);
chronic_sparse_Adjmatrix=sparse_Adjmatrix(:,cols4chronic);

% find patients with and without a chronic condition
whochronic=find(any(chronic_sparse_Adjmatrix,2));
whoNOTchronic=setdiff(1:max(size(sparse_Adjmatrix)),whochronic);
where2pull4append=any(chronic_sparse_Adjmatrix,2);
append_sparse_matrix=sparse_Adjmatrix(where2pull4append,:);

% clean workspace
clear list
clear temp
clear temparray
clear i
clear L
clear num_test_ppl
clear samplesize
clear stuff
clear subset_idx_begin
```

```
clear subset_idx_end
clear collapse_chronic_icd
clear collapse_list

% calculate the centered data matrices without the columns
% corresponding to chronic conditions
Ac=sparse_Adjmatrix-ones(size(sparse_Adjmatrix,1),1)*(sum(
    sparse_Adjmatrix,1)./size(sparse_Adjmatrix,1));
Dc=append_sparse_matrix;
Dc=Dc-ones(size(Dc,1),1)*(sum(Dc,1)./size(Dc,1));
Ac(:,cols4chronic)=[];
Dc(:,cols4chronic)=[];



% calculate the mean corrected sums of squares and cross
% products matrix
Ac_PSD=Ac'*Ac;
Dc_PSD=Dc'*Dc;

end
```

get_chronic_risk_PCAmotivatedwcrossvalidation.

This program finds the patients at risk of a chronic disease by creating a classifier that performs the best in cross validation over all 30 trials.

```
%--------------
% Author: Takayuki Iguchi
```

```
% Last modified: 14 Dec 2015
% Program description:
%   -This program creates a classifier that identifies patients
%   who are at risk of a chronic condition. We perform
%   cross-validation until we have enough instances where our
%   expectations (below) are met. We then select the best
%   performing classifier and implement it on the entire dataset
%   to obtain the patients who are at risk of a chronic
%   condition who do not already have one.
%   -Note that we impose our bias that the majority of
%   non-chronically ill people should not be at at-risk of a
%   chronic condition.
%
% Inputs:
%
%   whochronic: A list of patients with chronic conditions.
%   whoNOTchronic: A list of patients without chronic conditions.
%   -Ac: A centered version of the adjacency matrix without the
%   columns corresponding to chronic conditions.
%   -Dc: A centered version of the adjacency matrix of chronically
%   ill patients without columns corresponding to chronic conditions.
%   -trials: The number of trials to be performed (which also meet
%   our expectations.
%
% Outputs:
%
```

```
%   -chronic_risk: A list of patients at-risk of a chronic condition.
%   -crossvalidationresults: A record of the crossvalidation for
%   each trial to include those that do not meet our expectations.
%   -common_sensecheck: A record of the proportions of patients
%   who are chronically ill in the at-risk cluster and the
%   proportion of the non-chronically ill in the at-risk cluster.
%   -where_cmn_sns: A record of which trials met our expectations.
%--------------

function [chronic_risk,crossvalidationresults,common_sensecheck,
    where_cmn_sns]=get_chronic_risk_PCAmotivatedwcrossvalidation(
        whochronic,whoNOTchronic,Ac,Dc,trials)

% initialize loop to perform cross validation
crossvalidationresults=[];
common_sensecheck=[];
where_cmn_sns=[];
IDX_opt=zeros(size(Ac,1),1);
cross_opt=0;
cont=true;
successfull_cmn_sense=0;
count=1;

while cont

    % pull 2/3 of nonchronic and chronic, respectively
```

```
num_whochronic=max(size(whochronic));
num_pull_whochronic=floor(2*num_whochronic/3);
pull_whochronic=randsample(1:max(size(Dc)),num_pull_whochronic);
Ac_PSD_pulled=Ac(pull_whochronic,:)'*Ac(pull_whochronic,:);
Dc_PSD_pulled=Dc(pull_whochronic,:)'*Dc(pull_whochronic,:);

% find principal components of the regularization
i=6;
lambda=2^i;
A=Ac_PSD_pulled-lambda*Dc_PSD_pulled;
[V,~]=eigs(A);

% run k-means the PCA scores for a subset of the components
[IDX,~,~]=kmeans(Ac*V(:,2),2);

% record the proportions of patients who are chronically ill
% in the at-risk cluster and the proportion of the
% non-chronically ill in the at-risk cluster.
if max(size(find(IDX(whochronic)==1)))>=max(size(find(
    IDX(whochronic)==2)))
    common_sensecheck(count,1)=max(size(find(IDX(
        whochronic)==1)))/max(size(whochronic));
    common_sensecheck(count,2)=max(size(find(IDX(whoNOTchronic)=
        =1)))/max(size(whoNOTchronic));
else
    common_sensecheck(count,1)=max(size(find(IDX(whochronic)==2)))
```

```
        /max(size(whochronic));
    common_sensecheck(count,2)=max(size(find(IDX(whoNOTchronic)==2
        )))/max(size(whoNOTchronic));
end

% determine if the majority of non-chronically disease people
% are in the not-at risk cluster (i.e. test if the clustering
% meets our expectations
if max(size(find(IDX(whochronic)==1)))>=max(size(find(IDX(
    whochronic)==2)))
    if max(size(find(IDX(whoNOTchronic)==1)))<max(size(find(IDX(
        whoNOTchronic)==2)))
        fprintf('meets our expectations!\n')
        successfull_cmn_sense=successfull_cmn_sense+1;
        where_cmn_sns(end+1)=count;
    end
else
    if max(size(find(IDX(whoNOTchronic)==2)))<max(size(find(IDX(
        whoNOTchronic)==1)))
        fprintf('meets our expectations!\n')
        successfull_cmn_sense=successfull_cmn_sense+1;
        where_cmn_sns(end+1)=count;
    end
end

% test if we have enough trials which conforms to expectations
```

```
if successfull_cmn_sense>=trials
    cont=false;
end

%find vector between centroids
c2c_kmeans=sum(Ac(IDX==1,:),1)/size(IDX==1,1)-
    sum(Ac(IDX==2,:),1)/size(IDX==2,1);
c2c_kmeans=c2c_kmeans/norm(c2c_kmeans);

if max(size(find(IDX(whochronic)==1)))>=max(size(find(IDX(
    whochronic)==2)))
    means_atrisk    =sign(c2c_kmeans*(Ac(randsample(find(IDX==1),
        1),:))');
else
    means_atrisk    =sign(c2c_kmeans*(Ac(randsample(find(IDX==2),
        1),:))');
end

%get remaining 1/3 of chronic and inner prod with normal to
%hyperplane; threshold

testchron=setdiff(1:max(size(whochronic)),pull_whochronic);
testvec_chron=c2c_kmeans*(Dc(testchron,:))';
whichinDcatrisk=(sign(testvec_chron)==means_atrisk);

% calculate crossvalidation results
```

```
    crossvalidationresults(count)=sum(whichinDcatrisk)/max(size(
        testvec_chron));

    % store information regarding best cross validation results
    if crossvalidationresults(count)>cross_opt
        cross_opt=crossvalidationresults(count);
        IDX_opt=IDX;
        Acx=Ac*V(:,1);
        Ac2x=Ac*V(:,2);
        Ac3x=Ac*V(:,3);
    end

    % show progress
    disp([count,successfull_cmn_sense,crossvalidationresults(count)])

    % increment counter/pointer
    count=count+1;
end

% Using the best clustering under cross validation,
% find the people who are at risk of a chronic
% condition who does not already have one.
if max(size(find(IDX_opt(whochronic)==1)))>=max(size(find(IDX_opt(
    whochronic)==2)))
    chronic_risk=find(IDX_opt==1);
else
```

```
    chronic_risk=find(IDX_opt==2);
end
chronic_risk=setdiff(chronic_risk,whochronic);

% save results
filename='I:\My Documents\MATLAB\thesis\Create Baseline\WORKSPACE DATA
    \Baseline_results.mat';
save(filename,'chronic_risk','crossvalidationresults',
    'common_sensecheck','where_cmn_sns','-append')

end
```

get_children.

This program finds the patients who are children in the dataset.

```
%--------------
% Author: Takayuki Iguchi
% Last modified: 14 Dec 2015
% Program description:
%   -This program finds children and adults who were children
%   at some point in time in the dataset.
%
% Inputs:
%
%     -AGE : A list of patient ages organized by hospital
%     visit.
%     -PERSONID : A list of patient ID numbers organized by
```

```
%     hospital visit.
%
% Outputs:
%
%   -children: A list of patients who are children organized by
%   hospital visit.
%   -update_info: A list of adult patients who were children in the
%   dataset organized by hospital visit.
%--------------

function [children,update_info]=get_children(PERSONID,AGE)
max_age=18;
N=max(size(AGE));
unique_PPL=unique(PERSONID);
N_unique=max(size(unique_PPL));
update_info_data{N,1}=[];
children_data{N,1}=[];
parfor i=1:N_unique
   where=find(PERSONID==unique_PPL(i));
   ages=AGE(where);

   if max(ages)<max_age
   children_data(i)={where};
   elseif min(ages)<max_age
   where_child=where(ages<max_age);
   where_adult=where(ages>=max_age);
```

```
   children_data(i)={where_child};
   update_info_data(i)={where_adult};
   end
   disp(i)
end


children=[];
update_info=[];
for i=1:N_unique
    children=vertcat(children,children_data{i});
    update_info=vertcat(update_info,update_info_data{i});
    disp(i)
end
children=sort(children);
update_info=sort(update_info);
end
```

get_mothers_precompute.

The following MATLAB code is designed to find the eldest child in each family and their ages.

```
%--------------
%Author: Takayuki Iguchi
%Last Modified: 14 Dec 2015
%Program Description:
%    Finds eldest child in each family and records their ages.
%Input:
```

```
%     -children : A list of indices denoting the position of children
%     in the dataset organized by hospital visit.
%     -AGE : A list of patient ages organized by hospital
%     visit.
%     -PERSONID : A list of patient ID numbers organized by hospital
%     visit.
%     -FAMILYID : A list of family ID numbers organized by
%     hospital visit.
%     -FAMILYID : A list denoting the year the hospital visits occured.
% Output:
%     -eldest_child_ages : A list of the eldest child in each family.
%--------------

function [eldest_child_ages]=get_mothers_precompute(children,AGE,
    PERSONID,FAMILYID,YEAR)
families_w_children=unique(FAMILYID(children));
%find the oldest child in each family
n=max(size(families_w_children));
eldest_child_ages=zeros(n,1);
parfor i=1:n
    familyid=families_w_children(i);
    if familyid~=-1 && ~isnan(familyid)
        children_in_family_id=PERSONID(intersect(children,find(
            familyid==FAMILYID)));
        uni_children_in_family_id=unique(children_in_family_id);
        maxage=-1;
```

```
        for j=1:max(size(uni_children_in_family_id))
            temp=zeros(size(PERSONID,1),1);
            temp(PERSONID==uni_children_in_family_id(j))=AGE(PERSONID=
                =uni_children_in_family_id(j));
            [tempage,temp_location]=max(temp);
            if tempage>maxage
                maxage=tempage;
                if numel(temp_location)>1
                    temp_location=find(YEAR==max(YEAR(temp_location)));
                end
                max_location=temp_location;
            end
        end
        eldest_child_ages(i)=YEAR(max_location)-maxage;
    end
    disp([i,n])
end
end
```

get_mothers_new.

The following MATLAB code identifies mothers of the children in the dataset.

```
%--------------
%Author: Takayuki Iguchi
%Last Modified: 14 Dec 2015
%Program Description:
%    Finds the mother in each family.
```

```
%Input:
%     -children : A list of indices denoting the position of children
%     in the dataset organized by hospital visit.
%     -AGE : A list of patient ages organized by hospital
%     visit.
%     -PERSONID : A list of patient ID numbers organized by hospital
%     visit.
%     -FAMILYID : A list of family ID numbers organized by
%     hospital visit.
%     -FAMILYID : A list denoting the year the hospital visits occured.
%     -YEAR : A list of the years a hopsital visit was made.
%     -eldest_child_ages : A list of the eldest child in each family.
% Output:
%     -mothers : A list of patients who are identified as being mothers
%     organized by hospital visit.
%--------------

function mothers=get_mothers_new(children,AGE,PERSONID,FAMILYID,GENDER
    ,YEAR,eldest_child_ages)
mothers=zeros(size(PERSONID,1),1);
families_w_children=unique(FAMILYID(children));

N=size(PERSONID,1);
for index=1:N
    if strcmp(GENDER(index),'F')
        familyid=FAMILYID(index);
```

```
        if familyid~=-1
            age_of_person=AGE(index);
            if age_of_person>9
                familyid_match=(familyid==families_w_children);
                if any(familyid_match)
                    eldest_childage  =eldest_child_ages(familyid_match
                        ==1);
                    if age_of_person-YEAR(index)+eldest_childage>9
                        mothers(index)=index;
                    end
                end
            end
        end
    end
    disp([index,N])
end
mothers=unique(mothers(mothers~=0));
end
```

Code for the k-means model.

In this subsection we present the code used in generating the k-means model.

find_best_entropy_and_record_kmeans_output_version_correct.

This MATLAB script creates k-means models for $k = 2$ to $k = 10$ with a random initialization.

```
%--------------
```

```
%Author: Takayuki Iguchi
%Last Modified: 1 Feb 2016
%Program Description:
%    This program creates the k-means model by running multiple trials
%    for each value k specified. The program then selects the best
%    model based on conditional entropy and retains information
%    regarding it. Furthermore, in this program k-means runs on the
%    first five years of medical history after it has been transformed
%    into a 0-1 matrix. Namely, any 0 remains a 0 and any positive
%    entry turns into a 1.
%--------------

% load data                                             --------------
%load data from the first five years
filename4adjmatrix= 'I:\My Documents\MATLAB\thesis\Create Baseline
    \WORKSPACE DATAFIRSTCOLLAPSEDsparse_Adjmatrix.mat';
loadtemp=load(filename4adjmatrix);
sparse_AdjmatrixFIRST=loadtemp.FIRST_COLLAPSEDsparse_Adjmatrix;

%load data from the second five years
filename4adjmatrix='I:\My Documents\MATLAB\thesis\Create Baseline
    \WORKSPACE DATASECONDCOLLAPSEDsparse_Adjmatrix.mat';
loadtemp=load(filename4adjmatrix);
sparse_AdjmatrixSECOND=loadtemp.SECOND_COLLAPSEDsparse_Adjmatrix;

%clean workspace
```

```
clear loadtemp

%print progress
fprintf('Completed loading data\n')

%initialize filename for where to save data
filename4kmeanscalcs='I:\My Documents\MATLAB\thesis\
    Create kmeans model\WORKSPACE DATA\
        entropy_experiments_k2to10_corrected.mat';



% use only patients in both year groups                 --------------
people_exist_1=any(sparse_AdjmatrixFIRST,2);
people_exist_2=any(sparse_AdjmatrixSECOND,2);
people_exist_1and2=people_exist_1.*people_exist_2;
sparse_AdjmatrixFIRST=sparse_AdjmatrixFIRST(people_exist_1and2==1,:);
sparse_AdjmatrixSECOND=sparse_AdjmatrixSECOND(people_exist_1and2==1,:);
fprintf('Completed pulling only patients in both year groups\n')
clear people_exist_1
clear people_exist_2

% make Adjmatrices 0-1 matrices
sparse_Adjmatrix1st_01=sparse_AdjmatrixFIRST;
sparse_Adjmatrix1st_01(sparse_Adjmatrix1st_01>1)=1;

% parameter control                                     --------------
```

```
kmin=10;

kmax=10;

trials=4;


% initialize variables to record trials

[N,m]=size(sparse_Adjmatrix1st_01);


Harray      =zeros(kmax-kmin+1  ,1);

Harray_av   =zeros(kmax-kmin+1  ,1);

Harray_std  =zeros(kmax-kmin+1  ,1);

sumdarray   =zeros(kmax         ,kmax-kmin+1);

Carray      =zeros(kmax         ,m              ,kmax-kmin+1);

IDXarray    =zeros(N            ,kmax-kmin+1);

IDXarray4all=zeros(N            ,kmax-kmin+1,trials);

Htrials4all =zeros(kmax-kmin+1,trials);

count=1;

Hbest=realmax;

where_best=1;

fprintf('Completed initializing for loop on k.\n')


% perform trials                                          --------------

for k=kmin:kmax


    %initialize trials for a specific value of k

    fprintf(['Initializing for trials when k=',num2str(k),'.\n'])

    IDXtrials   =zeros(N    ,trials);
```

```
Ctrials     =zeros(k    ,m      ,trials);
sumdtrials  =zeros(k    ,trials);
Htrials     =zeros(1    ,trials);
if k==kmin
    save(filename4kmeanscalcs,...
        'Harray',...
        'Harray_av',...
        'Harray_std',...
        'sumdarray',...
        'Carray',...
        'IDXarray')
end

%print progress
fprintf('Completed initializing for loop on trials.\n')

%perform trials for a specific value of k
parfor ts=1:trials

    %calculate and record k-means clustering based on first
    %5 years
    [IDXtemp,Ctemp,sumdtemp] = kmeans(full(sparse_Adjmatrix1st_01)
        ,k);
    IDXtrials(:,ts)=IDXtemp;
    Ctrials(:,:,ts)=Ctemp;
    sumdtrials(:,ts)=sumdtemp;
```

```
    %print progress
    fprintf([’completed kmeans on first half for k=’,num2str(k),’
        , trial=’,num2str(ts),’\n’])
    fprintf([’Completed k-means for ts=’,num2str(ts),’ and when
        k=’,num2str(k),’.\n’])

    %calculate conditional entropy of the clustering
    [Htrials(ts),~]=entropy_calc_correct(IDXtrials(:,ts),
        sparse_AdjmatrixSECOND);

    %print progress
    fprintf([’completed calculating entropy for k=’,num2str(k),’,
        trial=’,num2str(ts),’\n’])
    fprintf([’Trial: ts=’,num2str(ts),’ and when k=’,
        num2str(k),’ Complete.\n’])
    c=clock;
    fprintf([’Time: ’,num2str(c(4)),’ : ’,num2str(c(5)),’hrs\n’])
end

%record important information from trials for specific value of k
[Harray(count),where_opt]=min(Htrials(Htrials~=0));
Htrials4all(count,:)=Htrials;
Harray_av(count)=mean(Htrials(Htrials~=0));
Harray_std(count)=std(Htrials(Htrials~=0));
if sum(Htrials==0)>0
```

```
    fprintf('There was a case of zero entropy??\n')
end
IDXarray(:,count)=IDXtrials(:,where_opt);
IDXarray4all(:,count,:)=IDXtrials;
Carray(1:k,:,count)=Ctrials(:,:,where_opt);
sumdarray(1:k,count)=sumdtrials(:,where_opt);
if Harray(count)<Hbest
    Hbest=Harray(count);
    where_best=count;
    IDXbest=IDXtrials(:,where_opt);
    Cbest=Ctrials(:,:,where_opt);
    sumdbest=sumdtrials(:,where_opt);
end

%save important information from trials for specific value of k
save(filename4kmeanscalcs,...
    'Harray',...
    'Harray_av',...
    'Harray_std',...
    'sumdarray',...
    'Carray',...
    'IDXarray',...
    'Hbest',...
    'where_best',...
    'IDXbest',...
    'Cbest',...
```

```
        'sumdbest',...
        'IDXarray4all',...
        'Htrials4all',...
        '-append');

    %print progress
    fprintf(['Completed saving when k=',num2str(k),'.\n'])

    %increment pointer
    count=count+1;
end
```

Miscellaneous code.

In this subsection we present miscellaneous, but important code.

entropy_calc_correct.

The following MATLAB code calculates the conditional entropy with which we concern ourselves.

```
%--------------
%Author: Takayuki Iguchi
%Last Modified: 8 Feb 2015
%Program Description:
%    This function calculates the conditional entropy of a patient
%    being given a diagnosis given he/she is in a particular cluster.
%Input:
%     -adjmatrix : An N x m array with rows corresponding to patients
```

```
%       and columns corresponding to possible diagnoses/ ICD codings.
%     -IDX : An N x 1 array describing the the cluster to which a
%       patient belongs.
% Output:
%     -H : The conditional entropy aforementioned.
%     -Harray : A k dimensional array (k being the number of clusters)
%      with the conditional entropy H(Y|X=x), i.e., the entropy of a
%      a diagnosis being given to a patient given the patient is in
%      cluster x.
%--------------

function [H,Harray]=entropy_calc_correct(IDX,adjmatrix)

%initialize
unique_IDX=unique(IDX);
num_clusters=max(size(unique_IDX));
parray=zeros(num_clusters,size(adjmatrix,2));
Harray=zeros(num_clusters,1);
num_pats=size(adjmatrix,1);
num_diag_given=sum(sum(adjmatrix));

for i=1:num_clusters
    %calculate the number of patients in a cluster
    num_pat_in_clust=sum(IDX==unique_IDX(i));
    p_of_x=num_pat_in_clust/num_pats;
```

```
    %for each 3 digit ICD code, calculate the conditional probability
    %of a diagnosis occuring in the ith cluster
    parray(i,:)=sum(adjmatrix(IDX==unique_IDX(i),:),1);
    parray(i,:)=parray(i,:)/sum(parray(i,:));
    temp_p=parray(i,:);

    %by convention 0 log(0) := 0.
    temp_p=temp_p(temp_p~=0);

    %calculate the conditional entropy H(Y|X=x).
    Harray(i)=sum(-temp_p.*log2(temp_p))*p_of_x;
end
%calculate the conditional entropy H(Y|X).
H=sum(Harray);
end
```

Bibliography

1. M. E. Porter and T. H. Lee, “The strategy that will fix health care,” *Harvard Business Review*, vol. 91, no. 10, pp. 50–70, 2013.
2. “Approaches to reducing federal spending on military health care.” *Congressional Budget Office*, January 2014. Available: `https://www.cbo.gov/sites/default/files/113th-congress-2013-2014/reports/44993-MilitaryHealthcare.pdf` [Last accessed: 21 January 2016].
3. “United states department of defense fiscal year 2016 budget request overview.” *Office of the Under Secretary of Defense (Comptroller) Chief Financial Officer*, February 2015. Available: `http://comptroller.defense.gov/Portals/45/documents/defbudget/fy2016/fy2016_Budget_Request_Overview_Book.pdf` [Last accessed: 21 January 2016].
4. M. E. Porter, E. A. Pabo, and T. H. Lee, “Redesigning primary care: a strategic vision to improve value by organizing around patients needs,” *Health Affairs*, vol. 32, no. 3, pp. 516–525, 2013.
5. M. E. Porter, C. Guth, E. Dannemiller, *et al.*, “The west german headache center: integrated migraine care,” *Boston: Harvard Business School Publishing*, 2007.
6. L. Barr, “New plan to dramatically change today’s health systems.” *Air Force Surgeon General Public Affairs*, 20 March 2015. Available: `http://www.buckley.af.mil/news/story.asp?id=123442489` [Last accessed: 14 January 2016].
7. “Air force future operating concept.” *Secretary of the Air Force Public Affairs Command Information*, September 2015. Available: `http://www.af.mil/Portals/1/images/airpower/AFFOC.pdf` [Last accessed: 21 January 2016].

8. S. Lloyd, "Least squares quantization in pcm," *Information Theory, IEEE Transactions on*, vol. 28, no. 2, pp. 129–137, 1982.

9. A. Coates, A. Y. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *International conference on artificial intelligence and statistics*, pp. 215–223, 2011.

10. H. Ng, S. Ong, K. Foong, P. Goh, and W. Nowinski, "Medical image segmentation using k-means clustering and improved watershed algorithm," in *Image Analysis and Interpretation, 2006 IEEE Southwest Symposium on*, pp. 61–65, IEEE.

11. W. Shi and W. Zeng, "Genetic k-means clustering approach for mapping human vulnerability to chemical hazards in the industrialized city: A case study of shanghai, china," *International journal of environmental research and public health*, vol. 10, no. 6, pp. 2578–2595, 2013.

12. P. S. Balakrishnan, M. C. Cooper, V. S. Jacob, and P. A. Lewis, "Comparative performance of the fscl neural net and k-means algorithm for market segmentation," *European Journal of Operational Research*, vol. 93, no. 2, pp. 346–357, 1996.

13. R. Kuo, L. Ho, and C. Hu, "Integration of self-organizing feature map and k-means algorithm for market segmentation," *Computers and Operations Research*, vol. 29, no. 11, pp. 1475 – 1493, 2002.

14. D.-M. Kim, K.-S. Kim, K.-H. Park, J.-H. Lee, and K. M. Lee, "A music recommendation system with a dynamic k-means clustering algorithm," in *Machine Learning and Applications, 2007. ICMLA 2007. Sixth International Conference on*, pp. 399–403, Dec 2007.

15. S. Yokoi, Y. N. Takayabu, K. Nishii, H. Nakamura, H. Endo, H. Ichikawa, T. Inoue, M. Kimoto, Y. Kosaka, T. Miyasaka, *et al.*, "Application of cluster analysis to climate model performance metrics," *Journal of Applied Meteorology and Climatology*, vol. 50, no. 8, pp. 1666–1675, 2011.

16. S. V. Nath, "Crime pattern detection using data mining," *Web Intelligence and Intelligent Agent Technology, International Conference on*, vol. 0, pp. 41–44, 2006.

17. G. Münz, S. Li, and G. Carle, "Traffic anomaly detection using k-means clustering,"

18. Y. Lee and S. Choi, "Minimum entropy, k-means, spectral clustering," in *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, vol. 1, IEEE, 2004.

19. K. Jain, M. Mahdian, and A. Saberi, "A new greedy approach for facility location problems," in *Proceedings of the 34th Annual ACM Symposium on Theory of Computing*, 2002.

20. D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*, 2007.

21. R. Ostrovsky, Y. Rabani, L. Schulman, and C. Swamy, "The effectiveness of lloyd-type methods for the k-means problem," in *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, 2006.

22. P. Awasthi, A. Bandeira, M. Charikar, K. Ravishankar, S. Villar, and R. Ward, "Relax, no need to round: Integrality of clustering formulations," *http://arxiv.org/abs/1408.4045*, 2014.

23. S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2009.

24. D. G. Mixon and J. D. Peterson, “Cone programming cheat sheet,” *Short, Fat Matrices (weblog)*, 2015.

25. A. S. Bandeira, “A note on probably certifiably correct algorithms,” *arXiv preprint arXiv:1509.00824*, 2015.

26. E. Abbe, A. S. Bandeira, and G. Hall, “Exact recovery in the stochastic block model,” *arXiv preprint arXiv:1405.3267*, 2014.

27. A. Celisse, J.-J. Daudin, L. Pierre, *et al.*, “Consistency of maximum-likelihood and variational estimators in the stochastic block model,” *Electronic Journal of Statistics*, vol. 6, pp. 1847–1899, 2012.

28. A. Decelle, F. Krzakala, C. Moore, and L. Zdeborová, “Asymptotic analysis of the stochastic block model for modular networks and its algorithmic applications,” *Physical Review E*, vol. 84, no. 6, p. 066106, 2011.

29. D. E. Fishkind, D. L. Sussman, M. Tang, J. T. Vogelstein, and C. E. Priebe, “Consistent adjacency-spectral partitioning for the stochastic block model when the model parameters are unknown,” *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 1, pp. 23–39, 2013.

30. D. B. Larremore, A. Clauset, and A. Z. Jacobs, “Efficiently inferring community structure in bipartite networks,” *Phys. Rev. E*, vol. 90, p. 012805, Jul 2014.

31. V. Feldman, W. Perkins, and S. Vempala, “Subsampled power iteration: a new algorithm for block models and planted csp’s,” *CoRR*, vol. abs/1407.2774, 2014.

32. L. Florescu and W. Perkins, "Spectral Thresholds in the Bipartite Stochastic Block Model," *ArXiv e-prints*, June 2015.

33. G. H. Golub and C. F. Van Loan, *Matrix computations*, vol. 3. JHU Press, 2012.

34. B. Laurent and P. Massart, "Adaptive estimation of a quadratic functional by model selection," *Annals of Statistics*, pp. 1302–1338, 2000.

35. J. Peng and Y. Wei, "Approximating k-means-type clustering via semidefinite programming," *SIAM Journal on Optimization*, vol. 18, no. 1, pp. 186–205, 2007.

36. I. S. Dhillon, Y. Guan, and B. Kulis, "Kernel k-means: spectral clustering and normalized cuts," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 551–556, ACM, 2004.

37. I. S. Dhillon, Y. Guan, and B. Kulis, "Weighted graph cuts without eigenvectors a multilevel approach," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 29, no. 11, pp. 1944–1957, 2007.

38. H. Wang and M. Song, "Ckmeans.1d.dp: optimal k-means clustering in one dimension by dynamic programming," *The R Journal*, vol. 3, no. 2, pp. 29–33, 2011.

39. M. Grant, S. Boyd, and Y. Ye, "Cvx: Matlab software for disciplined convex programming," 2008.

40. E. Elhamifar, G. Sapiro, and R. Vidal, "Finding exemplars from pairwise dissimilarities via simultaneous sparse recovery," in *Advances in Neural Information Processing Systems*, pp. 19–27, 2012.

41. A. Nellore and R. Ward, "Recovery guarantees for exemplar-based clustering," *arXiv:1309.3256*, 2013.

42. S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” *Signal Processing, IEEE Transactions on*, vol. 41, no. 12, pp. 3397–3415, 1993.

43. B. Karrer and M. E. J. Newman, “Stochastic blockmodels and community structure in networks,” *Physical Review E*, vol. 83, p. 016107, Jan. 2011.

44. A. S. Bandeira, “Community detection via semidefinite relaxation,” 2014.

45. E. Abbe and C. Sandon, “Community detection in general stochastic block models: fundamental limits and efficient recovery algorithms,” *arXiv preprint arXiv:1503.00609*, 2015.

46. M. Khalilia, S. Chakraborty, and M. Popescu, “Predicting disease risks from highly imbalanced data using random forest,” *BMC medical informatics and decision making*, vol. 11, no. 1, p. 51, 2011.

47. L. Van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. 2579-2605, p. 85, 2008.

48. J. A. Tropp, “User-friendly tail bounds for sums of random matrices,” *Foundations of Computational Mathematics*, vol. 12, no. 4, pp. 389–434, 2012.

49. R. Vershynin, “Introduction to the non-asymptotic analysis of random matrices,” *arXiv:1011.3027v7*, 2011.

50. S. Chawla, “14-859(m): Randomized algorithms: Chernoff bounds.” 2004.

51. J. Canny, “Lecture 10 chernoff bounds cs 174.” [Lecture notes].

52. D. C. Goodman, A. R. Esty, E. S. Fisher, and K. K. Bonner, *Trends and variation in end-of-life care for Medicare beneficiaries with severe chronic illness.* Dartmouth Institute for Health Policy and Clinical Practice, 2011.

53. L. I. Iezzoni, T. Heeren, S. M. Foley, J. Daley, J. Hughes, and G. A. Coffman, “Chronic conditions and risk of in-hospital death.,” *Health services research*, vol. 29, no. 4, p. 435, 1994.

54. A. Clegg, J. Young, S. Iliffe, M. O. Rikkert, and K. Rockwood, “Frailty in elderly people,” *The Lancet*, vol. 381, no. 9868, pp. 752 – 762, 2013.

55. K. Rockwood, X. Song, C. MacKnight, H. Bergman, D. B. Hogan, I. McDowell, and A. Mitnitski, “A global clinical measure of fitness and frailty in elderly people,” *Canadian Medical Association Journal*, vol. 173, no. 5, pp. 489 – 498, 2005.

56. D. A. Davis, N. V. Chawla, N. Blumm, N. Christakis, and A.-L. Barabási, “Predicting individual disease risk based on medical history,” in *Proceedings of the 17th ACM conference on Information and knowledge management*, pp. 769–778, ACM, 2008.

57. N. T. Thong *et al.*, “Intuitionistic fuzzy recommender systems: An effective tool for medical diagnosis,” *Knowledge-Based Systems*, vol. 74, pp. 133–150, 2015.

Vita

Second Lieutenant Takayuki Iguchi graduated from Jesuit High School in Portland, Oregon. He attended the United States Air Force Academy where he earned his commission and a Bachelors degree in Mathematics with a minor in Japanese. Upon graduation in May 20114, he was recognized as a Distinguished Graduate with military honors and the Outstanding Cadet in Mathematics.

His first assignment was at Wright Patterson Air Force Base as a student at the Graduate School of Engineering and management, Air Force Institute of Technology where he pursued a Masters of Science in Applied Mathematics. Upon graduation, he will be assigned to Edwards Air Force Base.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704–0188*

1. REPORT DATE *(DD–MM–YYYY)* 24–03–2016	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** *(From — To)* Sept 2014 — Mar 2016

4. TITLE AND SUBTITLE Clustering Theory and Data Driven Health Care Strategies	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Iguchi, Takayuki, Second Lieutenant, USAF	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENC) 2950 Hobson Way WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT-ENC-MS-16-M-001
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Genny M. Maupin, M.P.H. 711th Human Performance Wing 2610 Seventh Street Bldg. 441 WPAFB OH 45433 DSN 785-3252, COMM 937-522-3252	**10. SPONSOR/MONITOR'S ACRONYM(S)** 711 HPW **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION / AVAILABILITY STATEMENT

DISTRIBUTION STATEMENT A:
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

14. ABSTRACT

DoD health care requires reform with growing costs causing concerns of decreased military capability. One proposed radical strategy to fix current health care delivery systems is to organize medical teams around patients with similar treatment requirements. This is a clustering problem; how do you partition the set of patients so that each group has similar treatment needs? We provide advances in clustering theory relevant to this new health care strategy. In particular, we create fast certifiably optimal k-means clustering using what is known as Probably Certifiably Correct (PCC) algorithms which achieves state-of-the-art performance under certain models. Inspired by the health care clustering problem, we pay particular attention to a Bipartite Stochastic Block Model and produce an alternative PCC algorithm specific to this model. We conclude by demonstrating the potential utility of applying these clustering methods in health care. Using conditional entropy as a metric, clusters obtained from our methods vastly outperform partitions prescribed by subject matter experts.

15. SUBJECT TERMS

k-means, unsupervised learning, clustering, health care, Probably Certifiably Correct algorithm, semidefinite relaxation, stochastic block model, bipartite stochastic block model.

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Captain J. D. Peterson, PhD, AFIT/ENC
U	U	U	U	182	**19b. TELEPHONE NUMBER** *(include area code)* (937) 255-3636, x4619