FAIR AND DIVERSE DATA REPRESENTATION IN MACHINE LEARNING

A Dissertation
Presented to
The Academic Faculty

By

Uthaipon Tantipongpipat

In Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy in the
Algorithms, Combinatorics, and Optimization

Georgia Institute of Technology

August 2020

FAIR AND DIVERSE DATA REPRESENTATION IN MACHINE LEARNING

Approved by:

Dr. Mohit Singh, Advisor
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Rachel Cummings
School of Industrial and Systems Engineering
Georgia Institute of Technology

Dr. Aleksandar Nikolov
Department of Computer Science
University of Toronto

Dr. Sebastian Pokutta
Institute of Mathematics
Technical University of Berlin

Dr. Santosh Vempala
School of Computer Science
Georgia Institute of Technology

Date Approved: May 8, 2020

ACKNOWLEDGEMENTS

This thesis would not be complete without many great collaborators and much support I received. My advisor, Mohit Singh, is extremely supportive for me to explore my own research interests, while always happy to discuss even a smallest technical detail when I need help. His advising aim has always been for my best – for my career growth as a researcher and for a fulfilling life as a person. He is unquestionably an integral part of my success today.

I want to thank Rachel Cummings, Aleksandar (Sasho) Nikolov, Sebastian Pokutta, and Santosh Vempala for many research discussions we had together and their willingness to serve as my dissertation committees amidst COVID-19 disruption. I want to thank my other coauthors as part of my academic path: Digvijay Boob, Sara Krehbiel, Kevin Lai, Vivek Madan, Jamie Morgenstern, Samira Samadi, Amaresh (Ankit) Siva, Chris Waites, and Weijun Xie. I am grateful for friendly and supportive peers at Georgia Tech: Prateek Bhakta, Sarah Cannon, Zongchen Chen, Ben Cousins, David Durfee, Majid Farhadi, Yu Gao, Bhuvesh Kumar, Samantha Petti, Saurabh Sawlani, Sadra Yazdanbod, Sina Yazdanbod, and many others.

Janardhan Kulkarni and Sergey Yekhanin were wonderful supervisors and collaborators during my internship at Microsoft Research, Redmond. I enjoyed our research discussions that broaden my research horizon. Both of them were a memorable part of my internship experience that has significantly shaped my research and career path.

I would not be here today without continuous support of my family, friends, and mentors in my life, especially the support they gave when I needed most. I want to especially thank Jocelyn Davis, Doug Johns, Aaron Menikoff, Bryan Pillsbury, and MaryAnne Turner for many of their extremely helpful, timely wisdom on my career planning and personal life.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

The work consists of two major topics: subset selection and multi-criteria dimensionality reduction with an application to fairness. Subset selection can be applied to a classical problem *Optimal Design* in statistics and many others in machine learning including diverse sampling.

Our first contribution is to show that approximability of many criteria for subset selection can be obtained by novel polynomial-time sampling algorithms, improving upon best previous approximation ratios in the literature. The results apply to several generalizations of the problem, many of which are novel. We also show that the A-optimal criterion is NP-hard and that the best-known approximation for E-optimal criterion is tight up to the natural convex relaxation.

One of the most common heuristics used in practice to solve A and D-optimal criteria is the local search heuristic, also known as the Fedorov's exchange method [Fed72]. This is due to its simplicity and its empirical performance [CN80, MN94, ADT07]. However, despite its wide usage, no theoretical bound has been proven for this algorithm. We bridge this gap and prove approximation guarantees for the local search algorithms for A- and D-optimal criteria.

This thesis also extends the arguably most commonly used dimensionality reduction technique, Principal Component Analysis (PCA), to satisfy a fairness criterion of choice. We model an additional fairness constraint as *multi-criteria dimensionality reduction* where we are given multiple objectives that need to be optimized simultaneously. Our model of multi-criteria dimensionality reduction captures several fairness criteria for dimensionality reduction motivated from economic literature. Our technical contribution is to prove new low-rank properties of extreme point solutions to semi-definite programs, which gives theoretical performance to our algorithms for multi-criteria dimensionality reduction. Finally, we perform experiments on real-world datasets indicating the effectiveness of the algorithms and demonstrating empirical scalability of our proposed implementations in practice.

CHAPTER 1
INTRODUCTION

This thesis composes of two topics: subset selection and multi-criteria dimensionality reduction. Each topic outlines the introduction which includes the problem formulation and motivation, previous and related work, summary of contribution, and future directions.

1.1 Diverse Subset Selection

1.1.1 Introduction

Choosing a diverse representative subset of items from a given large set arises in many settings such as feature selection [BMI13], sensor placement [JB09], matrix sparsification [BSS12a, SS11], and column subset selection in numerical linear algebra [AB13]. In statistics, this subset selection problem captures a classical problem *Optimal Design* in statistics, also known as design of experiments [Fed72, Puk06]. Its real-world applications include efficient design of science experiments and CPU processors [WYS17], and material design [WUSK18]. In order to motivate the mathematical formulation of this problem, we first outline the motivation from the optimal design problem. We later present several applications of the mathematical formulation in the related work section.

Motivation and Problem Formulation from Optimal Design. In many settings of supervised learning, obtaining labels is costly, but analysts have an ability choose from the pool of datapoints from which labels are obtained, also known as an active learning setting. The problem of *optimal design* is to choose the best smaller set of datapoints to obtain labels to maximize the accuracy and confidence of the model that learns from those labelled datapoints. The standard form of optimal design concerns linear regression model, which is arguably the most fundamental concept in supervised machine learning.

Optimal design can be defined mathematically as follows. Let $v_1, v_2, \ldots, v_n \in \mathbb{R}^d$ be given unlabelled datapoints. We assume a linear regression model: there exists an unknown regression coefficient vector $\mathbf{x}^* \in \mathbb{R}^d$ such that, for any $i \in [n]$, the label y_i received from the datapoint v_i satisfies

$$\mathbf{y}_i = {v_i}^\top \mathbf{x}^* + \eta$$

where η_i is a random i.i.d. Gaussian noise. Our goal of optimal design is to approximate $\mathbf{x}^*$ with least amount of error. We are allowed to choose at most k of the design points $S \subseteq [n]$ to observe $y_i = v_i \cdot \mathbf{x}^* + \eta_i$ for each $i \in S$.

Suppose we have picked a subset $S \subseteq [n]$ of size k. Let V_S be a d-by-k matrix whose columns are v_i's, $i \in S$ and $\mathbf{y}_S$ be the column vector of y_i's, $i \in S$. The best unbiased estimator $\hat{\mathbf{x}}$ for $\mathbf{x}^*$ is the least square error estimator:

$$\hat{\mathbf{x}} = \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^d} \|\mathbf{y}_S - V_S^\top \mathbf{x}\|_2^2$$

which has a closed-form solution

$$\hat{\mathbf{x}} = (V_S V_S^\top)^{-1} \sum_{i \in S} y_i v_i$$

Suppose η_i's are i.i.d. Gaussian noise with $\eta_i \sim N(0, \delta)$, then $\hat{\mathbf{x}} - \mathbf{x}^*$ is distributed as d-dimensional Gaussian $\delta N(0, (V_S V_S^\top)^{-1})$. The matrix $\Sigma = (V_S V_S^\top)^{-1}$ characterizes the error of the estimate, and thus the goal is to minimizes Σ. Multiple criteria are proposed to minimize Σ. Some of the common ones are *A*-, *D*-, and *E*-optimal designs, whose objectives are to minimize $\operatorname{tr} \Sigma, \det(\Sigma), \lambda_{\max}(\Sigma) = \|\Sigma\|_{\text{spec}}$, respectively. Therefore, optimal design can be stated as a discrete optimization problem:

$$\min_{S \subseteq [n], |S| = k} f\left((V_S V_S^\top)^{-1}\right) \tag{1.1}$$

for a given criterion f of interest.

Similarly to variants of the objectives, one may generalize to obtain variants of constraints beyond the cardinality constraint $|S| = k$. For example, each datapoint v_i's belong to an experiment in one of m laboratories, and each laboratory has its own size budget k_i. This is *partitioning constraint*, where v_i's are partitioned into m sets, each of which has its own cardinality constraint.

Though optimal design is motivated from statistics, the optimization (1.1) is general enough to capture many problems in other areas including in graph and network design, welfare economy, and diversity. We provide more details in the related work section.

Previous results. It is known that optimal design for D, E criteria is NP-hard [ÇMI09]. As a result, the work focuses on efficient approximation algorithms, both randomized and deterministic, for solving optimal design. Previous approaches to optimal design in statistics have no strong theoretical guarantees (only guarantee with approximation ratio depending on n [AB13] exists). Existing common approaches studied in theory and used practice include local search heuristics, such as Federov exchange [Fed+55], and approximate design which solves the continuous relaxation of the problem and uses heuristics rounding. Recently, a new perspective to optimal design problem through a more sophisticated randomized rounding algorithm gave a reasonable approximation ratio guarantee within a polynomial running time [WYS17, ALSW17a].

1.1.2 Other Applications of Subset Selection and Related Work

As mentioned earlier, subset selection not only applies to optimal design, but also many other problems. This section lists some of those applications and related topics in some details.

Welfare economics of indivisible goods. There are t indivisible items to be distributed among d individuals, and the utility of item i to person j is $p_{i,j}$. The utility u_j of person j is the sum of utilities of items person j receives. One criteria to distribute items is to maximize the project of u_j's, as known as Nash social welfare [KN79]. The other is to maximize the minimum of u_j's, also known as Santa Claus problem [BS06]. Both Nash social welfare and Santa Claus problems are special cases of D- and E-optimal designs with partitioning constraints, where each item is one partition containing d vectors in each of the d axes, and the budget for each partition is one.

Graph sparsification. Given a graph $G = (V, E)$, graph sparsification is the problem of finding a subset $S \subseteq E$ of size at most k which retains the values of all graph cuts [BSS12a, SS11]. This is closely related to E-optimal design where one wants to maximize the minimum eigenvalue of the sum of rank-1 matrices $v_i v_i^\top$, $\lambda_{\min}\left(\sum_{i \in S} v_i v_i^\top\right)$. To relate E-optimal to graph sparsification, one can define an instance of E-optimal with input vectors as $e_{ij}, \ \{i,j\} \in E$. We note that there are two stark differences of two problems: we require *unweighted* selection of edges in E-optimal design, and that graph sparsification requires two-sided bound of eigenvalues of $\left(\sum_{i \in S} v_i v_i^\top\right)$.

Network design. Similar to graph sparsification, we are given a graph $G = (V, E)$ which corresponds to an instance of an optimal design problem with input vectors as e_{ij} 's. We want to pick a subset of edges $F \subseteq E$ so that the subgraph $H = (V, F)$ is well-connected. To maximize the connectivity, one measure is *effective resistance* [GBS08, SS11], a notion of connectivity in the middle between the two notions of edge connectivity and shortest path distance. The effective resistance in an electric circuit corresponds to the A-optimal objective [GBS08]. There is also another notion of connectivity, which is to maximize the number of spanning tree in the subgraph H (see [LPYZ19] and references in the work for other applications). This notion corresponds to maximizing the determinant of the covariance matrix of selected vector, i.e. the D-optimal design problem.

Diversity sampling. Intuitively, the analyst in the optimal design setting seeks to find a small set of datapoints that spreads over a wide region of space in order to maximize learning over the entire space. Optimal design naturally gives rise to a notion of diversity sampling, where one seeks to maximize the diversity of a smaller set from a given pool of items. Diversity sampling has many

Table 1.1: Summary of approximation ratios of optimal design of previous work and our work. Cells with an asterisk * indicates our results that improve the previous ones. No integrality gap result exists before our work.

Problems	Previous work		Our work			Lower bound (for general k, d)
	By relaxation	Combinatorial	By relaxation	Combinatorial	Integrality gaps	
A-optimal, k close to d	N/A	$\frac{n-d+1}{k-d+1}$	k [NST19]*	N/A	$\frac{k}{k-d+1}$ [NST19]*	$1+c$ for some small c [NST19]*
A-optimal, $k >> d$	$1+\epsilon$, for $k \geq \Omega\left(\frac{d}{\epsilon^2}\right)$	$\frac{n-d+1}{k-d+1}$	$1+\epsilon$, for $k \geq \Omega\left(\frac{d}{\epsilon}+\frac{\ln\frac{1}{\epsilon}}{\epsilon^2}\right)$ [NST19]*	$1+\epsilon$, for $k \geq \Omega\left(\frac{d\ln^2\frac{1}{\epsilon}}{\epsilon^3}\right)$ [MSTX19]*	$1+\epsilon$, for $k \geq \Omega\left(\frac{d}{\epsilon}\right)$ [NST19]*	
D-optimal, $k >> d$	$1+\epsilon$, for $k \geq \Omega\left(\frac{d}{\epsilon}+\frac{\ln\frac{1}{\epsilon}}{\epsilon^2}\right)$	$\frac{n-d+1}{k-d+1}$	$1+\epsilon$, for $k \geq \Omega\left(\frac{d}{\epsilon}+\frac{\ln\frac{1}{\epsilon}}{\epsilon^2}\right)$ [NST19]	$1+\epsilon$, for $k \geq \Omega\left(\frac{d}{\epsilon}\right)$ [MSTX19]*	N/A	$\frac{3}{2\sqrt{2}}$ [ÇMI09]
E-optimal, $k >> d$	$1+\epsilon$, for $k \geq \Omega\left(\frac{d}{\epsilon^2}\right)$	N/A	N/A	N/A	$1+\epsilon$, for $k \geq \Omega\left(\frac{d}{\epsilon^2}\right)$ [NST19]*	$\left(\frac{3}{2}\right)^{\frac{1}{2(k-1)}}$ [ÇMI09]

connections with machine learning, such as determinantal point processes (DPPs) [KT+12] and fair representation of the data in machine learning [CDKV16].

1.1.3 Summary of Contributions

The work in this direction consists of two papers: proportional volume sampling with Mohit Singh and Aleksandar Nikolov [NST19] and combinatorial algorithms for optimal design with Mohit Singh, Vivek Madan, and Weijun Xie [MSTX19]. Results from both papers are summarized in Table 1. The first work [NST19] focuses on *A*-optimal design, yet we also show its applicability to *D*-design and integrality gap of *E*-design. The second [MSTX19] shows approximation factors for *A*- and *D*-optimal design problems. The bound for D-design is better than A-design, and is the curerntly best known. The integrality gap in Table 1 refers to the worst-case ratio between optimum of the relaxation and (1.1), where the relaxation refers to relaxing the space of solution $S \subseteq [n], |S| = k$ in (1.1) by $\pi \in \mathbb{R}^n, 1 \geq \pi \geq 0, \sum_{i=1}^n \pi_i = k$ and replacing V_S with $\sum_{i=1}^n \pi_i v_i v_i^\top$.

Proportional volume sampling. To solve the optimal design problem, [NST19] first solves the natural relaxation of optimal design, then use the solution of the relaxation to define a novel distribution called *proportional volume sampling*. Sampling from this distribution provably obtained the best approximation ratio for *A*-optimal design and best-known ratio for *D*-optimal design for $k >> d$, and the k-approximation for any $k \geq d$. [NST19] does not improve approximation guarantee on *E*-optimal, but shows a tight integrality gap result which implies that any rounding algorithm based on natural relaxation cannot improve upon the previous work. Additionally, [NST19] also shows integrality gap and NP-hardness of A-optimal design.

Combinatorial algorithms. In [MSTX19], we give the first approximation guarantees which is independent of n for optimal design with combinatorial algorithms, i.e. algorithms that do not rely on solving the convex relaxation of optimal designs. The approximation ratio proven for D-optimal also is the best proven in the literature. This work gives theorectical underpinning of known simple heuristics [Fed+55] which are observed to work well in practice. The heuristics also avoid solving the convex relaxation, which in practice is observed to be the bottleneck compared to the existing rounding schemes [ALSW17a].

1.2 Fair Dimensionality Reduction

1.2.1 Introduction

Fairness in machine learning is a recent growing area in computer science. There are many instances of machine learning algorithms' outputs that are perceived as biased or unfair by users. For example, Google Photos returns queries for CEOs with images overwhelmingly male and white [KMM15]; record advertisements with higher frequency than searches white names [Swe13];

facial recognition has wildly different accuracy for white men than dark-skinned women [BG18]; and recidivism prediction software labels low-risk African Americans as high-risk at higher rates than low-risk white people [ALMK18].

There are many speculations on the source of bias. The past literature focuses on either bias in training data or in the algorithms (see a survey, [CR18], for example). We (Jamie Morgenstern, Samira Samadi, Mohit Singh, and Santosh Vempala, and I) discover another source of bias: in the data processing [Sam+18]. Using one of the most common prepossessing algorithm PCA (Principle Component Analysis, [Pea01, Jol86, Hot33, RSA99, IP91]), we show the gap of PCA's performance between majority and minority groups in real datasets. This gap persists even after reweighting the groups to have equal weights.

[Sam+18] and [Tan+19] propose a *fair dimensionality reduction* problem, which seeks to resolve the bias found. The problem can stated as follows.

Definition 1.2.1. (*Fair dimensionality reduction*) Given m data points in $\mathbb{R}^n$ with subgroups A and B, the *fair PCA* problem with social welfare objective f is to find low-rank data U by optimizing

$$\min_{U\in\mathbb{R}^{m\times n},\,\text{rank}(U)\leq d} f\big(\frac{1}{|A|}\|A-U_A\|_F^2, \frac{1}{|B|}\|B-U_B\|_F^2\big) \tag{1.2}$$

where U_A and U_B are matrices with rows corresponding to rows of U for groups A and B respectively.

The choice of f depends on the context. One natural choice is to let f be the max of two reconstruction errors, which equalizes the error to both groups. Also, the problem can be naturally generalized to more than two groups when U is partitioned into k parts and f has k arguments.

1.2.2 Related Work

This line of work is new, and therefore has minimal related work comparable to our work. However, related work that are helpful in developing our algorithms are listed in Summary of Contribution.

1.2.3 Summary of Contribution

Both [Sam+18] and [Tan+19] develop algorithms for approximately solving *fair dimensionality reduction* for a wide class of functions f. We summarize the algorithms and results as follows.

Convex relaxation and LP rounding.

[Sam+18] solves the convex relaxation of fair dimensionality reduction problem for $f(u_A, u_B) = \max\{au_A + \alpha, bu_B + \beta\}$ for real constants a, b, α, β. The technique relies on solving the convex relaxation of the problem, defining a polytope whose objective are guarantee to perform as good as the optimal, then rounding the fractional solution to the extreme point of that polytope. Using the property of duals of an LP, the solution is guaranteed to perform as good as optimum, and violates the rank constraint by at most one dimension.

Convex relaxation and SDP rounding.

[Tan+19] generalizes and improves the theorectical guarantee of [Sam+18] to solving any f for k groups that is concave and decreasing in each group's reconstruction error. The technique also replies on convex relaxation, and then defining a semi-definite cone instead of a polytope than maintains the objective value. We build on the low-rank property of extreme solution of an SDP by [Pat98] and show that the solution is guaranteed to perform as good as optimum, and violates the rank constraint by at most $\lfloor\sqrt{2k + \frac{1}{4}} - \frac{3}{2}\rfloor$ dimension. In particular, in the case of two groups, we can solve the problem exactly.

[Tan+19] also generalizes iterative LP rounding [LRS11] to iterative SDP rounding and applies the result to fair dimensionality reduction. Additionally, [Tan+19] discusses some complexity results including NP-hardness and integrality gap of the convex relaxation formulation for the dimensionality reduction problem.

1.2.4 Fast implementations

Running SDP becomes slow when the number of original dimensions n increases beyond moderate sizes (e.g. $n \approx 50 - 100$). We consider two alternative algorithms to solving SDP: multiplicative weight update (MW) and Frank Wolfe (FW).

MW has been considered and analyzed in [Sam+18] for solving this SDP. By the regret analysis from online learning theory [AHK12], the runtime of MW is $O(\frac{\log k}{\epsilon^2})$ iterations of standard PCA for k groups and for a given desired error bound $\epsilon > 0$. In practice, MW can be tuned to obtain a runtime of $O(1)$ iterations of standard PCA, showing that incorporating fairness to certain farness criteria to which MW applies costs only a constant runtime overhead. In this thesis, we obtain MW through convex duality, obtaining a primal-daual algorithm which gives a duality gap and hence a stopping condition with an error bound. We expand the experiments from [Sam+18] to large datasets to demonstrate effectiveness of MW. We also propose FW which performs better for differentiable fairness objective than MW. Both algorithms can be tuned by using a more aggressive learning rate, giving heuristics to solving SDP that are much faster than a standard SDP solver in practice.

1.2.5 Experiments

We run our algorithms on two real datasets: Default Credit data [YL09] and Adult Income data [UC]. We evaluate the performance of PCA solutions based on two fairness criteria motivated from welfare economics. Our results show that our algorithms are significantly better based on both criteria of fairness than standard PCA. The experiment details can be found in Section 5.8.

We also show how two heuristics, MW and FW, scale to a large dataset in practice on several fairness criteria on multiple groups. We show their efficiency on Census data [AA], which have more than 600,000 datapoints partitioned into 16 groups and lie in thousands of dimensions. The details can be found in Section 5.9. The experiments and heuristics are publicly available at:

https://github.com/SDPforAll/multiCriteriaDimReduction.

1.3 Future Directions

1.3.1 Generalized Linear Models

Generalized Linear Models (GLM) generalizes linear regression model where the mean of the response y is not a linear function of features $v_i^\top \mathbf{x}^*$ but another function, called *mean* function μ, that depends on $v_i^\top \mathbf{x}^*$. The function that relates the mean μ back to $v_i^\top \mathbf{x}^*$ is called a *link* function, denoted by $g(\mu)$. An example of GLM is logistics regression, where the mean is $\mu = \frac{1}{1+e^{-v_i^\top \mathbf{x}^*}}$ and the link function is $g(\mu) = \log\left(\frac{\mu}{1-\mu}\right)$.

Optimal design for GLM has been studied; e.g. see [SY12]. The optimal design objective is

$$\min_{S\subseteq[n],|S|=k} f\left((V_S W_{\mathbf{x}^*} V_S^\top)^{-1}\right) \tag{1.3}$$

where $W_{\mathbf{x}^*}$ is a diagonal matrix with entries w_i for $i = 1, \ldots, n$ are weighting on each example i. The challenge is that the weighting depends not only on the model (mean and link function) and input vector v_i which are constants as an input to the problem, but also on $\mathbf{x}^*$ which is unknown. If $\mathbf{x}^*$ was known, then one can scale the vector $v_i \mapsto \sqrt{\mathbf{x}_i^*} v_i$ and apply the standard optimal design algorithm. This gives rise to several ways to optimize (1.3).

One is to assume a prior, such as Gaussian centered at mean zero, on $\mathbf{x}^*$. This is also known as Bayesian experimental design, and reduces to optimal design with ℓ_2 regularizer. We treat this topic in more details, including giving our guarantee, in Section 3.9.

One may also assume some boundary conditions on $\mathbf{x}^*$, e.g. lying inside a box $[-M, M]^d$ for some real $M > 0$. Then, one can solve (1.3) as a robust optimization: optimize (1.3) over the worst-case $\mathbf{x}^*$ in this boundary of $\mathbf{x}^*$.

Another way is to estimate $\mathbf{x}^*$ and use the estimate to approximately solve (1.3). Then, use the responses from the design to get a better estimate of $\mathbf{x}^*$. This is similar to alternative minimization where one tries to minimize (1.3) with respect to subsets S, and then to $\mathbf{x}^*$, and continue alternatively.

Besides Bayesian optimal design which reduces to an optimal design with regularizer, it is unknown if good approximation exists, and whether our approach in this thesis can lead to any theoretical results. This remains an open direction in the future.

1.3.2 Ridge Regression

We showed that our algorithm can solve an objective motivated from expected square loss of ridge regression (see Inequality (3.47) and its derivation). However, the objective is motivated from an assumption on the regularizer parameter λ and is an upper bound of the true square loss, not exact. It remains an open question whether an approximation algorithm exists for general λ for minimizing the square loss of ridge regression directly.

1.3.3 Other Applications

To show approximability of *A*-optimal design, we show an efficient implementation of proportional volume sampling and its generalization for a large class of parameters (Theorem 3.6.2 and other results in Section 3.6 in general, and regularized version in Section 3.9.7). It remains an open question what other problems the efficiency of these classes of sampling may be applicable to.

CHAPTER 2
PRELIMINARIES

To show approximation results and lower bounds of approximations for diverse subset selections problems, we utilize a technique of relaxation and duality. Here we list the applications of relaxation sand integrality gaps to A- and D-optimal design problems. In general, it is crucial that the relaxations are convex problems to allow for efficient solvability of the relaxations. It is known that relaxations A- and D-optimal design are convex ([BV04]). We note that though the objective of D-optimal is not concave, its logarithm is. Hence, the relaxation is still efficiently solvable.

For details of convex duality in general, we refer readers to [BV04].

2.1 Convex Relaxation and its Dual of *A*-optimal Design

We first consider the convex relaxation for the A-optimal design problem given below for the settings without and with repetitions. The difference for without repetition is that there is no upper bound on the value of x_i (the same is true to other optimal design criteria). This relaxation is classical, and already appears in, e.g. [Che52]. It is easy to see that the objective $\operatorname{tr}\left(\sum_{i=1}^{n} x_i v_i v_i^{\top}\right)^{-1}$ is convex ([BV04], section 7.5).

(a)

$$A\text{-}\textbf{Rel}(V)$$

$$\begin{array}{lll} \min\limits_{x \in \mathbb{R}^n} & \operatorname{tr}\left(\left(\sum_{i=1}^{n} x_i v_i v_i^{\top}\right)^{-1}\right) & \\ & \sum_{i=1}^{n} x_i \leq k & \\ & x_i \geq 0 & i \in [n] \end{array}$$

(b)

$$A\text{-}\textbf{Rel-Dual}(V)$$

$$\begin{array}{lll} \max\limits_{\substack{\lambda \in \mathbb{R}^n \\ Y \in \mathbb{R}^{d \times d}}} & 2 \operatorname{tr}\left(Y^{1/2}\right) - k\lambda & \\ & \lambda - v_i^{\top} Y v_i \geq 0 & i \in [n] \\ & Y \succeq 0 & \end{array}$$

(c) Convex Relaxation and its Dual for the A-DESIGN problem

With Repetition	**Without Repetition**
$\min \operatorname{tr}\left(\sum_{i=1}^{n} x_i v_i v_i^{\top}\right)^{-1}$	$\min \operatorname{tr}\left(\sum_{i=1}^{n} x_i v_i v_i^{\top}\right)^{-1}$ (2.1)
s.t. $\sum_{i=1}^{n} x_i = k$	s.t. $\sum_{i=1}^{n} x_i = k$ (2.2)
$0 \leq x_i \quad \forall i \in [n]$	$0 \leq x_i \leq 1 \quad \forall i \in [n]$ (2.3)

Let us denote the optimal value of (2.1)–(2.3) by CP. By plugging in the indicator vector of an optimal integral solution for x, we see that CP $\leq$ OPT, where OPT denotes the value of the optimal solution. We also present the dual for A-optimal design in Figure 2.1c

2.2 Convex Relaxation of *D*-optimal Design

We describe the relaxation of D-design with repetitions in Figure 2.2c below. Let OPT denote the be the common optimum value of (D-REL) and its dual (D-REL-DUAL). Let $I^\star$ denote the indices of the vector in the optimal solution and let $\phi^{\mathsf{D}} = \det\left(\sum_{i \in I^\star} v_i v_i^\top\right)^{\frac{1}{d}}$ be its objective. Then we have that $\phi_f^{\mathsf{D}} \geq \log \phi^{\mathsf{D}}$ by plugging in the indicator vector of an optimal integral solution for x to the relaxation. We also present the convex relaxation and its dual without repetition in Figure 2.3c.

(a)

$$\begin{array}{llll} \max\limits_{x \in \mathbb{R}^n} & \frac{1}{d} \log\det\left(\sum\limits_{i=1}^{n} x_i v_i v_i^\top\right) & & \\ & \sum\limits_{i=1}^{n} x_i & \leq k & \\ & x_i & \geq 0 & i \in [1,n] \end{array}$$

(b)

$$\begin{array}{llll} \min\limits_{\substack{\mu \in \mathbb{R} \\ Y \in \mathbb{R}^{d \times d}}} & \frac{1}{d} \log\det(Y) + \frac{k}{d}\mu - 1 & & \\ & \mu - v_i^\top Y^{-1} v_i & \geq 0 & i \in [1,n] \\ & Y & \succeq 0 & \end{array}$$

(c) Convex Relaxation and its Dual for the D-DESIGN problem

2.3 Integrality Gaps

In this thesis, we use *integrality gaps* to show tightness of approximation and to show approximation factor without solving the convex relaxation, as known as dual-fitting. We refer readers to [Vaz13] for more examples and significance of integrality gaps, including more uses of dual-fitting techniques in approximation algorithms.

(a)
$$\begin{array}{llll} \max & \frac{1}{d}\log\det\left(\sum_{i=1}^{n} x_i v_i v_i^\top\right) & & \\ & \sum_{i=1}^{n} x_i & \leq k & \\ & 1 \geq \; x_i & \geq 0 & i \in [1,n] \end{array}$$

(b)
$$\begin{array}{llll} \min & \frac{1}{d}\log\det(Y) + \frac{k}{d}\mu + \frac{1}{d}\sum_{i=1}^{n}\eta_i - 1 & & \\ & \mu + \eta_i - v_i^\top Y^{-1} v_i & \geq 0 & i \in [1,n] \\ & \eta_i & \geq 0 & i \in [1,n] \\ & Y & \succeq 0 & \end{array}$$

(c) Convex Relaxation and its Dual for the D-DESIGN problem without repetitions

2.3.1 Tightness of Approximations.

To show tightness of approximation, we show the lower bound of approximation by showing the *integrality gaps* of relaxations. In particular, if the relaxation of a problem has integrality gap α, any rounding method from the relaxation will achieve an approximation no better than factor α. We use this to show the tightness of relaxation of E-optimal design and lower-bound of approximation of *A*-optimal design in this thesis. We also show that the relaxation of Fair Dimensionality Reduction is not tight through the existence of the gap.

2.3.2 Dual-Fitting

Convex relaxations have their corresponding *dual* problems. For most well-behaved convex problems, strong duality holds, i.e. the optimum of dual equals the primal optimum. Hence, one can use the dual feasible (and not necessarily optimal) solutions to bound the primal optimum. Here we give the primal and dual of relaxations of *A*- and *D*-optimal designs problems, and note that strong duality holds in both cases.

2.4 Local Search and Greedy Algorithms

The *local search* algorithm to maximize an objective f over sets $S \in \mathcal{P}$ where $\mathcal{P}$ is a given collection of sets of the same size is the algorithm that starts with any initial feasible solution in $\mathcal{P}$. Then, in each step, it checks if there is any swap of an element to increase f, i.e. to delete on current element in S and add another one such that f increases and the new set remains in $\mathcal{P}$.

The *greedy* algorithm to maximize an objective f over sets $S \in \mathcal{P}$ where $\mathcal{P}$ is a given collection of sets starts with any initial feasible solution in $\mathcal{P}$ (usually an empty set). In each step, the algorithm adds an element that increases f by the highest amount. The algorithm stops when it reaches a terminating condition to remain feasible. For example, if P consists of sets of size at most k, then one terminates when the set reaches size k.

Local search and greedy algorithms are among the most basic combinatorial algorithms. They do not necessarily have theoretical guarantees, yet sometimes the algorithms (or their modifications) do to some problems despite their simplicity. We refer readers to [Vaz13, CLRS09] for examples of their applications in approximation algorithms.

CHAPTER 3

SAMPLING-BASED APPROXIMATION ALGORITHM FOR SUBSET SELECTION

3.1 Introduction

Given a collection of vectors, a common problem is to select a subset of size $k \leq n$ that *represents* the given vectors. To quantify the representability of the chosen set, typically one considers spectral properties of certain natural matrices defined by the vectors. Such problems arise as experimental design ([Fed72, Puk06]) in statistics; feature selection ([BMI13]) and sensor placement problems ([JB09]) in machine learning; matrix sparsification ([BSS12a, SS11]) and column subset selection ([AB13]) in numerical linear algebra. In this work, we consider the optimization problem of choosing the representative subset that aims to optimize the A-*optimality criterion* in experimental design.

Experimental design is a classical problem in statistics ([Puk06]) with recent applications in machine learning ([JB09, WYS16]). Here the goal is to estimate an unknown vector $w \in \mathbb{R}^d$ via linear measurements of the form $y_i = v_i^\top w + \eta_i$ where v_i are possible experiments and η_i is assumed to be small i.i.d. unbiased Gaussian error introduced in the measurement. Given a set S of linear measurements, the maximum likelihood estimate $\hat{w}$ of w can be obtained via a least squares computation. The error vector $w-\hat{w}$ has a Gaussian distribution with mean 0 and covariance matrix $\left(\sum_{i\in S} v_i v_i^\top\right)^{-1}$. In the optimal experimental design problem the goal is to pick a cardinality k set S out of the n vectors such that the measurement error is minimized. Minimality is measured according to different criteria, which quantify the "size" of the covariance matrix. In this thesis, we study the classical A-optimality criterion, which aims to minimize the average variance over directions, or equivalently the trace of the covariance matrix, which is also the expectation of the squared Euclidean norm of the error vector $w - \hat{w}$.

Table 3.1: Summary of approximation ratios of A-optimal results. We list the best applicable previous work for comparison. The cells with asterisk * indicate that the ratios are tight with matching integrality gap of the convex relaxation (2.1)-(2.3).

Problem	Our result	Previous work
Case $k = d$	d *	$n - d + 1$ ([AB13])
Asymptotic $k >> d$ without Repetition	$1 + \epsilon$, for $k \geq \Omega\left(\frac{d}{\epsilon} + \frac{\log 1/\epsilon}{\epsilon^2}\right)$	$1 + \epsilon$, for $k \geq \Omega\left(\frac{d}{\epsilon^2}\right)$ ([ALSW17a])
Arbitrary k and d With Repetition	$\frac{k}{k-d+1}$ *	$n - d + 1$ ([AB13])
Asymptotic $k >> d$ With Repetition	$1 + \epsilon$, for $k \geq d + \frac{d}{\epsilon}$ *	$1 + \epsilon$, for $k \geq \Omega(\frac{d}{\epsilon^2})$ ([ALSW17a])

We let V denote the $d \times n$ matrix whose columns are the vectors $v_1, \ldots, v_n$ and $[n] = \{1, \ldots, n\}$. For any set $S \subseteq [n]$, we let V_S denote the $d \times |S|$ submatrix of V whose columns correspond to vectors indexed by S. Formally, in the A-optimal design problem our aim is to find a subset S of cardinality k that minimizes the trace of $(V_S V_S^\top)^{-1} = \left(\sum_{i \in S} v_i v_i^\top\right)^{-1}$. We also consider the A-optimal design problem with repetitions, where the chosen S can be a multi-set, thus allowing a vector to chosen more than once.

Apart from experimental design, the above formulation finds application in other areas such as sensor placement in wireless networks ([JB09]), sparse least squares regression ([BDM11]), feature selection for k-means clustering ([BMI13]), and matrix approximation ([AB13]). For example, in matrix approximation ([HM07, HM11, AB13]) given a $d \times n$ matrix V, one aims to select a set S of k such that the Frobenius norm of the Moore-Penrose pseudoinverse of the selected matrix V_S is minimized. It is easy to observe that this objective equals the A-optimality criterion for the vectors given by the columns of V.

3.1.1 Our Contributions and Results

Our main contribution is to introduce the *proportional volume sampling* class of probability measures to obtain improved approximation algorithms for the A-optimal design problem. We obtain improved algorithms for the problem with and without repetitions in regimes where k is close to d as well as in the asymptotic regime where $k \geq d$. The improvement is summarized in Table 3.1. Let $\mathcal{U}_k$ denote the collection of subsets of $[n]$ of size exactly k and $\mathcal{U}_{\leq k}$ denote the subsets of $[n]$ of size at most k. We will consider distributions on sets in $\mathcal{U}_k$ as well as $\mathcal{U}_{\leq k}$ and state the following definition more generally.

Definition 3.1.1. Let μ be probability measure on sets in $\mathcal{U}_k$ (or $\mathcal{U}_{\leq k}$). Then the proportional volume sampling with measure μ picks a set $S \in \mathcal{U}_k$ (or $\mathcal{U}_{\leq k}$) with probability proportional to $\mu(S)\det(V_S V_S^\top)$.

Observe that when μ is the uniform distribution and $k \leq d$ then we obtain the standard volume sampling where one picks a set S proportional to $\det(V_S V_S^\top)$, or, equivalently, to the volume of the parallelopiped spanned by the vectors indexed by S. Volume sampling measure was first introduced by [DRVW06] for low-rank matrix matrix approximation (with optimal guarantee in [DV06]) It has received much attention, and efficient algorithms are known for sampling from it ([DR10, GS12]). More recently, efficient algorithms were obtained even when $k \geq d$ ([LJS17, SX18]). We discuss the computational issues of sampling from proportional volume sampling in Lemma 3.1.10 and Section 3.6.2.

Our first result shows that approximating the A-optimal design problem can be reduced to finding distributions on $\mathcal{U}_k$ (or $\mathcal{U}_{\leq k}$) that are *approximately independent*. First, we define the exact formulation of approximate independence needed in our setting.

Definition 3.1.2. Given integers $d \leq k \leq n$ and a vector $x \in [0,1]^n$ such that $1^\top x = k$, we call a measure μ on sets in $\mathcal{U}_k$ (or $\mathcal{U}_{\leq k}$), α-approximate $(d-1,d)$-wise independent with respect to x if for any subsets $T, R \subseteq [n]$ with $|T| = d-1$ and $|R| = d$, we have

$$\frac{\mathsf{Pr}_{\mathcal{S}\sim\mu}[T \subseteq \mathcal{S}]}{\mathsf{Pr}_{\mathcal{S}\sim\mu}[R \subseteq \mathcal{S}]} \leq \alpha \frac{x^T}{x^R}$$

where $x^L := \prod_{i \in L} x_i$ for any $L \subseteq [n]$. We omit “with respect to x" when the context is clear.

Observe that if the measure μ corresponds to picking each element i independently with probability x_i, then $\frac{\Pr_{\mathcal{S}\sim\mu}[T\subseteq\mathcal{S}]}{\Pr_{\mathcal{S}\sim\mu}[R\subseteq\mathcal{S}]} = \frac{x^T}{x^R}$. However, this distribution has support on all sets and not just sets in $\mathcal{U}_k$ or $\mathcal{U}_{\leq k}$, so it is not allowed by the definition above.

Our first result reduces the search for approximation algorithms for A-optimal design to construction of approximate $(d-1,d)$-wise independent distributions. This result generalizes the connection between volume sampling and A-optimal design established in [AB13] to proportional volume sampling, which allows us to exploit the power of the convex relaxation and get a significantly improved approximation.

Theorem 3.1.3. *Given integers $d \leq k \leq n$, suppose that for any a vector $x \in [0,1]^n$ such that $1^\top x = k$ there exists a distribution μ on sets in $\mathcal{U}_k$ (or $\mathcal{U}_{\leq k}$) that is α-approximate $(d-1,d)$-wise independent. Then the proportional volume sampling with measure μ gives an α-approximation algorithm for the A-optimal design problem.*

In the above theorem, we in fact only need an approximately independent distribution μ for the optimal solution x of the natural convex relaxation for the problem, which is given in (2.1)–(2.3). The result also bounds the integrality gap of the convex relaxation by α. Theorem 3.1.3 is proved in Section 3.2.

Theorem 3.1.3 reduces our aim to constructing distributions that have approximate $(d-1,d)$-independence. One way to construct such distribution is through a general class of *hard-core distributions,* defined as follow(s).

Definition 3.1.4. We call a distribution μ on $\mathcal{U}_k$ (or $\mathcal{U}_{\leq k}$) a *hard-core* distribution with parameter $\lambda \in \mathbb{R}^n_+$ if $\mu(S) \propto \lambda^S := \prod_{i\in S}\lambda_i$ for each set in $\mathcal{U}_k$ (or $\mathcal{U}_{\leq k}$).

Convex duality implies that hard-core distributions have the maximum entropy among all distributions which match the marginals of μ ([BV04]). Observe that, while μ places non-zero probability on exponentially many sets, it is enough to specify μ succinctly by describing λ. Hard-core distributions over various structures including spanning trees ([GSS11]) or matchings ([Kah96, Kah00]) in a graph display *approximate independence* and this has found use in combinatorics as well as algorithm design. Following this theme, we show that certain hard core distributions on $\mathcal{U}_k$ and $\mathcal{U}_{\leq k}$ exhibit approximate $(d-1,d)$-independence when $k=d$ and in the asymptotic regime when $k >> d$.

Theorem 3.1.5. *Given integers $d \leq k \leq n$ and a vector $x \in [0,1]^n$ such that $1^\top x = k$, there exists a hard-core distribution μ on sets in $\mathcal{U}_k$ that is d-approximate $(d-1,d)$-wise independent when $k=d$. Moreover, for any $\epsilon > 0$, if $k = \Omega\left(\frac{d}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{1}{\epsilon}\right)$, then there is a hard-core distribution μ on $\mathcal{U}_{\leq k}$ that is $(1+\epsilon)$-approximate $(d-1,d)$-wise independent. Thus we obtain a d-approximation algorithm for the A-optimal design problem when $k=d$ and $(1+\epsilon)$-approximation algorithm when $k = \Omega\left(\frac{d}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{1}{\epsilon}\right)$.*

The above theorem relies on two natural hard-core distributions. In the first one, we consider the hard-core distribution with parameter $\lambda = x$ on sets in $\mathcal{U}_k$ and in the second we consider the hard-core distribution with parameter $\lambda = \frac{(1-\epsilon)x}{1-(1-\epsilon)x}$ (defined co-ordinate wise) on sets in $\mathcal{U}_{\leq k}$. We prove the theorem in Section 3.3.

Our techniques also apply to the A-optimal design problem with repetitions where we obtain an even stronger result, described below. The main idea is to introduce multiple, possibly exponentially many, copies of each vector, depending on the fractional solution, and then apply proportional volume sampling to obtain the following result.

Theorem 3.1.6. *For all $k \geq d$ and $0 < \epsilon \leq 1$, there is a $(\frac{k}{k-d+1} + \epsilon)$-approximation algorithm for the A-optimal design problem with repetitions. In particular, there is a $(1+\epsilon)$-approximation when $k \geq d + \frac{d}{\epsilon}$.*

We remark that the integrality gap of the natural convex relaxation is at least $\frac{k}{k-d+1}$ (see Section 3.7.2) and thus the above theorem results in an exact characterization of the integrality gap of the convex program (2.1)–(2.3), stated in the following corollary. The proof of Theorem 3.1.6 appears in Section 3.6.3.

Corollary 3.1.7. *For any integers $k \geq d$, the integrality gap of the convex program* (2.1)–(2.3) *for the A-optimal design with repetitions is exactly $\frac{k}{k-d+1}$.*

We also show that A-optimal design is NP-hard for $k = d$ and moreover, hard to approximate within a constant factor.

Theorem 3.1.8. *There exists a constant $c > 1$ such that the A-optimal design problem is NP-hard to c-approximate when $k = d$.*

The $k \leq d$ case. The A-optimal design problem has a natural extension to choosing fewer than d vectors: our objective in this case is to select a set $S \subseteq [n]$ of size k so that we minimize $\sum_{i=1}^{k} \lambda_i^{-1}$, where $\lambda_1, \ldots, \lambda_k$ are the k largest eigenvalues of the matrix $V_S V_S^\top$. While this problem no longer corresponds to minimizing the variance in an experimental design setting, we will abuse terminology and still call it the A-optimal design problem. This is a natural formulation of the geometric problem of picking a set of vectors which are as "spread out" as possible. If $v_1, \ldots, v_n$ are the points in a dataset, we can see an optimal solution as a maximally diverse representative sample of the dataset. Similar problems, but with a determinant objective, have been widely studied in computational geometry, linear algebra, and machine learning: for example the largest volume simplex problem, and the maximum subdeterminant problem (see [Nik15] for references to prior work). [ÇMI09] also studied an analogous problem with the sum in the objective replaced by a maximum (which extends E-optimal design).

While our rounding extends easily to the $k \leq d$ regime, coming up with a convex relaxation becomes less trivial. We do find such a relaxation and obtain the following result whose proof appears in Section 3.5.1.

Theorem 3.1.9. *There exists a $\operatorname{poly}(d, n)$-time k-approximation algorithm for the A-optimal design problem when $k \leq d$.*

Integrality Gap. Experimental design problems come with many different objectives including A, D, E, G, T, each corresponding to a different function of the covariance matrix of the error $w - \hat{w}$. A natural question is whether they all behave similarly in terms of approximation algorithms. Indeed, recent results of [ALSW17a, ALSW17b] and [WYS16] give the $(1+\epsilon)$-approximation algorithm in the asymptotic regime, $k \geq \Omega\left(\frac{d}{\epsilon^2}\right)$ and $k \geq \Omega\left(\frac{d^2}{\epsilon}\right)$, for many of these variants. In contrast, we show the *optimal bounds* that can be obtained via the standard convex relaxation are different for different objectives. We show that for the E-optimality criterion (in which we minimize the largest eigenvalue of the covariance matrix) getting a $(1+\epsilon)$-approximation with the natural convex relaxation requires $k = \Omega(\frac{d}{\epsilon^2})$, both with and without repetitions. This is in sharp contrast to results we obtain here for A-optimality. Thus, different criteria behave differently in terms of approximability. Our proof of the integrality gap (in Section 3.7.1) builds on a connection to spectral graph theory and in particular on the Alon-Boppana bound ([Alo86, Nil91]). We prove an Alon-Boppana style bound for the unnormalized Laplacian of not necessarily regular graphs with a given average degree.

Computational Issues. While it is not clear whether sampling from proportional volume sampling is possible under general assumptions (for example given a sampling oracle for μ), we obtain an efficient sampling algorithm when μ is a hard-core distribution.

Lemma 3.1.10. *There exists a* $\operatorname{poly}(d, n)$*-time algorithm that, given a matrix* $d \times n$ *matrix* V*, integer* $k \leq n$*, and a hard-core distribution* μ *on sets in* $\mathcal{U}_k$ *(or* $\mathcal{U}_{\leq k}$*) with parameter* λ*, efficiently samples a set from the proportional volume measure defined by* μ*.*

When $k \leq d$ and μ is a hard-core distribution, the proportional volume sampling can be implemented by the standard volume sampling after scaling the vectors appropriately. When $k > d$, such a method does not suffice and we appeal to properties of hardcore distributions to obtain the result. We also present an efficient implementation of Theorem 3.1.6 which runs in time polynomial in $\log(1/\epsilon)$. This requires more work since the basic description of the algorithm involves implementing proportional volume sampling on an exponentially-sized ground set. This is done in Section 3.6.3.

We also outline efficient deterministic implementation of algorithms in Theorem 3.1.5 and 3.1.6 in Section 3.6.2 and 3.6.4.

3.1.2 Related Work

Experimental design is the problem of maximizing information obtained from selecting subsets of experiments to perform, which is equivalent to minimizing the covariance matrix $\left(\sum_{i\in S} v_i v_i^\top\right)^{-1}$. We focus on A-optimality, one of the criteria that has been studied intensely. We restrict our attention to approximation algorithms for these problems and refer the reader to [Puk06] for a broad survey on experimental design.

[AB13] studied the A- and E-optimal design problems and analyzed various combinatorial algorithms and algorithms based on volume sampling, and achieved approximation ratio $\frac{n-d+1}{k-d+1}$. [WYS16] found connections between optimal design and matrix sparsification, and used these connections to obtain a $(1+\epsilon)$-approximation when $k \geq \frac{d^2}{\epsilon}$, and also approximation algorithms under certain technical assumptions. More recently, [ALSW17a, ALSW17b] obtained a $(1+\epsilon)$-approximation when $k = \Omega\left(\frac{d}{\epsilon^2}\right)$ both with and without repetitions. We remark that their result also applies to other criteria such as E and D-optimality that aim to maximize the minimum eigenvalue, and the geometric mean of the eigenvalues of $\sum_{i\in S} v_i v_i^\top$, respectively. More generally, their result applies to any objective function that satisfies certain regularity criteria.

Improved bounds for D-optimality were obtained by [SX18] who give an e-approximation for all k and d, and $(1+\epsilon)$-approximation algorithm when $k = \Omega(\frac{d}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{1}{\epsilon})$, with a weaker condition of $k \geq \frac{2d}{\epsilon}$ if repetitions are allowed. The D-optimality criterion when $k \leq d$ has also been extensively studied. It captures maximum a-posteriori inference in constrained determinantal point process models ([KT+12]), and also the maximum volume simplex problem. [Nik15], improving on a long line of work, gave a e-approximation. The problem has also been studied under more general matroid constraints rather than cardinality constraints ([NS16, AG17, SV17]).

[ÇMI09] also studied several related problems in the $k \leq d$ regime, including D- and E-optimality. We are not aware of any prior work on A-optimality in this regime.

The criterion of E-optimality, whose objective is to maximize the minimum eigenvalue of $\sum_{i \in S} v_i v_i^\top$, is closely related to the problem of matrix sparsification ([BSS12a, SS11]) but incomparable. In matrix sparsification, we are allowed to weigh the selected vectors, but need to bound both the largest and the smallest eigenvalue of the matrix we output.

The restricted invertibility principle was first proved in the work of [BT87], and was later strengthened by [Ver01], [SS10], and [NY17]. Spielman and Srivastava gave a deterministic algorithm to find the well-invertible submatrix whose existence is guaranteed by the theorem. Besides its numerous applications in geometry (see [Ver01] and [You14]), the principle has also found applications to differential privacy ([NTZ16]), and to approximation algorithms for discrepancy ([NT15]).

Volume sampling [DRVW06] where a set S is sampled with probability proportional to $\det(V_S V_S^\top)$ has been studied extensively and efficient algorithms were given by [DR10] and improved by [GS12]. The probability distribution is also called a *determinantal point process* (DPP) and finds many applications in machine learning ([KT+12]). Recently, fast algorithms for volume sampling have been considered in [DW17a, DW17b].

While NP-hardness is known for the D- and E-optimality criteria ([ÇMI09]), to the best of our knowledge no NP-hardness for A-optimality was known prior to our work. Proving such a hardness result was stated as an open problem in [AB13].

Restricted Invertibility Principle for Harmonic Mean. As an application of Theorem 3.1.9, we prove a new restricted invertibility principle (RIP) ([BT87]) for the harmonic mean of singular values. The RIP is a robust version of the elementary fact in linear algebra that if V is a $d \times n$ rank r matrix, then it has an invertible submatrix V_S for some $S \subseteq [n]$ of size r. The RIP shows that if

V has stable rank r, then it has a well-invertible submatrix consisting of $\Omega(r)$ columns. Here the stable rank of V is the ratio $(\|V\|_{HS}^2/\|V\|^2)$, where $\|\cdot\|_{HS} = \sqrt{\operatorname{tr}(VV^\top)}$ is the Hilbert-Schmidt, or Frobenius, norm of V, and $\|\cdot\|$ is the operator norm. The classical restricted invertibility principle ([BT87, Ver01, SS10]) shows that when the stable rank of V is r, then there exists a subset of its columns S of size $k = \Omega(r)$ so that the k-th singular value of V_S is $\Omega\left(\|V\|_{HS}/\sqrt{m}\right)$. [Nik15] showed there exists a submatrix V_S of k columns of V so that the geometric mean its top k singular values is on the same order, even when k equals the stable rank. We show an analogous result for the harmonic mean when k is slightly less than r. While this is implied by the classical restricted invertibility principle, the dependence on parameters is better in our result for the harmonic mean. For example, when $k = (1-\epsilon)r$, the harmonic mean of squared singular values of V_S can be made at least $\Omega\left(\epsilon\|V\|_{HS}^2/m\right)$, while the tight restricted invertibility principle of Spielman and Srivastava ([SS11]) would only give ϵ^2 in the place of ϵ. See Theorem 3.5.4 for the precise formulation of our restricted invertibility principle.

3.1.3 Problem Variants

We discuss several generalization of A-optimal objectives and corresponding modifications to the algorithms and our results in this chapter. We summarize this in Table 3.2. Here, $E_\ell(M)$ denote the elementary symmetric polynomial of degree ℓ of matrix M.

Our first variant is the case $k \leq d$, where we generalize d-approximation when $k = d$ to k-approximation when $k \leq d$. The objective is modified accordingly as k selected vectors span only in k and not d dimensions. Details can be found in Section 3.5.1.

We generalize A-optimal design objective to the generalized ratio objective, where its special case also has been considered by [MS17]. We show that all approximation results in A-optimal applies to this setting, with a better bound on k. This generalization includes D-optimal design, and hence proportional volume sampling also gives approximation algorithms for D-design. Summary of approximation results are in Table 3.3 in Section 3.5.3 which also includes details of this variant.

Table 3.2: Summary of variants of objectives for A-optimal design problems and their corresponding modifications of algorithms and approximation guarantees

Variants	Objectives	Sampling distributions	Modification of the results	Integrality gaps
Original	$\operatorname{tr}\left(\left(V_S V_S^\top\right)^{-1}\right)$	$\propto \mu(S)\det(V_S V_S^\top)$	N/A	$\frac{k}{k-d+1}$
$k \leq d$	$\frac{E_{k-1}\left(\sum_{i=1}^n x_i v_i v_i^\top\right)}{E_k\left(\sum_{i=1}^n x_i v_i v_i^\top\right)}$	$\propto \mu(S)E_k(V_S V_S^\top)$	k-approximation instead of d	k
Generalized ratio	$\frac{E_{l'}\left(V_S V_S^\top\right)}{E_l\left(V_S V_S^\top\right)}$	$\propto \mu(S)E_l(V_S V_S^\top)$	Replace d with l in the bounds of k	$\frac{k}{k-l'+1}$
Ridge	$\operatorname{tr}\left(\left(V_S V_S^\top + \lambda I\right)^{-1}\right)$	$\propto \mu(S)\det(V_S V_S^\top + \lambda I)$	$(1+\epsilon_\lambda)$-approx where $\epsilon_0 = \epsilon$ (same as original) and $\epsilon_\lambda \rightarrow 0$ as $\lambda \rightarrow \infty$	Not yet analyzed

Another variant is *ridge regression,* which motivates the objective of A-design with regularizer term added. We show that the main result of A-optimal design, namely the $(1+\epsilon)$-approximation without repetition for large k, generalizes to this setting and improves as the regularizer parameter increases. We have not attempted to generalize other results if A-optimal design in this chapter, though we suspect that similar analyses can be done. We also have not attempted to check the integrality gap in this setting. Details can be found in Section 3.9.

Finally, we note that each modified version can be implemented efficiently (including their deterministic derandomization counterpart).

3.1.4 Organization

In this chapter, we first show the reduction of A-optimal design problem to constructing an efficient α-approximate $(d-1,d)$-independent distribution μ, i.e. Theorem 3.1.3, in Section 3.2. We show d-approximation and asymptotically optimal approximation for A-optimal design without repetition in Section 3.3. We show approximations result when repetitions are allowed in Section

3.4. We discuss several generalizations of *A*-optimal design in Section 3.5. In Section 3.6, we provide efficient randomized and deterministic derandomization of proportional volume sampling with parameter μ for any hard-core measure μ. In Section 3.7, we show in integrality gaps results of A- and E-optimal design. In Section 3.8, we prove APX-hardness of *A*-optimal design. In Section 3.9, we discuss *A*-optimal design when an ℓ_2-regularizer, also known as ridge regression, and show that a modification of proportional volume sampling still acheives an approximation guarantee for this problem.

3.2 Approximation via Near Independent Distributions

In this section, we prove Theorem 3.1.3 and give an α-approximation algorithm for the A-optimal design problem given an α-approximate $(d-1,d)$-independent distribution μ.

We first consider the convex relaxation for the problem given below for the settings without and with repetitions. The relaxation is stated in Preliminaries at (2.1)–(2.3). Let us denote the optimal value of (2.1)–(2.3) by CP, and denote the value of the optimal solution by OPT. For this section, we focus on the case when repetitions are not allowed.

3.2.1 Approximately Independent Distributions

Let us use the notation $x^S = \prod_{i \in S} x_i$, V_S a matrix of column vectors $v_i \in \mathbb{R}^d$ for $i \in S$, and $V_S(x)$ a matrix of column vectors $\sqrt{x_i} v_i \in \mathbb{R}^d$ for $i \in S$. Let $e_k(x_1, \ldots, x_n)$ be the degree k elementary symmetric polynomial in the variables $x_1, \ldots, x_n$, i.e. $e_k(x_1, \ldots, x_n) = \sum_{S \in \mathcal{U}_k} x^S$. By convention, $e_0(x) = 1$ for any x. For any positive semidefinite $n \times n$ matrix M, we define $E_k(M)$ to be $e_k(\lambda_1, \ldots, \lambda_n)$, where $\lambda(M) = (\lambda_1, \ldots, \lambda_n)$ is the vector of eigenvalues of M. Notice that $E_1(M) = \mathrm{tr}(M)$ and $E_n(M) = \det(M)$.

To prove Theorem 3.1.3, we give the following algorithm $\mathcal{A}$ which is a general framework to sample S to solve the A-optimal design problem.

We first prove the following lemma which is needed for proving Theorem 3.1.3.

Algorithm 3.1 The proportional volume sampling algorithm

```
Given an input $V = [v_1, \ldots, v_n]$ where $v_i \in \mathbb{R}^d$, $k$ a positive integer, and measure $\mu$ on sets in $\mathcal{U}_k$ (or $\mathcal{U}_{\leq k}$).
Solve convex relaxation CP to get a fractional solution $x \in \mathbb{R}^n_+$ with $\sum_{i=1}^n x_i = k$.
Sample set $\mathcal{S}$ (from $\mathcal{U}_{\leq k}$ or $\mathcal{U}_k$) where $\mathsf{Pr}\left[\mathcal{S} = S\right] \propto \mu(S) \det(V_S V_S^\top)$ for any $S \in \mathcal{U}_k$ (or $\mathcal{U}_{\leq k}$).
    $\triangleright$ $\mu(S)$ may be defined using the solution $x$
Output $\mathcal{S}$ (If $|\mathcal{S}| < k$, add $k - |\mathcal{S}|$ arbitrary vectors to $\mathcal{S}$ first).
```

Lemma 3.2.1. *Let $T \subseteq [n]$ be of size no more than d. Then*

$$\det(V_T(x)^\top V_T(x)) = x^T \det(V_T^\top V_T)$$

Proof. The statement is true by multilinearity of the determinant and the exact formula for $V_T(x)^\top V_T(x)$ as follows. The matrix $V_T(x)^\top V_T(x)$ has (i, j) entry

$$\left(V_T(x)^\top V_T(x)\right)_{i,j} = \sqrt{x_i} v_i \cdot \sqrt{x_j} v_j = \sqrt{x_i x_j} v_i \cdot v_j$$

for each pair $i, j \in [|T|]$. By the multilinearity of the determinant, we can take the factor $\sqrt{x_i}$ out from each row i of $V_T(x)^\top V_T(x)$ and the factor $\sqrt{x_j}$ out from each column j of $V_T(x)^\top V_T(x)$. This gives

$$\det(V_T(x)^\top V_T(x)) = \prod_{i \in [|T|]} \sqrt{x_i} \prod_{j \in [|T|]} \sqrt{x_j} \det(V_T^\top V_T) = x^T \det(V_T^\top V_T)$$

$\square$

We also need the following identity, which is well-known and extends the Cauchy-Binet formula for the determinant to the functions E_k.

$$E_k(VV^\top) = E_k(V^\top V) = \sum_{S \in \mathcal{U}_k} \det(V_S^\top V_S). \tag{3.1}$$

The identity (3.1) appeared in [MS17] and, specifically for $k = d - 1$, as Lemma 3.8 in [AB13]. Now we are ready to prove Theorem 3.1.3.

Proof of Theorem 3.1.3: Let μ' denote the sampling distribution over $\mathcal{U}$, where $\mathcal{U} = \mathcal{U}_k$ or $\mathcal{U}_{\leq k}$, with probability of sampling $S \in \mathcal{U}$ proportional to $\mu(S)\det(V_S V_S^\top)$. Because $\operatorname{tr}\left(\sum_{i\in[n]} x_i v_i v_i^\top\right)^{-1} = \mathsf{CP} \leq$ OPT, it is enough to show that

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\operatorname{tr}\left(\sum_{i\in\mathcal{S}} v_i v_i^\top\right)^{-1}\right] \leq \alpha \operatorname{tr}\left(\sum_{i\in[n]} x_i v_i v_i^\top\right)^{-1}. \tag{3.2}$$

Note that in case $|\mathcal{S}| < k$, algorithm $\mathcal{A}$ adds $k - |\mathcal{S}|$ arbitrary vector to $\mathcal{S}$, which can only decrease the objective value of the solution.

First, a simple but important observation ([AB13]): for any $d \times d$ matrix M of rank d, we have

$$\operatorname{tr} M^{-1} = \sum_{i=1}^{d} \frac{1}{\lambda_i(M)} = \frac{e_{d-1}(\lambda(M))}{e_d(\lambda(M))} = \frac{E_{d-1}(M)}{\det M}. \tag{3.3}$$

Therefore, we have

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\operatorname{tr}\left(\sum_{i\in\mathcal{S}} v_i v_i^\top\right)^{-1}\right] &= \sum_{S\in\mathcal{U}} \Pr_{\mu'}[\mathcal{S} = S] \operatorname{tr}\left(V_S V_S^\top\right)^{-1} \\ &= \sum_{S\in\mathcal{U}} \frac{\mu(S)\det\left(V_S V_S^\top\right)}{\sum_{S'\in\mathcal{U}}\mu(S')\det(V_{S'}V_{S'}^\top)} \frac{E_{d-1}(V_S V_S^\top)}{\det\left(V_S V_S^\top\right)} \\ &= \frac{\sum_{S\in\mathcal{U}}\mu(S)E_{d-1}(V_S V_S^\top)}{\sum_{S\in\mathcal{U}}\mu(S)\det(V_S V_S^\top)}.\end{aligned}$$

We can now apply the Cauchy-Binet formula (3.1) for E_{d-1}, $E_d = \det$, and the matrix $V_S V_S^\top$ to the numerator and denominator on the right hand side, and we get

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\operatorname{tr}\left(\sum_{i\in\mathcal{S}} v_i v_i^\top\right)^{-1}\right] = \frac{\sum_{S\in\mathcal{U}}\sum_{|T|=d-1,T\subseteq S}\mu(S)\det(V_T^\top V_T)}{\sum_{S\in\mathcal{U}}\mu(S)\sum_{|R|=d,R\subseteq S}\det(V_R^\top V_R)}$$
$$= \frac{\sum_{|T|=d-1,T\subseteq[n]}\det\left(V_T^\top V_T\right)\sum_{S\in\mathcal{U},S\supseteq T}\mu(S)}{\sum_{|R|=d,R\subseteq[n]}\det\left(V_R^\top V_R\right)\sum_{S\in\mathcal{U},S\supseteq R}\mu(S)}$$
$$= \frac{\sum_{|T|=d-1,T\subseteq[n]}\det\left(V_T^\top V_T\right)\Pr_{\mu}\left[\mathcal{S}\supseteq T\right]}{\sum_{|R|=d,R\subseteq[n]}\det\left(V_R^\top V_R\right)\Pr_{\mu}\left[\mathcal{S}\supseteq R\right]}$$

where we change the order of summation at the second to last equality. Next, we apply (3.3) and the Cauchy-Binet formula (3.1) in a similar way to the matrix $V(x)V(x)^\top$:

$$\operatorname{tr}\left(V(x)V(x)^\top\right)^{-1} = \frac{E_{d-1}(V(x)V(x)^\top)}{\det(V(x)V(x)^\top)} = \frac{\sum_{|T|=d-1,T\subseteq[n]}\det(V_T(x)^\top V_T(x))}{\sum_{|R|=d,R\subseteq[n]}\det(V_R(x)^\top V_R(x))}$$
$$= \frac{\sum_{|T|=d-1,T\subseteq[n]}\det\left(V_T^\top V_T\right)x^T}{\sum_{|R|=d,R\subseteq[n]}\det\left(V_R^\top V_R\right)x^R}$$

where we use the fact that $\det(V_R(x)^\top V_R(x)) = x^R\det(V_R^\top V_R)$ and $\det(V_T(x)^\top V_T(x)) = x^T\det(V_T^\top V_T)$ in the last equality by Lemma 3.2.1.

Hence, the inequality (3.2) which we want to show is equivalent to

$$\frac{\sum_{|T|=d-1,T\subseteq[n]}\det\left(V_T^\top V_T\right)\Pr_{\mu}\left[\mathcal{S}\supseteq T\right]}{\sum_{|R|=d,R\subseteq[n]}\det\left(V_R^\top V_R\right)\Pr_{\mu}\left[\mathcal{S}\supseteq R\right]} \leq \alpha\frac{\sum_{|T|=d-1,T\subseteq[n]}\det\left(V_T^\top V_T\right)x^T}{\sum_{|R|=d,R\subseteq[n]}\det\left(V_R^\top V_R\right)x^R} \tag{3.4}$$

which is equivalent to

$$\sum_{|T|=d-1,|R|=d}\det\left(V_T^\top V_T\right)\det\left(V_R^\top V_R\right)\cdot x^R\cdot\Pr_{\mu}\left[\mathcal{S}\supseteq T\right]$$
$$\leq \alpha\sum_{|T|=d-1,|R|=d}\det\left(V_T^\top V_T\right)\det\left(V_R^\top V_R\right)\cdot x^T\cdot\Pr_{\mu}\left[\mathcal{S}\supseteq R\right]. \tag{3.5}$$

By the assumption that $\frac{\Pr_\mu[\mathcal{S}\supseteq T]}{\Pr_\mu[\mathcal{S}\supseteq R]} \leq \alpha \frac{x^T}{x^R}$ for each subset $T, R \subseteq [n]$ with $|T| = d-1$ and $|R| = d$,

$$\det\left(V_T^\top V_T\right) \det\left(V_R^\top V_R\right) \cdot x^R \cdot \Pr_\mu\left[\mathcal{S} \supseteq T\right] \leq \alpha \det\left(V_T^\top V_T\right) \det\left(V_R^\top V_R\right) \cdot x^T \cdot \Pr_\mu\left[\mathcal{S} \supseteq R\right] \quad (3.6)$$

Summing (3.6) over all T, R proves (3.5). □

3.3 Approximating Optimal Design without Repetitions

In this section, we prove Theorem 3.1.5 by constructing α-approximate $(d-1, d)$-independent distributions for appropriate values of α. We first consider the case when $k = d$ and then the asymptotic case when $k = \Omega\left(\frac{d}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{1}{\epsilon}\right)$. We also remark that the argument for $k = d$ can be generalized for all $k \leq d$, and we discuss this generalization in Section 3.5.1.

3.3.1 d-approximation for $k = d$

We prove the following lemma which, together with Theorem 3.1.3, implies the d-approximation for A-optimal design when $k = d$.

Lemma 3.3.1. *Let $k = d$. The hard-core distribution μ on $\mathcal{U}_k$ with parameter x is d-approximate $(d-1, d)$-independent.*

Proof. Observe that for any $S \in \mathcal{U}_k$, we have $\mu(S) = \frac{x^S}{Z}$ where $Z = \sum_{S' \in \mathcal{U}_k} x^{S'}$ is the normalization factor. For any $T \subseteq [n]$ such that $|T| = d-1$, we have

$$\Pr_{\mathcal{S}\sim\mu}\left[\mathcal{S} \supseteq T\right] = \sum_{S \in \mathcal{U}_k : S \supseteq T} \frac{x^S}{Z} = \frac{x^T}{Z} \cdot \left(\sum_{i \in [n]\setminus T} x_i\right) \leq d\frac{x^T}{Z}.$$

where we use $k = d$ and $\sum_{i\in[n]\setminus T} x_i \leq k = d$. For any $R \subseteq [n]$ such that $|R| = d$, we have

$$\Pr_{\mathcal{S}\sim\mu}[\mathcal{S} \supseteq R] = \sum_{S\in\mathcal{U}_k : S\supseteq R} \frac{x^S}{Z} = \frac{x^R}{Z}.$$

Thus for any $T, R \subseteq [n]$ such that $|T| = d - 1$ and $|R| = d$, we have

$$\frac{\Pr_{\mathcal{S}\sim\mu}[\mathcal{S} \supseteq T]}{\Pr_{\mathcal{S}\sim\mu}[\mathcal{S} \supseteq R]} \leq d\frac{x^T}{x^R}.$$

□

3.3.2 $(1 + \epsilon)$-approximation

Now, we show that there is a hard-core distribution μ on $\mathcal{U}_{\leq k}$ that is $(1+\epsilon)$-approximate $(d-1, d)$-independent when $k = \Omega\left(\frac{d}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{1}{\epsilon}\right)$.

Lemma 3.3.2. *Fix some $0 < \epsilon \leq 2$, and let $k = \Omega\left(\frac{d}{\epsilon} + \frac{\log(1/\epsilon)}{\epsilon^2}\right)$. The hard-core distribution μ on $\mathcal{U}_{\leq k}$ with parameter λ, defined by*

$$\lambda_i = \frac{x_i}{1 + \frac{\epsilon}{4} - x_i},$$

is $(1 + \epsilon)$-approximate $(d - 1, d)$-wise independent.

Proof. For simplicity of notation, let us denote $\beta = 1+\frac{\epsilon}{4}$, and $\xi_i = \frac{x_i}{\beta}$. Observe that the probability mass under μ of any set S of size at most k is proportional to $\left(\prod_{i\in S}\xi_i\right)\left(\prod_{i\notin S}(1-\xi_i)\right)$. Thus, μ is equivalent to the following distribution: sample a set $\mathcal{B} \subseteq [n]$ by including every $i \in [n]$ in $\mathcal{B}$ independently with probability ξ_i; then we have $\mu(S) = \Pr[\mathcal{B} = S \mid |\mathcal{B}| \leq k]$ for every S of size at most k. Let us fix for the rest of the proof arbitrary sets $T, R \subseteq [n]$ of size $d - 1$ and d, respectively. By the observation above, for $\mathcal{S}$ sampled according to μ, and $\mathcal{B}$ as above, we have

$$\frac{\Pr[\mathcal{S} \supseteq T]}{\Pr[\mathcal{S} \supseteq R]} = \frac{\Pr[\mathcal{B} \supseteq T \text{ and } |\mathcal{B}| \leq k]}{\Pr[\mathcal{B} \supseteq R \text{ and } |\mathcal{B}| \leq k]} \leq \frac{\Pr[\mathcal{B} \supseteq T]}{\Pr[\mathcal{B} \supseteq R \text{ and } |\mathcal{B}| \leq k]}$$

We have $\Pr[\mathcal{B} \supseteq T] = \xi^T = \frac{x^T}{\beta^{d-1}}$. To simplify the probability in the denominator, let us introduce, for each $i \in [n]$, the indicator random variable Y_i, defined to be 1 if $i \in \mathcal{B}$ and 0 otherwise. By the choice of $\mathcal{B}$, the Y_i's are independent Bernoulli random variables with mean ξ_i, respectively. We can write

$$\begin{aligned} \Pr[\mathcal{B} \supseteq R \text{ and } |\mathcal{B}| \leq k] &= \Pr\left[\forall i \in R : Y_i = 1 \text{ and } \sum_{i \notin R} Y_i \leq k - d\right] \\ &= \Pr[\forall i \in R : Y_i = 1] \Pr\left[\sum_{i \notin R} Y_i \leq k - d\right], \end{aligned}$$

where the last equality follows by the independence of the Y_i. The first probability on the right hand side is just $\xi^R = \frac{x^R}{\beta^d}$, and plugging into the inequality above, we get

$$\frac{\Pr[\mathcal{S} \supseteq T]}{\Pr[\mathcal{S} \supseteq R]} \leq \beta \frac{x^T}{x^R \Pr[\sum_{i \notin R} Y_i \leq k - d]}. \tag{3.7}$$

We claim that

$$\Pr[\sum_{i \notin R} Y_i \leq k - d] \geq 1 - \frac{\epsilon}{4}$$

as long as $k = \Omega\left(\frac{d}{\epsilon} + \frac{1}{\epsilon^2} \log \frac{1}{\epsilon}\right)$. The proof follows from standard concentration of measure arguments. Let $Y = \sum_{i \notin R} Y_i$, and observe that $\mathbb{E}[Y] = \frac{1}{\beta}(k - x(R))$, where $x(R)$ is shorthand for $\sum_{i \in R} x_i$. By Chernoff's bound,

$$\Pr[Y > k - d] < e^{-\frac{\delta^2}{3\beta}(k - x(R))} \tag{3.8}$$

where

$$\delta = \frac{\beta(k-d)}{k - x(R)} - 1 = \frac{(\beta - 1)k + x(R) - \beta d}{k - x(R)}.$$

The exponent on the right hand side of (3.8) simplifies to

$$\frac{\delta^2(k-x(R))}{3\beta}=\frac{((\beta-1)k+x(R)-\beta d)^2}{3\beta(k-x(R))}\geq\frac{((\beta-1)k-\beta d)^2}{3\beta k}.$$

For the bound $\Pr[Y>k-d]\leq\frac{\epsilon}{4}$, it suffices to have

$$(\beta-1)k-\beta d\geq\sqrt{3\beta\log(4/\epsilon)k}.$$

Assuming that $k\geq\frac{C\log(4/\epsilon)}{\epsilon^2}$ for a sufficiently big constant C, the right hand side is at most $\frac{\epsilon k}{8}$. So, as long as $k\geq\frac{\beta d}{\beta-1-\frac{\epsilon}{8}}$, the inequality is satisfied and $\Pr[Y>k-d]<\frac{\epsilon}{4}$, as we claimed.

The proof of the lemma now follows since for any $|T|=d-1$ and $|R|=d$, we have

$$\frac{\Pr[\mathcal{S}\supseteq T]}{\Pr[\mathcal{S}\supseteq R]}\leq\beta\frac{x^T}{x^R\Pr[\sum_{i\notin R}Y_i\leq k-d]}\leq\frac{1+\frac{\epsilon}{4}}{1-\frac{\epsilon}{4}}\frac{x^T}{x^R},\tag{3.9}$$

and $\frac{1+\frac{\epsilon}{4}}{1-\frac{\epsilon}{4}}\leq 1+\epsilon$. □

The $(1+\epsilon)$-approximation for large enough k in Theorem 3.1.5 now follows directly from Lemma 3.3.2 and Theorem 3.1.3.

3.4 Approximately Optimal Design with Repetitions

In this section, we consider the A-optimal design without the bound $x_i\leq 1$ and prove Theorem 3.1.6. That is, we allow the sample set S to be a multi-set. We obtain a tight bound on the integrality gap in this case. Interestingly, we reduce the problem to a special case of A-optimal design without repetitions that allows us to obtained an improved approximation.

We first describe a sampling Algorithm 3.2 that achieves a $\frac{k(1+\epsilon)}{k-d+1}$-approximation for any $\epsilon>0$. In the algorithm, we introduce $\mathrm{poly}(n,1/\epsilon)$ number of copies of each vector to ensure that the fractional solution assigns equal fractional value for each copy of each vector. Then we use the proportional volume sampling where the measure distribution μ is defined on sets of the new larger ground set U over copies of the original input vectors. The distribution μ is just the uniform distribution over subsets of size k of U, and we are effectively using traditional volume sampling over U. Notice, however, that the distribution over multisets of the original set of vectors is different. The proportional volume sampling used in the algorithm can be implemented in the same way as the one used for without repetition setting, as described in Section 3.6.1, which runs in $\mathrm{poly}(n, d, k, 1/\epsilon)$ time.

In Section 3.6.3, we describe a new implementation of proportional volume sampling procedure which improves the running time to $\mathrm{poly}(n, d, k, \log(1/\epsilon))$. The new algorithm is still efficient even when U has exponential size by exploiting the facts that μ is uniform and that U has only at most n distinct vectors.

Algorithm 3.2 Approximation Algorithm for A-optimal design with repetitions

```
Given $x \in \mathbb{R}^n_+$ with $\sum_{i=1}^n x_i = k$, $\epsilon > 0$, and vectors $v_1, \ldots, v_n$.
Let $q = \frac{2n}{\epsilon k}$. Set $x'_i := \frac{k-n/q}{k} x_i$ for each $i$, and round each $x'_i$ up to a multiple of $1/q$.
If $\sum_{i=1}^n x'_i < k$, add $1/q$ to any $x'_i$ until $\sum_{i=1}^n x'_i = k$.
Create $qx'_i$ copies of vector $v_i$ for each $i \in [n]$. Denote $W$ the set of size $\sum_{i=1}^n qx'_i = qk$ of all
    those copies of vectors. Denote $U$ the new index set of $W$of size $qk$.    ▷ This implies that we
    can assume that our new fractional solution $y_i = 1/q$ is equal over all $i \in U$
Sample a subset $\mathcal{S}$ of $U$ of size $k$ where $\Pr[\mathcal{S} = S] \propto \det(W_S W_S^\top)$ for each $S \subseteq U$ of size $k$.
Set $X_i = \sum_{w \in W_{\mathcal{S}}} \mathbb{1}(w \text{ is a copy of } v_i)$ for all $i \in [n]$ ▷ Get an integral solution $X$ by counting
    numbers of copies of $v_i$ in $\mathcal{S}$.
Output $X$.
```

Lemma 3.4.1. *Algorithm 3.2, when given as input $x \in \mathbb{R}^n_+$ s.t. $\sum_{i=1}^n x_i = k$, $1 \geq \epsilon > 0$, and $v_1, \ldots, v_n$, outputs a random $X \in \mathbb{Z}^n_+$ with $\sum_{i=1}^n X_i = k$ such that*

$$\mathbb{E}\left[\mathrm{tr}\left(\sum_{i=1}^n X_i v_i v_i^\top\right)^{-1}\right] \leq \frac{k(1+\epsilon)}{k-d+1} \mathrm{tr}\left(\sum_{i=1}^n x_i v_i v_i^\top\right)^{-1}$$

Proof. Define $x'_i, y, W, U, \mathcal{S}, X$ as in the algorithm. We will show that

$$\mathbb{E}\left[\operatorname{tr}\left(\sum_{i=1}^{n} X_i v_i v_i^\top\right)^{-1}\right] \leq \frac{k}{k-d+1}\operatorname{tr}\left(\sum_{i=1}^{n} x'_i v_i v_i^\top\right)^{-1} \leq \frac{k(1+\epsilon)}{k-d+1}\operatorname{tr}\left(\sum_{i=1}^{n} x_i v_i v_i^\top\right)^{-1}$$

The second inequality is by observing that the scaling $x'_i := \frac{k-n/q}{k} x_i$ multiplies the objective $\operatorname{tr}\left(\sum_{i=1}^{n} x_i v_i v_i^\top\right)^{-1}$ by a factor of $\left(\frac{k-n/q}{k}\right)^{-1} = (1-\epsilon/2)^{-1} \leq 1+\epsilon$, and that rounding x_i up and adding $1/q$ to any x_i can only decrease the objective.

To show the first inequality, we first translate the two key quantities $\operatorname{tr}\left(\sum_{i=1}^{n} x'_i v_i v_i^\top\right)^{-1}$ and $\operatorname{tr}\left(\sum_{i=1}^{n} X_i v_i v_i^\top\right)^{-1}$ from the with-repetition setting over V and $[n]$ to the without-repetition setting over W and U. First, $\operatorname{tr}\left(\sum_{i=1}^{n} x'_i v_i v_i^\top\right)^{-1} = \operatorname{tr}\left(\sum_{i\in U} y_i w_i w_i{}^\top\right)^{-1}$, where $y_i = \frac{1}{q}$ are all equal over all $i \in U$, and w_i is the copied vector in W at index $i \in U$. Second, $\operatorname{tr}\left(\sum_{i=1}^{n} X_i v_i v_i^\top\right)^{-1} = \operatorname{tr}\left(\sum_{i\in\mathcal{S}\subseteq U} w_i w_i{}^\top\right)^{-1}$.

Let μ' be the distribution over subsets S of U of size k defined by $\mu'(S) \propto \det(W_S W_S^\top)$. It is, therefore, sufficient to show that the sampling distribution μ' satisfies

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\operatorname{tr}\left(\sum_{i\in\mathcal{S}\subseteq U} w_i w_i{}^\top\right)^{-1}\right] \leq \frac{k}{k-d+1}\operatorname{tr}\left(\sum_{i\in U} y_i w_i w_i{}^\top\right)^{-1} \tag{3.10}$$

Observe that μ' is the same as sampling a set $S \subseteq U$ of size k with probability proportional to $\mu(S)\det(W_S W_S^\top)$ where μ is uniform. Hence, by Theorem 3.1.3, it is enough to show that for all $T, R \subseteq U$ with $|T| = d-1, |R| = d$,

$$\frac{\Pr_{\mu}[\mathcal{S} \supseteq T]}{\Pr_{\mu}[\mathcal{S} \supseteq R]} \leq \left(\frac{k}{k-d+1}\right)\frac{y^T}{y^R} \tag{3.11}$$

With μ being uniform and y_i being all equal to $1/q$, the calculation is straightforward:

$$\frac{\Pr_{\mu}[\mathcal{S} \supseteq T]}{\Pr_{\mu}[\mathcal{S} \supseteq R]} = \frac{\binom{qk-d+1}{k-d+1}/\binom{qk}{k}}{\binom{qk-d}{k-d}/\binom{qk}{k}} = \frac{qk-d+1}{k-d+1} \text{ and } \frac{y^T}{y^R} = \frac{1}{y_i} = q \tag{3.12}$$

Therefore, (3.11) holds because

$$\frac{\Pr_{\mu}[\mathcal{S} \supseteq T]}{\Pr_{\mu}[\mathcal{S} \supseteq R]} \cdot \left(\frac{y^T}{y^R}\right)^{-1} = \frac{qk-d+1}{k-d+1} \cdot \frac{1}{q} \leq \frac{qk}{k-d+1} \cdot \frac{1}{q} = \frac{k}{k-d+1},$$

□

Remark 3.4.2. The approximation ratio for A-optimality with repetitions for $k \geq d$ is tight, since it matches the integrality gap lower bound stated in Theorem 3.7.3.

3.5 Generalizations

In this section we show that our arguments extend to the regime $k \leq d$ and give a k-approximation (without repetitions), which matches the integrality gap of our convex relaxation. We also derive a restricted invertibility principle for the harmonic mean of eigenvalues.

3.5.1 k-Approximation Algorithm for $k \leq d$

Recall that our aim is to select a set $S \subseteq [n]$ of size $k \leq d$ that minimizes $\sum_{i=1}^{k} \lambda_i^{-1}$, where $\lambda_1, \ldots, \lambda_k$ are the k largest eigenvalues of the matrix $V_S V_S^\top$. We need to reformulate our convex relaxation since when $k < d$, the inverse of $M(S) = \sum_{i \in S} v_i v_i^\top$ for $|S| = k$ is no longer well-defined. We write a new convex program:

$$\min \frac{E_{k-1}\left(\sum_{i=1}^n x_i v_i v_i^\top\right)}{E_k\left(\sum_{i=1}^n x_i v_i v_i^\top\right)} \tag{3.13}$$

s.t.

$$\sum_{i=1}^n x_i = k \tag{3.14}$$

$$0 \leq x_i \leq 1 \quad \forall i \in [n] \tag{3.15}$$

Once again we denote the optimal value of (3.13)–(3.15) by CP. While the proof that this relaxes the original problem is easy, the convexity is non-trivial. Fortunately, ratios of symmetric polynomials are known to be convex.

Lemma 3.5.1. *The optimization problem* (3.13)–(3.15) *is a convex relaxation of the A-optimal design problem when $k \leq d$.*

Proof. To prove convexity, we first note that the function $f(M) = \frac{E_k(M)}{E_{k-1}(M)}$ is concave on positive semidefinite matrices M of rank at least k. This was proved by [BM61, Theorem 4] for positive definite M, and can be extended to M of rank at least k by a limiting argument. Alternatively, we can use the theorem of [ML57] that the function $g(\lambda) = \frac{e_k(\lambda)}{e_{k-1}(\lambda)}$ is concave on vectors $\lambda \in \mathbb{R}^d$ with non-negative entries and at least k positive entries. Because g is symmetric under permutations of its arguments and concave, and $f(M) = g(\lambda(M))$, where $\lambda(M)$ is the vector of eigenvalues of M, by a classical result of [Dav57] the function f is concave on positive semidefinite matrices of rank at least k.

Notice that the objective (3.13) equals $\frac{1}{f(M(x))}$ for the linear matrix-valued function $M(x) = \sum_{i=1}^n x_i v_i v_i^\top$. Therefore, to prove that (3.13) is convex in x for non-negative x, it suffices to show that $\frac{1}{f(M)}$ is convex in M for positive semidefinite M. Since the function $\frac{1}{z}$ is convex and monotone decreasing over positive reals z, and f is concave and non-negative over positive semidefinite matrices of rank at least k, we have that $\frac{1}{f(M)}$ is convex in M, as desired. Then (3.13)–(3.15) is an optimization problem with a convex objective and affine constraints, so we have a convex optimization problem.

Let OPT be the optimal value of the A-optimal design problem, and let S be an optimal solution. We need to show that $\mathsf{CP} \leq$ OPT. To this end, let x be the indicator vector of S, i.e. $x_i = 1$ if and only if $i \in S$, and $x_i = 0$ otherwise. Then,

$$\mathsf{CP} \leq \frac{E_{k-1}(M(S))}{E_k(M(S))} = \frac{\sum_{i=1}^{k} \prod_{j \neq i} \lambda_j(M(S))}{\prod_i \lambda_i(M(S))} = \sum_{i=1}^{k} \frac{1}{\lambda_i(M(S))} = \text{OPT}.$$

Above, $\lambda_1(M(S)), \ldots, \lambda_k(M(S))$ are, again, the nonzero eigenvalues of $M(S) = \sum_{i \in S} v_i v_i^\top$. □

We shall use the natural analog of proportional volume sampling: given a measure μ on subsets of size k, we sample a set S with probability proportional to $\mu(S)E_k(M(S))$. In fact, we will only take $\mu(S)$ proportional to x^S, so this reduces to sampling S with probability proportional to $E_k(\sum_{i \in S} x_i v_i v_i^\top)$, which is the standard volume sampling with vectors scaled by $\sqrt{x_i}$, and can be implemented efficiently using, e.g. the algorithm of [DR10].

The following version of Theorem 3.1.3 still holds with this modified proportional volume sampling. The proof is exactly the same, except for mechanically replacing every instance of determinant by E_k, of E_{d-1} by E_{k-1}, and in general of d by k.

Theorem 3.5.2. *Given integers $k \leq d \leq n$ and a vector $x \in [0,1]^n$ such that $1^\top x = k$, suppose there exists a measure μ on $\mathcal{U}_k$ that is α-approximate $(k-1,k)$-wise independent. Then for x the optimal solution of* (3.13)–(3.15)*, proportional volume sampling with measure μ gives a polynomial time α-approximation algorithm for the A-optimal design problem.*

We can now give the main approximation guarantee we have for $k \leq d$.

Theorem 3.5.3. *For any $k \leq d$, proportional volume sampling with the hard-core measure μ on $\mathcal{U}_k$ with parameter x equal to the optimal solution of* (3.13)–(3.15) *gives a k-approximation to the A-optimal design problem.*

Proof. In view of Theorem 3.5.2, we only need to show that μ is k-approximate $(k-1,k)$-wise independent. This is a straightforward calculation: for $\mathcal{S} \sim \mu$, and any $T \subseteq [n]$ of size $k-1$ and $R \subseteq [n]$ of size k,

$$\frac{\Pr[\mathcal{S} \supseteq T]}{\Pr[\mathcal{S} \supseteq R]} = \frac{x^T \sum_{i \notin T} x_i}{x^R} \leq k\frac{x^T}{x^R}.$$

This completes the proof. □

The algorithm can be derandomized using the method of conditional expectations analogously to the case of $k = d$ that we will show in Theorem 3.6.5.

The k-approximation also matches the integrality gap of (3.13)–(3.15). Indeed, we can take a k-dimensional integrality gap instance $v_1, \ldots, v_n$, and embed it in $\mathbb{R}^d$ for any $d > k$ by padding each vector with 0's. On such an instance, the convex program (3.13)–(3.15) is equivalent to the convex program (2.1)–(2.3). Thus the integrality gap that we will show in Theorem 3.7.3 implies an integrality gap of k for all $d \geq k$.

3.5.2 Restricted Invertibility Principle for Harmonic Mean

Next we state and prove our restricted invertibility principle for harmonic mean in a general form.

Theorem 3.5.4. *Let $v_1, \ldots, v_n \in \mathbb{R}^d$, and $c_1, \ldots, c_n \in \mathbb{R}_+$, and define $M = \sum_{i=1}^n c_i v_i v_i^\top$. For any $k \leq r = \frac{\operatorname{tr}(M)}{\|M\|}$, there exists a subset $S \subseteq [n]$ of size k such that the k largest eigenvalues $\lambda_1, \ldots, \lambda_k$ of the matrix $\sum_{i \in S} v_i v_i^\top$ satisfy*

$$\left(\frac{1}{k}\sum_{i=1}^{k} \frac{1}{\lambda_i}\right)^{-1} \geq \frac{r-k+1}{r} \cdot \frac{\operatorname{tr}(M)}{\sum_{i=1}^n c_i}.$$

Proof. Without loss of generality we can assume that $\sum_{i=1}^n c_i = k$. Then, by Theorem 3.5.3, proportional volume sampling with the hard-core measure μ on $\mathcal{U}_k$ with parameter $c = (c_1, \ldots, c_n)$

gives a random set $\mathcal{S}$ of size k such that

$$\mathbb{E}\left[\sum_{i=1}^{k} \frac{1}{\lambda_i(M(\mathcal{S}))}\right] \leq k\frac{E_{k-1}(M)}{E_k(M)},$$

where $\lambda_i(M(\mathcal{S}))$ is the i-th largest eigenvalues of $M(\mathcal{S}) = \sum_{i\in S} v_i v_i^\top$. Therefore, there exists a set S of size k such that

$$\left(\frac{1}{k}\sum_{i=1}^{k} \frac{1}{\lambda_i(M(S))}\right)^{-1} \geq \frac{E_k(M)}{E_{k-1}(M)} = \frac{e_k(\lambda(M))}{e_{k-1}(\lambda(M))},$$

where $\lambda(M)$ is the vector of eigenvalues of M. In the rest of the proof we compare the right hand side above with $\mathrm{tr}(M)$.

Recall that a vector $x \in \mathbb{R}^d_+$ is majorized by a vector $y \in \mathbb{R}^d_+$, written $x \prec y$, if $\sum_{j=1}^{i} x_{(j)} \leq \sum_{j=1}^{i} y_{(j)}$ holds for all $i \in [n]$, and $\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} y_i$. Here $x_{(j)}$ denotes the j-th largest coordinate of x, and similarly for $y_{(j)}$. Recall further that a function $f : \mathbb{R}^d_+ \to \mathbb{R}$ is Schur-concave if $x \prec y$ implies $f(x) \geq f(y)$. The function $\frac{e_k(x)}{e_{k-1}(x)}$ was shown to be Schur concave by [GS12]; alternatively, it is symmetric under permutations of x and concave, as shown in [ML57] (and mentioned above), which immediately implies that it is Schur-concave. We define a vector x which majorizes $\lambda(M)$ by setting $x_i = \frac{1}{r}\sum_{i=1}^{d} \lambda_i(M)$ for $i \in [r]$, and $x_i = 0$ for $i > r$ (we assume here that $\lambda_1(M) \geq \ldots \geq \lambda_d(M)$). By Schur concavity,

$$\frac{e_k(\lambda(M))}{e_{k-1}(\lambda(M))} \leq \frac{e_k(x)}{e_{k-1}(x)} = \frac{r-k+1}{rk}\sum_{i=1}^{d} \lambda_i(M).$$

Since $\sum_{i=1}^{d} \lambda_i(M) = \mathrm{tr}(M)$, and we assumed that $\sum_{i=1}^{n} c_i = k$, this completes the proof of the theorem. □

3.5.3 The Generalized Ratio Objective

In A-optimal design, given $V = [v_1 \ldots\ v_n] \in \mathbb{R}^{d\times n}$, we state the objective as minimizing

$$\operatorname{tr}\left(\sum_{i\in S} v_i v_i^\top\right)^{-1} = \frac{E_{d-1}(V_S V_S^\top)}{E_d(V_S V_S^\top)}.$$

over subset $S \subseteq [n]$ of size k. In this section, for any given $0 \leq l' < l \leq d$, we consider the following *generalized ratio problem*:

$$\min_{S\subseteq[n],|S|=k} \left(\frac{E_{l'}(V_S V_S^\top)}{E_l(V_S V_S^\top)}\right)^{\frac{1}{l-l'}} \tag{3.16}$$

The above problem naturally interpolates between A-optimality and D-optimality. This follows since for $l = d$ and $l' = 0$, the objective reduces to

$$\min_{S\subseteq[n],|S|=k} \left(\frac{1}{\det(V_S V_S^\top)}\right)^{\frac{1}{d}}. \tag{3.17}$$

A closely related generalization between A- and D-criteria was considered in [MS17]. Indeed, their generalization corresponds to the case when $l = d$ and l' takes any value from 0 and $d-1$.

In this section, we show that our results extend to solving generalized ratio problem. We begin by describing a convex program for the generalized ratio problem. We then generalize the proportional volume sampling algorithm to *proportional l-volume sampling*. Following the same plan as in the proof of A-optimality, we then reduce the approximation guarantee to near-independence properties of certain distribution. Here again, we appeal to the same product measure and obtain identical bounds, summarized in Table 3.3, on the performance of the algorithm.

Convex Relaxation

As in solving A-optimality, we may define relaxations for with and without repetitions as follows.

Table 3.3: Summary of approximation ratio obtained by our work on generalized ratio problem.

Problem	A-optimal $(l' = d-1, l = d)$	$\min_{\lvert S\rvert=k} \left(\frac{E_{l'}(V_S V_S^\top)}{E_l(V_S V_S^\top)}\right)^{\frac{1}{l-l'}}$	D-optimal $(l' = 0, l = d)$
Case $k = d$	d	$l \cdot [(l-l')!]^{-\frac{1}{l-l'}} \leq \frac{el}{l-l'}$	e
Asymptotic $k >> d$ without Repetitions	$1+\epsilon$, for $k \geq \Omega\left(\frac{d}{\epsilon} + \frac{\log 1/\epsilon}{\epsilon^2}\right)$	$1+\epsilon$, for $k \geq \Omega\left(\frac{l}{\epsilon} + \frac{\log 1/\epsilon}{\epsilon^2}\right)$	$1+\epsilon$, for $k \geq \Omega\left(\frac{d}{\epsilon} + \frac{\log 1/\epsilon}{\epsilon^2}\right)$
Arbitrary k and d With Repetitions	$\frac{k}{k-d+1}$	$\frac{k}{k-l+1}$	$\frac{k}{k-d+1}$
Asymptotic $k >> d$ With Repetitions	$1+\epsilon$, for $k \geq d + \frac{d}{\epsilon}$	$1+\epsilon$, for $k \geq l + \frac{l}{\epsilon}$	$1+\epsilon$, for $k \geq d + \frac{d}{\epsilon}$

With Repetitions	**Without Repetitions**
$\min \left(\frac{E_{l'}\left(V(x)V(x)^\top\right)}{E_l\left(V(x)V(x)^\top\right)}\right)^{\frac{1}{l-l'}}$	$\min \left(\frac{E_{l'}\left(V(x)V(x)^\top\right)}{E_l\left(V(x)V(x)^\top\right)}\right)^{\frac{1}{l-l'}} \quad (3.18)$
s.t. $\sum_{i=1}^{n} x_i = k$	s.t. $\sum_{i=1}^{n} x_i = k \quad (3.19)$
$0 \leq x_i \quad \forall i \in [n]$	$0 \leq x_i \leq 1 \quad \forall i \in [n] \quad (3.20)$

We now show that $\left(\frac{E_{l'}\left(V(x)V(x)^\top\right)}{E_l\left(V(x)V(x)^\top\right)}\right)^{\frac{1}{l-l'}}$ is convex in x.

Lemma 3.5.5. *Let d be a positive integer. For any given pair $0 \leq l' < l \leq d$, the function*

$$f_{l',l}(M) = \left(\frac{E_{l'}(M)}{E_l(M)}\right)^{\frac{1}{l-l'}} \tag{3.21}$$

is convex over $d \times d$ positive semidefinite matrix M.

Proof. By Theorem 3 in [BM61], $(f_{l',l}(M))^{-1} = \left(\frac{E_l(M)}{E_{l'}(M)}\right)^{\frac{1}{l-l'}}$ is concave on positive semidefinite matrices M for each $0 \leq l' < l \leq d$. The function $\frac{1}{z}$ is convex and monotone decreasing over the positive reals z, and this, together with the concavity of $(f_{l',l}(M))^{-1}$ and that $(f_{l',l}(M))^{-1} > 0$, implies that $f_{l',l}(M)$ is convex in M. $\square$

Approximation via (l', l)-Wise Independent Distribution

Let $0 \leq l' < l \leq d$ and $\mathcal{U} \in \{\mathcal{U}_k, \mathcal{U}_{\leq k}\}$. We first show connection of approximation guarantees on objectives $\left(\frac{E_{l'}(V_S V_S^\top)}{E_l(V_S V_S^\top)}\right)^{\frac{1}{l-l'}}$ and $\frac{E_{l'}(V_S V_S^\top)}{E_l(V_S V_S^\top)}$. Suppose we already solve the convex relaxation of generalized ratio problem (3.18)-(3.20) and get a fractional solution x. Suppose that a randomized algorithm $\mathcal{A}$, upon receiving input $V \in \mathbb{R}^{d\times n}$ and $x \in \mathbb{R}^n$, outputs $S \in \mathcal{U}$ such that

$$\mathop{\mathbb{E}}_{S\sim\mathcal{A}}\left[\frac{E_{l'}(V_S V_S^\top)}{E_l(V_S V_S^\top)}\right] \leq \alpha' \frac{E_{l'}(V(x)V(x)^\top)}{E_l(V(x)V(x)^\top)} \tag{3.22}$$

for some constant $\alpha' > 0$. By the convexity of the function $f(z) = z^{l-l'}$ over positive reals z, we have

$$\mathbb{E}\left[\frac{E_{l'}(M)}{E_l(M)}\right] \geq \mathbb{E}\left[\left(\frac{E_{l'}(M)}{E_l(M)}\right)^{\frac{1}{l-l'}}\right]^{l-l'} \tag{3.23}$$

for any semi-positive definite matrix M. Combining (3.22) and (3.23) gives

$$\mathop{\mathbb{E}}_{S\sim\mathcal{A}}\left[\left(\frac{E_{l'}(V_S V_S^\top)}{E_l(V_S V_S^\top)}\right)^{\frac{1}{l-l'}}\right] \leq \alpha \left(\frac{E_{l'}(V(x)V(x)^\top)}{E_l(V(x)V(x)^\top)}\right)^{\frac{1}{l-l'}} \tag{3.24}$$

where $\alpha = (\alpha')^{\frac{1}{l-l'}}$. Therefore, it is sufficient for an algorithm to satisfy (3.22) and give a bound on α' in order to solve the generalized ratio problem up to factor α.

To show (3.22), we first define the proportional l-volume sampling and α-approximate (l', l)-wise independent distribution.

Definition 3.5.6. Let μ be probability measure on sets in $\mathcal{U}_k$ (or $\mathcal{U}_{\leq k}$). Then the proportional l-volume sampling with measure μ picks a set of vectors indexed by $S \in \mathcal{U}_k$ (or $\mathcal{U}_{\leq k}$) with probability proportional to $\mu(S)E_l(V_S V_S^\top)$.

Definition 3.5.7. Given integers d, k, n, a pair of integers $0 \leq l' \leq l \leq d$, and a vector $x \in [0,1]^n$ such that $1^\top x = k$, we call a measure μ on sets in $\mathcal{U}_k$ (or $\mathcal{U}_{\leq k}$), α-approximate (l', l)-wise independent with respect to x if for any subsets $T', T \subseteq [n]$ with $|T'| = l'$ and $|T| = l$, we have

$$\frac{\Pr_{\mathcal{S}\sim\mu}[T' \subseteq \mathcal{S}]}{\Pr_{\mathcal{S}\sim\mu}[T \subseteq \mathcal{S}]} \leq \alpha^{l-l'} \cdot \frac{x^{T'}}{x^T}$$

where $x^L := \prod_{i\in L} x_i$ for any $L \subseteq [n]$. We omit “with respect to x" when the context is clear.

The following theorem reduces the approximation guarantee in (3.22) to α-approximate (l', l)-wise independence properties of a certain distribution μ by utilizing proportional l-volume sampling.

Theorem 3.5.8. *Given integers d, k, n, $V = [v_1 \dots v_n] \in \mathbb{R}^{d\times n}$, and a vector $x \in [0,1]^n$ such that $1^\top x = k$, suppose there exists a distribution μ on sets in $\mathcal{U}_k$ (or $\mathcal{U}_{\leq k}$) and is α-approximate (l', l)-wise independent for some $0 \leq l' < l \leq d$. Then the proportional l-volume sampling with measure μ gives an α-approximation algorithm for minimizing $\left(\frac{E_{l'}(V_S V_S^\top)}{E_l(V_S V_S^\top)}\right)^{\frac{1}{l-l'}}$ over subsets $S \subseteq [n]$ of size k.*

Proof. Let μ' denote the sampling distribution over $\mathcal{U}$, where $\mathcal{U} = \mathcal{U}_k$ or $\mathcal{U}_{\leq k}$, with probability of sampling $S \in \mathcal{U}$ proportional to $\mu(S)E_l(V_S V_S^\top)$. We mechanically replace $T, R, d-1, d,$ and $\det$ in the proof of Theorem 3.1.3 with $T', T, l', l,$ and E_l to obtain

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\operatorname{tr}\left(\sum_{i\in\mathcal{S}} v_i v_i^\top\right)^{-1}\right] \leq \alpha^{l-l'} \operatorname{tr}\left(\sum_{i\in[n]} x_i v_i v_i^\top\right)^{-1}.$$

We finish the proof by observing that (3.22) implies (3.24), as discussed earlier. □

The following subsections generalize algorithms and proofs for with and without repetitions. The algorithm for generalized ratio problem can be summarized in Algorithm 3.4. Note that efficient implementation of the sampling is described in Section 3.6.5.

Algorithm 3.3 Generalized ratio approximation algorithm

```
Given an input $V = [v_1, \dots, v_n]$ where $v_i \in \mathbb{R}^d$, $k$ a positive integer, and a pair of integers $0 \le l' \le l \le d$.
Solve the convex relaxation $x = \text{argmin}_{x \in J^n : 1^\top x = k} \left( \frac{E_{l'}\left(V(x)V(x)^\top\right)}{E_l\left(V(x)V(x)^\top\right)} \right)^{\frac{1}{l-l'}}$ where $J = [0,1]$ if without repetitions or $\mathbb{R}^+$ if with repetitions.
if $k = l$ then
    Sample $\mu'(S) \propto x^S E_l\left(V_S V_S^\top\right)$ for each $S \in \mathcal{U}_k$
else if without repetition setting and $k \ge \Omega\left(\frac{d}{\epsilon} + \frac{\log(1/\epsilon)}{\epsilon^2}\right)$ then
    Sample $\mu'(S) \propto \lambda^S E_l\left(V_S V_S^\top\right)$ for each $S \in \mathcal{U}_{\le k}$ where $\lambda_i := \frac{x_i}{1+\epsilon/4-x_i}$
else if with repetition setting then
    Run Algorithm 3.2, except modifying the sampling step to sample a subset $\mathcal{S}$ of $U$ of size $k$ with $\Pr[\mathcal{S} = S] \propto E_l(W_S W_S^\top)$.
Output $\mathcal{S}$ (If $|\mathcal{S}| < k$, add $k - |\mathcal{S}|$ arbitrary vectors to $\mathcal{S}$ first).
```

Approximation Guarantee for Generalized Ratio Problem without Repetitions

We prove the following theorem which generalize Lemmas 3.3.1 and 3.3.2. The α-approximate (l', l)-wise independence property, together with Theorem 3.5.8, implies an approximation guarantee for generalized ratio problem without repetitions for $k = l$ and asymptotically for $k = \Omega\left(\frac{l}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{1}{\epsilon}\right)$.

Theorem 3.5.9. *Given integers d, k, n, a pair of integers $0 \le l' \le l \le d$, and a vector $x \in [0,1]^n$ such that $1^\top x = k$, the hard-core distribution μ on sets in $\mathcal{U}_k$ with parameter x is α-approximate (l', l)-wise independent when $k = l$ for*

$$\alpha = l \cdot [(l - l')!]^{-\frac{1}{l-l'}} \le \frac{el}{l - l'} \tag{3.25}$$

Moreover, for any $0 < \epsilon \le 2$ when $k = \Omega\left(\frac{l}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{1}{\epsilon}\right)$, the hard-core distribution μ on $\mathcal{U}_{\le k}$ with parameter λ, defined by

$$\lambda_i = \frac{x_i}{1 + \frac{\epsilon}{4} - x_i},$$

is $(1 + \epsilon)$-approximate (l', l)-wise independent.

Thus for minimizing the generalized ratio problem $\left(\frac{E_{l'}(V_S V_S^\top)}{E_l(V_S V_S^\top)}\right)^{\frac{1}{l-l'}}$ *over subsets* $S \in [n]$ *of size* k*, we obtain*

- $(\frac{el}{l-l'})$*-approximation algorithm when* $k = l$*, and*
- $(1+\epsilon)$*-approximation algorithm when* $k = \Omega\left(\frac{l}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{1}{\epsilon}\right)$.

Proof. We first prove the result for $k = l$. For all $T', T \subseteq [n]$ such that $|T'| = l', |T| = l$,

$$\frac{\underset{\mathcal{S}\sim\mu}{\mathsf{Pr}}[S \supseteq T']}{\underset{\mathcal{S}\sim\mu}{\mathsf{Pr}}[S \supseteq T]} = \frac{\sum_{|S|=k, S\supseteq T'} x^S}{\sum_{|S|=k, S\supseteq T} x^S} = \frac{x^{T'} \sum_{L \in \binom{[n]\setminus T'}{k-l'}} x^L}{x^T} \leq \frac{x^{T'} \sum_{L \in \binom{[n]}{k-l'}} x^L}{x^T}$$

We now use Maclaurin's inequality ([LT93]) to bound the quantity on the right-hand side

$$\sum_{L \in \binom{[n]}{k-l'}} x^L = e_{l-l'}(x) \leq \binom{n}{l-l'} (e_1(x)/n)^{l-l'} \leq \frac{n^{l-l'}}{(l-l')!} (l/n)^{l-l'} = \frac{l^{l-l'}}{(l-l')!} \tag{3.26}$$

Therefore,

$$\frac{\underset{\mathcal{S}\sim\mu}{\mathsf{Pr}}[S \supseteq T']}{\underset{\mathcal{S}\sim\mu}{\mathsf{Pr}}[S \supseteq T]} \leq \frac{l^{l-l'}}{(l-l')!} \frac{x^{T'}}{x^T} \tag{3.27}$$

which proves the (l', l)-wise independent property of μ with required approximation ratio from (3.25).

We now prove the result for $k = \Omega\left(\frac{l}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{1}{\epsilon}\right)$. The proof follows similarly from Lemma 3.3.2 by replacing T, R with T', T of sizes l', l instead of sizes $d-1, d$. In particular, the equation (3.7) becomes

$$\frac{\Pr[\mathcal{S} \supseteq T']}{\Pr[\mathcal{S} \supseteq T]} \leq \left(1 + \frac{\epsilon}{4}\right)^{l-l'} \frac{x^{T'}}{x^T \Pr[\sum_{i \notin T} Y_i \leq k - l]}. \tag{3.28}$$

and the Chernoff's bound (3.8) still holds by mechanically replacing d, R with l, T respectively. The resulting approximation ratio α satisfies

$$\alpha^{l-l'} = \frac{\left(1 + \frac{\epsilon}{4}\right)^{l-l'}}{1 - \frac{\epsilon}{4}} \leq (1+\epsilon)^{l-l'}.$$

where the inequality holds because $\epsilon \leq 2$. □

Approximation Guarantee for Generalized Ratio Problem with Repetitions

We now consider the generalized ratio problem *with repetitions*. The following statement is a generalization of Lemma 3.4.1.

Theorem 3.5.10. *Given $V = [v_1 \; v_2 \ldots v_n]$ where $v_i \in \mathbb{R}^d$, a pair of integers $0 \leq l' \leq l \leq d$, an integer $k \geq l$, and $1 \geq \epsilon > 0$, there is an α-approximation algorithm for minimizing $\left(\frac{E_{l'}(V_S V_S^\top)}{E_l(V_S V_S^\top)}\right)^{\frac{1}{l-l'}}$ over subsets $S \subseteq [n]$ of size k with repetitions for*

$$\alpha \leq \frac{k(1+\epsilon)}{k-l+1} \tag{3.29}$$

Proof. We use the algorithm similar to Algorithm 3.2 except that in step (5), we sample $\mathcal{S} \subseteq U$ of size k where $\Pr[\mathcal{S} = S] \propto E_l(W_S W_S^\top)$ in place of $\Pr[\mathcal{S} = S] \propto E_l(W_S W_S^\top)$. The analysis follows on the same lines as in Lemma 3.4.1. In Lemma 3.4.1, it is sufficient to show that the uniform distribution μ over subsets $S \subseteq U$ of size k is $\frac{k}{k-d+1}$-approximate $(d-1, d)$-wise independent (as in (3.10)). Here, it is sufficient to show that the same uniform distribution μ is $\frac{k}{k-l+1}$-approximate (l', l)-wise independent. For $T, T' \subseteq [n]$ of size l', l, the calculation of $\frac{\Pr_\mu[\mathcal{S} \supseteq T']}{\Pr_\mu[\mathcal{S} \supseteq T]}$ and $\frac{y^{T'}}{y^T}$ is straightforward

$$\frac{\Pr_\mu\left[\mathcal{S} \supseteq T'\right]}{\Pr_\mu\left[\mathcal{S} \supseteq T\right]} = \frac{\binom{qk-l'}{k-l'}/\binom{qk}{k}}{\binom{qk-l}{k-l}/\binom{qk}{k}} \leq \frac{(qk)^{l-l'}(k-l)!}{(k-l')!} \text{ and } \frac{y^{T'}}{y^T} = q^{l-l'} \tag{3.30}$$

Therefore, μ is α-approximate (l', l)-wise independent for

$$\alpha = \left(\frac{\Pr_{\mu}[\mathcal{S} \supseteq T']}{\Pr_{\mu}[\mathcal{S} \supseteq T]} \cdot \frac{y^T}{y^{T'}} \right)^{\frac{1}{l-l'}} \leq \left(\frac{(qk)^{l-l'}(k-l)!}{(k-l')!} q^{l'-l} \right)^{\frac{1}{l-l'}}$$
$$= \frac{k}{[(k-l')(k-l'-1)\cdots(k-l+1)]^{\frac{1}{l-l'}}} \leq \frac{k}{k-l+1}$$

as we wanted to show. □

We note that the l-proportional volume sampling in the proof of Theorem 3.5.10 can be implemented efficiently, and the proof is outlined in Section 3.6.5.

Integrality Gap

Finally, we state an integrality gap for minimizing generalized ratio objective $\left(\frac{E_{l'}(V_S V_S^\top)}{E_l(V_S V_S^\top)}\right)^{\frac{1}{l-l'}}$ over subsets $S \subseteq [n]$ of size k. The integrality gap matches our approximation ratio of our algorithm with repetitions when k is large.

Theorem 3.5.11. *For any given positive integers k, d and a pair of integers $0 \leq l' \leq l \leq d$ with $k > l'$, there exists an instance $V = [v_1, \ldots, v_n] \in \mathbb{R}^{d \times n}$ to the problem of minimizing $\left(\frac{E_{l'}(V_S V_S^\top)}{E_l(V_S V_S^\top)}\right)^{\frac{1}{l-l'}}$ over subsets $S \subseteq [n]$ of size k such that*

$$\mathrm{OPT} \geq \left(\frac{k}{k-l'} - \delta \right) \cdot \mathsf{CP}$$

for all $\delta > 0$, where OPT *denotes the value of the optimal integral solution and* CP *denotes the value of the convex program.*

This implies that the integrality gap is at least $\frac{k}{k-l'}$ for minimizing $\left(\frac{E_{l'}(V_S V_S^\top)}{E_l(V_S V_S^\top)}\right)^{\frac{1}{l-l'}}$ over subsets $S \subseteq [n]$ of size k. The theorem applies to both with and without repetitions.

Proof. The instance $V = [v_1, \ldots, v_n]$ will be the same for with and without repetitions. For each

$1 \leq i \leq d$, let e_i denote the unit vector in the direction of axis i. Choose

$$v_i = \begin{cases} \sqrt{N} \cdot e_i & \text{for } i = 1, \dots, l' \\ \\ e_i & \text{for } i = 1, \dots, l' \end{cases}$$

where $N > 0$ is a constant to be chosen later. Set $v_i, i > l$ to be at least k copies of each of these v_i for $i \leq l$, as we can make n as big as needed. Hence, we may assume that we are allowed to pick only $v_i, i \leq l$, but with repetitions.

Let S^* represent the set of vectors in OPT and y_i be the number of copies of v_i in S^* for $1 \leq i \leq l$. Clearly $y_i \geq 1$ for all $i = 1, \dots, l$ (else the objective is unbounded). The eigenvalues of $V_{S^*} V_{S^*}^\top$ are

$$\lambda(V_{S^*} V_{S^*}^\top) = (y_1 N, y_2 N, \dots, y_{l'} N, y_{l'+1}, y_{l'+2}, \dots, y_l, 0, \dots, 0)$$

Hence, both $E_{l'}(V_{S^*} V_{S^*}^\top) = e_{l'}(\lambda)$ and $E_l(V_{S^*} V_{S^*}^\top) = e_l(\lambda)$ are polynomials in variables N of degree l'.

Now let $N \to \infty$. To compute $(\text{OPT})^{l-l'} = \frac{E_{l'}(V_{S^*} V_{S^*}^\top)}{E_l(V_{S^*} V_{S^*}^\top)}$, we only need to compute the coefficient of the highest degree monomial $N^{l'}$. The coefficient of $N^{l'}$ in $e_{l'}(\lambda), e_l(\lambda)$ are exactly $\prod_{i=1}^{l'} y_i, \prod_{i=1}^{l} y_i$, and therefore

$$(\text{OPT})^{l-l'} = \frac{E_{l'}(V_{S^*} V_{S^*}^\top)}{E_l(V_{S^*} V_{S^*}^\top)} \to \frac{\prod_{i=1}^{l'} y_i}{\prod_{i=1}^{l} y_i} = \left(\prod_{i=l'+1}^{l} y_i \right)^{-1}$$

Observe that $\prod_{i=l'+1}^{l} y_i$ is maximized under the budget constraint $\sum_{i=1}^{l} y_i = |S^*| = k$ when $y_j = 1$ for $j = 1, \dots, l'$. Therefore,

$$\prod_{i=l'+1}^{l} y_i \leq \left(\frac{1}{l - l'} \sum_{i=l'+1}^{l} y_i \right)^{l-l'} = \left(\frac{k - l'}{l - l'} \right)^{l-l'}$$

where the inequality is by AM-GM. Hence, OPT is lower bounded by a quantity that converges to

$\frac{l-l'}{k-l'}$ as $N \to \infty$.

We now give a valid fractional solution x to upper bound CP for each $N > 0$. Choose

$$x_i = \begin{cases} \frac{k}{\sqrt{N}} & \text{for } i = 1, \ldots, l' \\ \frac{k - \frac{kl'}{\sqrt{N}}}{l-l'} & \text{for } i = l'+1, \ldots, l \\ 0 & \text{for } i > l \end{cases}$$

Then, eigenvalues of $V(x)V(x)^\top$ are

$$\begin{aligned} \lambda' := \lambda(V(x)V(x)^\top) &= (x_1 N, x_2 N, \ldots, x_{l'} N, x_{l'+1}, x_{l'+2}, \ldots, x_l, 0, \ldots, 0) \\ &= (k\sqrt{N}, k\sqrt{N}, \ldots, k\sqrt{N}, x_{l'+1}, x_{l'+2}, \ldots, x_l, 0, \ldots, 0) \end{aligned}$$

Now as $N \to \infty$, the dominating terms of $E_{l'}(V(x)V(x)^\top) = e_{l'}(\lambda')$ is $\prod_{i=1}^{l'}(k\sqrt{N}) = k^{l'}(\sqrt{N})^{l'}$. Also, we have

$$\begin{aligned} E_l(V(x)V(x)^\top) = e_l(\lambda') &= \prod_{i=1}^{l'}(k\sqrt{N}) \prod_{i=l'+1}^{l} x_i \\ &= k^{l'} \left(\frac{k - \frac{kl'}{\sqrt{N}}}{l-l'} \right)^{l-l'} (\sqrt{N})^{l'} \to k^{l'} \left(\frac{k}{l-l'} \right)^{l-l'} (\sqrt{N})^{l'} \end{aligned}$$

Hence,

$$\mathsf{CP} \le \left(\frac{E_{l'}(V(x)V(x)^\top)}{E_l(V(x)V(x)^\top)} \right)^{l-l'} \to \frac{l-l'}{k}$$

Therefore, $\frac{\mathrm{OPT}}{\mathsf{CP}}$ is lower bounded by a ratio which converges to $\frac{l-l'}{k-l'} \cdot \frac{k}{l-l'} = \frac{k}{k-l'}$. □

3.6 Efficient Algorithms

In this section, we outline efficient sampling algorithms, as well as deterministic implementations of our rounding algorithms, both for with and without repetition settings.

3.6.1 Efficient Randomized Proportional Volume

Given a vector $\lambda \in \mathbb{R}_+^n$, we show that proportional volume sampling with $\mu(S) \propto \lambda^S$ for $S \in \mathcal{U}$, where $\mathcal{U} \in \{\mathcal{U}_k, \mathcal{U}_{\leq k}\}$ can be done in time polynomial in the size n of the ground set. We start by stating a lemma which is very useful both for the sampling algorithms and the deterministic implementations.

Lemma 3.6.1. *Let $\lambda \in \mathbb{R}_+^n, v_1, \ldots, v_n \in \mathbb{R}^d$, and $V = [v_1, \ldots, v_n]$. Let $I, J \subseteq [n]$ be disjoint. Let $1 \leq k \leq n, 0 \leq d_0 \leq d$. Consider the following function*

$$F(t_1, t_2, t_3) = \det\left(I_n + t_1 \textbf{diag}(y) + t_1 t_2 \textbf{diag}(y)^{1/2} V V^\top \textbf{diag}(y)^{1/2}\right)$$

where $t_1, t_2, t_3 \in \mathbb{R}$ are indeterminate, I_n is the $n \times n$ identity matrix, and $y \in \mathbb{R}^n$ with

$$y_i = \begin{cases} \lambda_i t_3, & \textit{if } i \in I \\ 0, & \textit{if } i \in J \\ \lambda_i, & \textit{otherwise} \end{cases} .$$

Then $F(t_1, t_2, t_3)$ is a polynomial and the quantity

$$\sum_{|S|=k, I \subseteq S, J \cap S = \emptyset} \lambda^S \sum_{|T|=d_0, T \subseteq S} \det(V_T^\top V_T) \tag{3.31}$$

is the coefficient of the monomial $t_1^k t_2^{d_0} t_3^{|I|}$. Moreover, this quantity can be computed in $O\left(n^3 d_0 k |I| \cdot \log(d_0 k |I|)\right)$ number of arithmetic operations.

Proof. Let us first fix some $S \subseteq [n]$. Then we have

$$\sum_{|T|=d_0, T \subseteq S} \det(V_T^\top V_T) = E_{d_0}(V_S^\top V_S) = [t_2^{d_0}] \det(I_S + t_2 V_S V_S^\top),$$

where the notation $[t_2^{d_0}]p(t_2)$ denotes the coefficient of t^{d_0} in the polynomial $p(t_2) = \det(I_S + t_2 V_S V_S^\top)$. The first equality is just Cauchy-Binet, and the second one is standard and follows from the Leibniz formula for the determinant. Therefore, (3.31) equals

$$[t_2^{d_0}] \sum_{|S|=k, I\subseteq S, J\cap S=\emptyset} \lambda^S \det(I_S + t_2 V_S V_S^\top).$$

To complete the proof, we establish the following claim.

Claim 1. *Let L be an $n \times n$ matrix, and let λ, I, J, k, y be as in the statement of the Lemma. Then,*

$$\begin{aligned}\sum_{|S|=k, I\subseteq S, J\cap S=\emptyset} \lambda^S \det(L_{S,S}) &= [t_3^{|I|}] E_k\left(\text{diag}(y)^{1/2} L \ \text{diag}(y)^{1/2}\right) \\ &= [t_1^k t_3^{|I|}] \det\left(I_n + t_1 \text{diag}(y)^{1/2} L \ \text{diag}(y)^{1/2}\right).\end{aligned}$$

Proof. By Cauchy-Binet,

$$\begin{aligned}E_k\left(\text{diag}(y)^{1/2} L \ \text{diag}(y)^{1/2}\right) &= \sum_{|S|=k} y^S \det(L_{S,S}) \\ &= \sum_{|S|=k, J\cap S=\emptyset} t_3^{|S\cap I|} \lambda^S \det(L_{S,S}).\end{aligned}$$

The first equality follows. The second is, again, a consequence of the Leibniz formula for the determinant. □

Plugging in $L = I_n + t_2 V V^\top$ in Claim 1 gives that (3.31) equals

$$\begin{aligned}&[t_1^k t_2^{d_0} t_3^{|I|}] \det\left(I_n + t_1 \text{diag}(y)^{1/2} (I_n + t_2 V V^\top) \text{diag}(y)^{1/2}\right) \\ &\qquad = [t_1^k t_2^{d_0} t_3^{|I|}] \det\left(I_n + t_1 \text{diag}(y) + t_1 t_2 \text{diag}(y)^{1/2} V V^\top \text{diag}(y)^{1/2}\right).\end{aligned}$$

This completes the proof. For the running time, the standard computation time of matrix multiplication and determinant of $n \times n$ matrices is $O(n^3)$ entry-wise arithmetic operations. We need to keep all monomials in the form $t_1^a t_2^b t_3^c$ where $a \leq k, b \leq d_0, c \leq |I|$, of which there are $O(d_0 k|I|)$ of those. By representing multivariate monomials in single variable ([Pan94]), we may use Fast Fourier Transform to do one polynomial multiplication of entries of the matrix in $O\left(d_0 k|I| \cdot \log(d_0 k|I|)\right)$ number of arithmetic operations. This gives the total running time of $O\left(n^3 d_0 k|I| \cdot \log(d_0 k|I|)\right)$. □

Using the above lemma, we now prove the following theorem that will directly imply Lemma 3.1.10.

Theorem 3.6.2. *Let $\lambda \in \mathbb{R}_+^n, v_1, \ldots, v_n \in \mathbb{R}^d, 1 \leq k \leq n, \mathcal{U} \in \{\mathcal{U}_k, \mathcal{U}_{\leq k}\}$, and $V = [v_1, \ldots, v_n]$. Then there is a randomized algorithm $\mathcal{A}$ which outputs $\mathcal{S} \in \mathcal{U}$ such that*

$$\Pr_{\mathcal{S} \sim \mathcal{A}}[\mathcal{S} = S] = \frac{\lambda^S \det(V_S V_S^\top)}{\sum_{S' \in \mathcal{U}} \lambda^{S'} \det(V_{S'} V_{S'}^\top)} =: \mu'(S)$$

That is, the algorithm correctly implements proportional volume sampling μ' with hard-core measure μ on $\mathcal{U}$ with parameter λ. Moreover, the algorithm runs in $O\left(n^4 dk^2 \log(dk)\right)$ number of arithmetic operations.

Observation 3.6.3. [WYS16] shows that we may assume that the support of an extreme fractional solution of convex relaxation has size at most $k + d^2$. Thus, the runtime of proportional volume sampling is $O\left((k + d^2)^4 dk^2 \log(dk)\right)$. While the degrees in d, k are not small, this runtime is independent of n.

Observation 3.6.4. It is true in theory and observed in practice that solving the continuous relaxation rather than the rounding algorithm is a bottleneck in computation time, as discussed in [ALSW17a]. In particular, solving the continuous relaxation of A-optimal design takes $O\left(n^{2+\omega} \log n\right)$ number of iterations by standard ellipsoid method and $O\left((n + d^2)^{3.5}\right)$ number of iterations by SDP, where $O(n^\omega)$ denotes the runtime of $n \times n$ matrix multiplication. In most applications where $n >> k$, these running times dominates one of proportional volume sampling.

Proof. We can sample by starting with an empty set $S = \emptyset$. Then, in each step $i = 1, 2, \ldots, n$, the algorithm decides with the correct probability

$$\Pr_{\mathcal{S}\sim\mu'}\left[i \in \mathcal{S} | I \subseteq \mathcal{S}, J \cap \mathcal{S} = \emptyset\right]$$

whether to include i in S or not, given that we already know that we have included I in S and excluded J from S from previous steps $1, 2, \ldots, i-1$. Let $I' = I \cup \{i\}$. This probability equals to

$$\begin{aligned}\Pr_{\mathcal{S}\sim\mu'}\left[i \in \mathcal{S} | I \subseteq \mathcal{S}, J \cap \mathcal{S} = \emptyset\right] &= \frac{\Pr_{\mathcal{S}\sim\mu'}\left[I' \subseteq \mathcal{S}, J \cap \mathcal{S} = \emptyset\right]}{\Pr_{\mathcal{S}\sim\mu'}\left[I \subseteq \mathcal{S}, J \cap \mathcal{S} = \emptyset\right]} \\ &= \frac{\sum_{S\in\mathcal{U}, I'\subseteq S, J\cap S=\emptyset} \lambda^S \det(V_S V_S^\top)}{\sum_{S\in\mathcal{U}, I\subseteq S, J\cap S=\emptyset} \lambda^S \det(V_S V_S^\top)} \\ &= \frac{\sum_{S\in\mathcal{U}, I'\subseteq S, J\cap S=\emptyset} \lambda^S \sum_{|R|=d, R\subset S} \det(V_R V_R^\top)}{\sum_{S\in\mathcal{U}, I\subseteq S, J\cap S=\emptyset} \lambda^S \sum_{|R|=d, R\subset S} \det(V_R V_R^\top)}\end{aligned}$$

where we apply the Cauchy-Binet formula in the last equality. For $\mathcal{U} = \mathcal{U}_k$, both the numerator and denominator are summations over S restricted to $|S| = k$, which can be computed in $O\left(n^3 dk^2 \log(dk)\right)$ number of arithmetic operations by Lemma 3.6.1. For the case $\mathcal{U} = \mathcal{U}_{\leq k}$, we can evaluate summations in the numerator and denominator restricted to $|S| = k_0$ for each $k_0 = 1, 2, \ldots k$ by computing polynomial $F(t_1, t_2, t_3)$ in Lemma 3.6.1 only once, and then sum those quantities over k_0. □

3.6.2 Efficient Deterministic Proportional Volume

We show that for hard-core measures there is a deterministic algorithm that achieves the same objective value as the expected objective value achieved by proportional volume sampling. The basic idea is to use the method of conditional expectations.

Theorem 3.6.5. *Let $\lambda \in \mathbb{R}_+^n, v_1, \ldots, v_n \in \mathbb{R}^d, 1 \leq k \leq n, \mathcal{U} \in \{\mathcal{U}_k, \mathcal{U}_{\leq k}\}$, and $V = [v_1, \ldots, v_n]$.*

Then there is a deterministic algorithm $\mathcal{A}'$ which outputs $S^* \subseteq [n]$ of size k such that

$$\operatorname{tr}\left(V_{S^*}V_{S^*}^\top\right)^{-1} \geq \mathop{\mathbb{E}}_{\mu'}\left[\operatorname{tr}\left(V_{\mathcal{S}}V_{\mathcal{S}}^\top\right)^{-1}\right]$$

where μ' is the probability distribution defined by $\mu'(S) \propto \lambda^S \det(V_S V_S^\top)$ for all $S \in \mathcal{U}$. Moreover, the algorithm runs in $O\left(n^4 dk^2 \log(dk)\right)$ number of arithmetic operations.

Again, with the assumption that $n \leq k + d^2$ (Observation 3.6.3), the runtime for deterministic proportional volume sampling is $O\left((k + d^2)^4 dk^2 \log(dk)\right)$.

Proof. To prove the theorem, we derandomize the sampling algorithm in Theorem 3.6.2 by the method of conditional expectations. The deterministic algorithm starts with $S^* = \emptyset$, and then chooses, at each step $i = 1, 2, \ldots, n$, whether to pick i to be in S^* or not, given that we know from previous steps to include or exclude each element $1, 2, \ldots, i-1$ from S^*. The main challenge is to calculate exactly the quantity of the form

$$X(I, J) := \mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\operatorname{tr}\left(V_{\mathcal{S}}V_{\mathcal{S}}^\top\right)^{-1} | I \subset \mathcal{S}, J \cap \mathcal{S} = \emptyset\right]$$

where $I, J \subseteq [n]$ are disjoint. If we can efficiently calculate the quantity of such form, the algorithm can, at each step $i = 1, 2, \ldots, n$, calculate $X(I' \cup \{i\}, J')$ and $X(I', J' \cup \{i\})$ where $I', J' \subseteq [i-1]$ denote elements we have decided to pick and not to pick, respectively, and then include i to S^* if and only if $X(I' \cup \{i\}, J') \geq X(I', J' \cup \{i\})$.

Note that the quantity $X(I,J)$ equals

$$\begin{aligned}
\mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\operatorname{tr}\left(V_{\mathcal{S}}V_{\mathcal{S}}^{\top}\right)^{-1}|I\subset\mathcal{S}, J\cap\mathcal{S}=\emptyset\right] &= \sum_{\substack{S\in\mathcal{U},\\ I\subseteq S, J\cap S=\emptyset}} \Pr_{\mu'}\left[\mathcal{S}=S|I\subseteq\mathcal{S},\mathcal{S}\cap J=\emptyset\right]\operatorname{tr}\left[(V_SV_S^{\top})^{-1}\right] \\
&= \sum_{\substack{S\in\mathcal{U},\\ I\subseteq S, J\cap S=\emptyset}} \frac{\lambda^S\det(V_SV_S^{\top})}{\sum_{S'\in\mathcal{U},I\subseteq S,J\cap S=\emptyset}\lambda^{S'}\det(V_{S'}V_{S'}^{\top})}\operatorname{tr}\left[(V_SV_S^{\top})^{-1}\right] \\
&= \frac{\sum_{S\in\mathcal{U},I\subseteq S,J\cap S=\emptyset}\lambda^S E_{d-1}(V_SV_S^{\top})}{\sum_{S\in\mathcal{U},I\subseteq S,J\cap S=\emptyset}\lambda^S\sum_{|R|=d,R\subset S}\det(V_RV_R^{\top})} \\
&= \frac{\sum_{S\in\mathcal{U},I\subseteq S,J\cap S=\emptyset}\lambda^S\sum_{|T|=d-1,T\subset S}\det(V_T^{\top}V_T)}{\sum_{S\in\mathcal{U},I\subseteq S,J\cap S=\emptyset}\lambda^S\sum_{|R|=d,R\subset S}\det(V_RV_R^{\top})}
\end{aligned}$$

where we write inverse of trace as ratio of symmetric polynomials of eigenvalues in the third equality and use Cauchy-Binet formula for the third and the fourth equality. The rest of the proof is now identical to the proof of Theorem 3.6.2, except with different parameters $d_0 = d-1, d$ in $f(t_1,t_2,t_3)$ when applying Lemma 3.6.1. □

3.6.3 Efficient Randomized Implementation of $k/(k-d+1)$-Approximation Algorithm With Repetitions

First, we need to state several Lemmas needed to compute particular sums. The main motivation that we need a different method from Section 3.6.1 and 3.6.2 to compute a similar sum is that we want to allow the ground set U of indices of all copies of vectors to have an exponential size. This makes Lemma 3.6.1 not useful, as the matrix needed to be computed has dimension $|U|\times|U|$. The main difference, however, is that the parameter λ is now a constant, allowing us to obtain sums by computing a more compact $d\times d$ matrix.

Lemma 3.6.6. *Let $V=[v_1,\ldots,v_m]$ be a matrix of vectors $v_i\in\mathbb{R}^d$ with $n\geq d$ distinct vectors. Let $F\subseteq[m]$ and let $0\leq r\leq d$ and $0\leq d_0\leq d$ be integers. Then the quantity*

$\sum_{|T|=d_0,|F\cap R|=r}\det(V_T^\top V_T)$ is the coefficient of $t_1^{d-d_0}t_2^{d_0-r}t_3^r$ in

$$f(t_1,t_2,t_3)=\det\left(t_1I_d+\sum_{i\in F}t_3v_iv_i^\top+\sum_{i\notin F}t_2v_iv_i^\top\right) \tag{3.32}$$

where $t_1,t_2,t_3\in\mathbb{R}$ are indeterminate and I_d is the $d\times d$ identity matrix. Furthermore, this quantity can be computed in $O\left(n(d-d_0+1)d_0^2d^2\log d\right)$ number of arithmetic operations.

Proof. First, note that $\det\left(t_1I+\sum_{i\in F}t_3v_iv_i^\top+\sum_{i\notin F}t_2v_iv_i^\top\right)=\prod_{i=1}^d(t_1+\nu_i)$ where $\nu(M)=\{\nu_1,\ldots,\nu_d\}$ is the vector of eigenvalues of the matrix $M=\sum_{i\in F}t_3v_iv_i^\top+\sum_{i\notin F}t_2v_iv_i^\top$. Hence, the coefficient of $t_1^{d-d_0}$ in $\det\left(t_1I+\sum_{i\in F}t_3v_iv_i^\top+\sum_{i\notin F}t_2v_iv_i^\top\right)$ is $e_{d_0}(\nu(M))$.

Next, observe that M is in the form $V'V'^\top$ where V' is the matrix where columns are $\sqrt{t_3}v_i$, $i\in F$ and $\sqrt{t_2}v_i, i\notin F$. Applying Cauchy-Binet to $E_{d_0}(V'V'^\top)$, we get

$$\begin{aligned}E_{d_0}\left(\sum_{i\in F}t_3v_iv_i^\top+\sum_{i\notin F}t_2v_iv_i^\top\right)=E_{d_0}(V'V'^\top)&=\sum_{|T|=d_0}\det(V_T'^\top V_T')\\&=\sum_{l=0}^{|F|}\sum_{|T|=d_0,|T\cap F|=l}\det(V_T'^\top V_T')\\&=\sum_{l=0}^{|F|}\sum_{|T|=d_0,|T\cap F|=l}t_3^lt_2^{d_0-l}\det(V_T^\top V_T),\end{aligned}$$

where we use Lemma 3.2.1 for the last equality. The desired quantity $\sum_{|T|=d_0,|F\cap R|=r}\det(V_T^\top V_T)$ is then exactly the coefficient at $l=r$ in the sum on the right hand side.

To compute the running time, since there are only n distinct vectors, we may represent sets V,F compactly with distinct v_i's and number of copies of each distinct v_i's. Therefore, computing the matrix sum takes $O\left(nd^2\right)$ entry-wise operations. Next, the standard computation time of determinant of $d\times d$ matrix is $O(d^3)$ entry-wise arithmetic operations. This gives a total of $O\left(nd^2+d^3\right)=O\left(nd^2\right)$ entry-wise operations.

For each entry-wise operation, we keep all monomials in the form $t_1^at_2^bt_3^c$ where $a\leq d-d_0, b\leq$

$d_0 - r, c \leq r$, of which there are $O((d-d_0+1)d_0^2)$. By representing multivariate monomials in single variable ([Pan94]) of degree $O((d-d_0+1)d_0^2)$, we may use Fast Fourier Transform to do one polynomial multiplication of entries of the matrix in $O\left((d-d_0+1)d_0^2 \log d\right)$ number of arithmetic operations. This gives the total runtime of $O\left(n(d-d_0+1)d_0^2 d^2 \log d\right)$ arithmetic operations. □

Lemma 3.6.7. *Let $V = [v_1, \ldots, v_m]$ be a matrix of vectors $v_i \in \mathbb{R}^d$ with $n \geq d$ distinct vectors. Let $F \subseteq [m]$ and let $0 \leq r \leq d$ and $0 \leq d_0 \leq d$ be integers. There is an algorithm to compute $\sum_{|S|=k, S\supseteq F} E_{d_0}(V_S V_S^\top)$ with $O\left(n(d-d_0+1)d_0^2 d^2 \log d\right)$ number of arithmetic operations.*

Proof. We apply Cauchy-Binet:

$$\begin{aligned}
\sum_{|S|=k, S\supseteq F} E_{d_0}(V_S V_S^T) &= \sum_{|S|=k, S\supseteq F} \sum_{|T|=d_0, T\subset S} \det(V_T^\top V_T) \\
&= \sum_{|T|=d_0} \det(V_T^\top V_T) \binom{m - |F| - d_0 + |F \cap T|}{k - |F| - d_0 + |F \cap T|} \\
&= \sum_{r=0}^{d} \binom{m - |F| - d_0 + r}{k - |F| - d_0 + r} \sum_{|T|=d_0, |F\cap T|=r} \det(V_T^\top V_T)
\end{aligned}$$

where we change the order of summations for the second equality, and enumerate over possible sizes of $F \cap T$ to get the third equality. We compute $f(t_1, t_2, t_3)$ in Lemma 3.6.6 once with $O\left(n(d-d_0+1)d_0^2 d^2 \log d\right)$ number of arithmetic operations, so we obtain values of $\sum_{|T|=d_0, |F\cap T|=r} \det(V_T^\top V_T)$ for all $r = 0, \ldots, d_0$. The rest is a straightforward calculation. □

We now present an efficient sampling procedure for Algorithm 3.2. We want to sample S proportional to $\det(W_S W_S^\top)$. The set S is a subset of all copies of at most n distinct vectors, and there can be exponentially many copies. However, the key is that the quantity $f(t_1, t_2, t_3)$ in (3.32) is still efficiently computable because exponentially many of these copies of vectors are the same.

Theorem 3.6.8. *Given inputs $n, d, k, \epsilon, x \in \mathbb{R}_+^n$ with $\sum_{i=1}^n x_i = k$, and vectors $v_1, \ldots, v_n$ to Algorithm 3.2 we define q, U, W as in Algorithm 3.2. Then, there exists an implementation $\mathcal{A}$*

that samples $\mathcal{S}$ from the distribution μ' over all subsets $S \subseteq U$ of size k, where μ' is defined by $\Pr_{\mathcal{S}\sim\mu'}[\mathcal{S} = S] \propto \det(W_S W_S^\top)$ for each $S \subseteq U, |S| = k$. Moreover, $\mathcal{A}$ runs in $O\left(n^2 d^4 k \log d\right)$ number of arithmetic operations.

Theorem 3.6.8 says that steps (4)-(5) in Algorithm 3.2 can be efficiently implemented. Other steps except (4)-(5) obviously use $O\left(n^2 d^4 k \log d\right)$ number of arithmetic operations, so the above statement implies that Algorithm 3.2 runs in $O\left(n^2 d^4 k \log d\right)$ number of arithmetic operations. Again, by Observation 3.6.3, the number of arithmetic operations is in fact $O\left((k + d^2)^2 d^4 k \log d\right)$.

Proof. Let $m_i = q x_i'$ be the number of copies of vector v_i (recall that $q = \frac{2n}{\epsilon k}$). Let $w_{i,j}$ denote the jth copy of vector v_i. Write $U = \{(i,j) : i \in [n], j \in [m_i]\}$ be the new set of indices after the copying procedure. Denote $\mathcal{S}$ a random subset (not multiset) of U that we want to sample. Write W as the matrix with columns $w_{i,j}$ for all $(i,j) \in U$. Let $E_i = \{w_{i,j} : j = 1, \ldots, m_i\}$ be the set of copies of vector v_i. For any $A \subseteq U$, we say that A *has* k_i *copies of* v_i to mean that $|A \cap E_i| = k_i$.

We can define the sampling algorithm $\mathcal{A}$ by sampling, at each step $t = 1, \ldots, n$, how many copies of v_i are to be included in $\mathcal{S} \subseteq U$. Denote μ' the volume sampling on W we want to sample. The problem then reduces to efficiently computing

$$\begin{aligned}
&\Pr_{\mu'}\left[\mathcal{S} \text{ has } k_t \text{ copies of } v_t | \mathcal{S} \text{ has } k_i \text{ copies of } v_i, \forall i = 1, \ldots, t-1\right] \\
&\qquad = \frac{\Pr_{\mu'}\left[\mathcal{S} \text{ has } k_i \text{ copies of } v_i, \forall i = 1, \ldots, t\right]}{\Pr_{\mu'}\left[\mathcal{S} \text{ has } k_i \text{ copies of } v_i, \forall i = 1, \ldots, t-1\right]}
\end{aligned} \tag{3.33}$$

for each $k_t = 0, 1, \ldots, k - \sum_{i=1}^{t-1} k_i$. Thus, it suffices to efficiently compute quantity (3.33) for any given $1 \le t \le n$ and $k_1, \ldots, k_t$ such that $\sum_{i=1}^{t} k_i \le k$.

We now fix $t, k_1, \ldots, k_t$. Note that for any $i \in [n]$, getting any set of k_i copies of v_i is the same, i.e. events $\mathcal{S} \cap E_i = F_i$ and $\mathcal{S} \cap E_i = F_i'$ under $\mathcal{S} \sim \mu'$ have the same probability for any subsets $F_i, F_i' \subseteq E_i$ of the same size. Therefore, we fix one set of k_i copies of v_i to be

$F_i = \{w_{i,j} : j = 1, \ldots, k_i\}$ for all $i \in [n]$ and obtain

$$\mathsf{Pr}\left[\mathcal{S} \text{ has } k_i \text{ copies of } v_i, \forall i = 1, \ldots, t\right] = \prod_{i=1}^{t} \binom{m_i}{k_i} \mathsf{Pr}\left[\mathcal{S} \cap E_i = F_i, \forall i = 1, \ldots t\right]$$

Therefore, (3.33) equals

$$\frac{\prod_{i=1}^{t} \binom{m_i}{k_i} \mathsf{Pr}\left[\mathcal{S} \cap E_i = F_i, \forall i = 1, \ldots t\right]}{\prod_{i=1}^{t-1} \binom{m_i}{k_i} \mathsf{Pr}\left[\mathcal{S} \cap E_i = F_i, \forall i = 1, \ldots t-1\right]} = \binom{m_t}{k_t} \frac{\sum_{|S|=k, S\cap E_i = F_i, \forall i=1,\ldots t} \det(W_S W_S^\top)}{\sum_{|S|=k, S\cap E_i = F_i, \forall i=1,\ldots t-1} \det(W_S W_S^\top)} \tag{3.34}$$

To compute the numerator, define W' a matrix of vectors in W restricted to indices $U \backslash \left(\bigcup_{i=1}^{t} E_i \setminus F_i\right)$, and $F := \bigcup_{i=1}^{t} F_i$, then we have

$$\sum_{|S|=k, S\subseteq W, S\cap E_i = F_i, \forall i=1,\ldots t} \det(W_S W_S^\top) = \sum_{|S|=k, S\subseteq W', S\supseteq F} \det(W'_S {W'_S}^\top) \tag{3.35}$$

By Lemma 3.6.7, the number of arithmetic operations to compute (3.35) is $O\left(n(d-d_0+1)d_0^2 d^2 \log d\right) = O\left(nd^4 \log d\right)$ (by applying $d_0 = d$). Therefore, because in each step $t = 1, 2, \ldots, n$, we compute (3.33) at most k times for different values of k_t, the total number of arithmetic operations for sampling algorithm $\mathcal{A}$ is $O\left(n^2 d^4 k \log d\right)$. □

Remark 3.6.9. Although Theorem 3.6.8 and Observation 3.6.3 imply that randomized rounding for *A*-optimal design with repetition takes $O\left((k+d^2)^2 d^4 k \log d\right)$ number of arithmetic operations, this does not take into account the size of numbers used in the computation which may scale with input ϵ. It is not hard to see that the sizes of coefficients $f(t_1, t_2, t_3)$ in Lemma 3.6.6, of the number $\binom{m-|F|-d_0+r}{k-|F|-d_0+r}$ in the proof of Lemma 3.6.7, and of $\binom{m_t}{k_t}$ in (3.34) scale linearly with $O(k \log(m))$ where $m = \sum_{i=1}^{n} m_i$. As we apply $m \leq qk = \frac{2n}{\epsilon}$ in the proof of Theorem 3.6.8, the runtime of randomized rounding for *A*-optimal design with repetition, after taking into account the size of numbers in the computation, has an extra factor of $k \log(\frac{n}{\epsilon})$ and becomes

$O\left((k+d^2)^2 d^4 k^2 \log d \log(\frac{k+d^2}{\epsilon}))\right)$.

3.6.4 Efficient Deterministic Implementation of $k/(k-d+1)$-Approximation Algorithm With Repetitions

We show a *deterministic* implementation of proportional volume sampling used for the $\frac{k}{k-d+1}$-approximation algorithm with repetitions. In particular, we derandomized the efficient implementation of steps (4)-(5) of Algorithm 3.2, and show that the running time of deterministic version is the same as that of the randomized one.

Theorem 3.6.10. *Given inputs $n, d, k, \epsilon, x \in \mathbb{R}_+^n$ with $\sum_{i=1}^n x_i = k$, and vectors $v_1, \ldots, v_n$ to Algorithm 3.2, we define q, U, W as in Algorithm 3.2. Then, there exists a deterministic algorithm $\mathcal{A}'$ that outputs $S^* \subseteq U$ of size k such that*

$$\operatorname{tr}\left(W_{S^*} W_{S^*}^\top\right)^{-1} \geq \mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\operatorname{tr}\left(W_{\mathcal{S}} W_{\mathcal{S}}^\top\right)^{-1}\right]$$

where μ' is a distribution over all subsets $S \subseteq U$ of size k defined by $\mu'(S) \propto \det(W_S W_S^\top)$ for each set $S \subseteq U$ of size k. Moreover, $\mathcal{A}'$ runs in $O\left(n^2 d^4 k \log d\right)$ number of arithmetic operations.

Again, together with Observation 3.6.3 and Remark 3.6.9, Theorem 3.6.10 implies that the $\frac{k}{k-d+1}$-approximation algorithm for *A*-optimal design with repetitions can be implemented deterministically in $O\left((k+d^2)^2 d^4 k \log d\right)$ number of arithmetic operations and, after taking into account the size of numbers in the computation, in $O\left((k+d^2)^2 d^4 k^2 \log d \log(\frac{k+d^2}{\epsilon})\right)$ time.

Proof. We can define the deterministic algorithm $\mathcal{A}'$ by deciding, at each step $t = 1, \ldots, n$, how many copies of v_i are to be included in $S^* \subseteq U$. The problem then reduces to efficiently computing

$$X(k_1, \ldots, k_t) := \mathop{\mathbb{E}}_{\mu'}\left[\operatorname{tr}\left(W_{\mathcal{S}} W_{\mathcal{S}}^\top\right)^{-1} | \mathcal{S} \text{ has } k_i \text{ copies of } v_i, \forall i = 1, \ldots, t-1, t\right] \tag{3.36}$$

where $k_1, \ldots, k_{t-1}$ is already decided by previously steps of the algorithm, and now we compute (3.36) for each $k_t = 0, 1, \ldots, k - \sum_{i=1}^{t-1} k_i$. $\mathcal{A}'$ then chooses value of k_t which maximizes (3.36) to complete step t.

Recall the definitions from proof of Theorem 3.6.8 that F_i, E_i are the sets of fixed k_i copies and all copies of v_i, respectively, W' is the matrix of vectors in W restricted to indices $U \setminus \left(\bigcup_{i=1}^{t} E_i \setminus F_i\right)$, and $F := \bigcup_{i=1}^{t} F_i$. Consider that

$$\begin{aligned}
X(k_1, \ldots, k_t) &= \sum_{\substack{S \subseteq U; |S|=k; \\ |S \cap E_i| = k_i, \forall i=1,\ldots,t}} \underset{\mu'}{\mathsf{Pr}} \left[\mathcal{S} = S | \mathcal{S} \text{ has } k_i \text{ copies of } v_i, \forall i = 1, \ldots, t\right] \operatorname{tr}\left[(W_S W_S^\top)^{-1}\right] \\
&= \sum_{\substack{S \subseteq U; |S|=k; \\ |S \cap E_i| = k_i, \forall i=1,\ldots,t}} \frac{\det(W_S W_S^\top)}{\sum_{S' \subseteq U; |S'|=k; |S' \cap E_i| = k_i, \forall i=1,\ldots,t} \det(W_{S'} W_{S'}^\top)} \operatorname{tr}\left[(W_S W_S^\top)^{-1}\right] \\
&= \frac{\sum_{S \subseteq U; |S|=k; |S \cap E_i| = k_i, \forall i=1,\ldots,t} E_{d-1}(W_S W_S^\top)}{\sum_{S \subseteq U; |S|=k; |S \cap E_i| = k_i, \forall i=1,\ldots,t} \det(W_S W_S^\top)} \\
&= \frac{\prod_{i=1}^{t} \binom{m_i}{k_i} \sum_{S \subseteq U; |S|=k; S \supseteq F} E_{d-1}(W'_S {W'_S}^\top)}{\prod_{i=1}^{t} \binom{m_i}{k_i} \sum_{S \subseteq U; |S|=k; S \supseteq F} \det(W'_S {W'_S}^\top)} \\
&= \frac{\sum_{S \subseteq U; |S|=k; S \supseteq F} E_{d-1}(W'_S {W'_S}^\top)}{\sum_{S \subseteq U; |S|=k; S \supseteq F} \det(W'_S {W'_S}^\top)}
\end{aligned}$$

By Lemma 3.6.7, we can compute the numerator and denominator in $O\left(n(d - d_0 + 1) d_0^2 d^2 \log d\right) = O\left(nd^4 \log d\right)$ (by applying $d_0 = d - 1, d$) number of arithmetic operations. Therefore, because in each step $t = 1, 2, \ldots, n$, we compute (3.36) at most k times for different values of k_t, the total number of arithmetic operations for sampling algorithm $\mathcal{A}$ is $O\left(n^2 d^4 k \log d\right)$. □

3.6.5 Efficient Implementations for the Generalized Ratio Objective

In Section 3.6.1-3.6.2 we obtain efficient randomized and deterministic implementations of proportional volume sampling with measure μ when μ is a hard-core distribution over all subsets $S \in \mathcal{U}$ (where $\mathcal{U} \in \{\mathcal{U}_k, \mathcal{U}_{\leq k}\}$) with any given parameter $\lambda \in \mathbb{R}_+^n$. Both implementations run in

$O\left(n^4dk^2\log(dk)\right)$ number of arithmetic operations. In Section 3.6.3-3.6.4, we obtain efficient randomized and deterministic implementations of proportional volume sampling over exponentially-sized matrix $W=[w_{i,j}]$ of m vectors containing n *distinct* vectors in $O\left(n^2d^4k\log d\right)$ number of arithmetic operations. In this section, we show that the results from Section 3.6.1-3.6.4 generalize to proportional l-volume sampling for generalized ratio problem.

Theorem 3.6.11. *Let n,d,k be positive integers, $\lambda\in\mathbb{R}_+^n$, $\mathcal{U}\in\{\mathcal{U}_k,\mathcal{U}_{\leq k}\}$, $V=[v_1,\ldots,v_n]\in\mathbb{R}^{d\times n}$, and $0\leq l'<l\leq d$ be a pair of integers. Let μ' be the l-proportional volume sampling distribution over $\mathcal{U}$ with hard-core measure μ of parameter λ, i.e. $\mu'(S)\propto\lambda^SE_l\left(V_SV_S^\top\right)$ for all $S\in\mathcal{U}$. There are*

- *an implementation to sample from μ' that runs in $O\left(n^4lk^2\log(lk)\right)$ number of arithmetic operations, and*
- *a deterministic algorithm that outputs a set $S^*\in\mathcal{U}$ of size k such that*

$$\left(\frac{E_{l'}(V_{S^*}V_{S^*}^\top)}{E_l(V_{S^*}V_{S^*}^\top)}\right)^{\frac{1}{l-l'}}\geq\mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\left(\frac{E_{l'}(V_\mathcal{S}V_\mathcal{S}^\top)}{E_l(V_\mathcal{S}V_\mathcal{S}^\top)}\right)^{\frac{1}{l-l'}}\right]\tag{3.37}$$

  that runs in $O\left(n^4lk^2\log(lk)\right)$ number of arithmetic operations.

Moreover, let $W=[w_{i,j}]$ be a matrix of m vectors where $w_{i,j}=v_i$ for all $i\in[n]$ and j. Denote U the index set of W. Let μ' be the l-proportional volume sampling over all subsets $S\subseteq U$ of size k with measure μ that is uniform, i.e. $\mu'(S)\propto E_l\left(W_SW_S^\top\right)$ for all $S\subseteq U,|S|=k$. There are

- *an implementation to sample from μ' that runs in $O\left(n^2(d-l+1)l^2d^2k\log d\right)$ number of arithmetic operations, and*
- *a deterministic algorithm that outputs a set $S^*\in\mathcal{U}$ of size k such that*

$$\left(\frac{E_{l'}(W_{S^*}W_{S^*}^\top)}{E_l(W_{S^*}W_{S^*}^\top)}\right)^{\frac{1}{l-l'}}\geq\mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\left(\frac{E_{l'}(W_\mathcal{S}W_\mathcal{S}^\top)}{E_l(W_\mathcal{S}W_\mathcal{S}^\top)}\right)^{\frac{1}{l-l'}}\right]\tag{3.38}$$

that runs in $O\left(n^2\left((d-l'+1)l'^2+(d-l+1)l^2\right)d^2k\log d\right)$ *number of arithmetic operations.*

As in Observation 3.6.3, note that we can replace $n = k + d^2$ in all running times in Theorem 3.6.11 so that running times of all variants of proportional volume sampling are independent of n. We also note, as in Remark 3.6.9, that running times of l-proportional volume sampling over m vectors with n distinct vectors has an extra factor of $k\log m$ after taking into account the size of numbers in computation, allowing us to do sampling over exponential-sized ground set $[m]$.

Proof. By the convexity of $f(z) = z^{l-l'}$ over positive reals z, we have $\mathbb{E}\left[X\right] \geq \left(\mathbb{E}\left[X^{\frac{1}{l-l'}}\right]\right)^{l-l'}$ for a nonnegative random variable X. Therefore, to show (3.37), it is sufficient to show that

$$\frac{E_{l'}(V_{S^*}V_{S^*}^\top)}{E_l(V_{S^*}V_{S^*}^\top)} \geq \underset{\mathcal{S}\sim\mu'}{\mathbb{E}}\left[\frac{E_{l'}(V_{\mathcal{S}}V_{\mathcal{S}}^\top)}{E_l(V_{\mathcal{S}}V_{\mathcal{S}}^\top)}\right] \tag{3.39}$$

That is, it is enough to derandomized with respect to the objective $\frac{E_{l'}(V_{\mathcal{S}}V_{\mathcal{S}}^\top)}{E_l(V_{\mathcal{S}}V_{\mathcal{S}}^\top)}$, and the same is true for showing (3.38). Hence, we choose to calculate the conditional expectations with respect to this objective.

We follow the exact same calculation for l-proportional volume sampling for generalized ratio objective as original proofs of efficient implementations of all four algorithms in A-optimal objective. We observe that those proofs in A-optimal objective ultimately rely on the ability to, given disjoint $I, J \subseteq [n]$ (or in the other case, $[m]$), efficiently compute

$$\sum_{S\in\mathcal{U},I\subseteq S,J\cap S=\phi}\lambda^S\sum_{|R|=d,R\subseteq S}\det(V_RV_R^\top) \text{ and } \sum_{S\in\mathcal{U},I\subseteq S,J\cap S=\phi}\lambda^S\sum_{|T|=d-1,T\subseteq S}\det(V_T^\top V_T)$$

(or in the other case, replace V with W and $\lambda^S = 1$ for all S). The proofs for generalized ratio objective follow the same line as those proofs of four algorithms, except that we instead need to efficiently compute

$$\sum_{S\in\mathcal{U},I\subseteq S,J\cap S=\phi} \lambda^S \sum_{|T|=l,R\subseteq S} \det(V_T^\top V_T) \text{ and } \sum_{S\in\mathcal{U},I\subseteq S,J\cap S=\phi} \lambda^S \sum_{|T'|=l',T'\subseteq S} \det(V_{T'}^\top V_{T'})$$

(note the change of R, T of size $d, d-1$ to T, T' of size l, l' respectively). But the computations can indeed be done efficiently by using different $d_0 = l', l$ instead of $d_0 = d-1, d$ when applying Lemmas 3.6.1, 3.6.6, and 3.6.7 in the proofs and then following a similar calculation. The proofs for running times are identical. □

3.7 Integrality Gaps

3.7.1 Integrality Gap for E-Optimality

Here we consider another objective for optimal design of experiments, the E-optimal design objective, and show that our results in the asymptotic regime do not extend to it. Once again, the input is a set of vectors $v_1, \ldots, v_n \in \mathbb{R}^d$, and our goal is to select a set $S \subseteq [n]$ of size k, but this time we minimize the objective $\left\|(\sum_{i\in S} v_i v_i^\top)^{-1}\right\|$, where $\|\cdot\|$ is the operator norm, i.e. the largest singular value. By taking the inverse of the objective, this is equivalent to maximizing $\lambda_1(\sum_{i\in S} v_i v_i^\top)$, where $\lambda_i(M)$ denotes the ith smallest eigenvalue of M. This problem also has a natural convex relaxation, analogous to the one we use for the A objective:

$$\max \lambda_1\left(\sum_{i=1}^n x_i v_i v_i^\top\right) \tag{3.40}$$

s.t.

$$\sum_{i=1}^n x_i = k \tag{3.41}$$

$$0 \le x_i \le 1 \quad \forall i \in [n] \tag{3.42}$$

We prove the following integrality gap result for (3.40)–(3.42).

Theorem 3.7.1. *There exists a constant $c > 0$ such that the following holds. For any small enough $\epsilon > 0$, and all integers $d \geq d_0(\epsilon)$, if $k < \frac{cd}{\epsilon^2}$, then there exists an instance $v_1, \dots v_n \in \mathbb{R}^d$ of the E-optimal design problem, for which the value* CP *of* (3.40)–(3.42) *satisfies*

$$\mathsf{CP} > (1+\epsilon)\text{OPT} = (1+\epsilon) \max_{S \subseteq [n]:|S|=k} \lambda_1 \left(\sum_{i \in S} v_i v_i^\top \right)$$

Recall that for the A-objective we achieve a $(1+\epsilon)$-approximation for $k = \Omega(\frac{d}{\epsilon} + \frac{\log(1/\epsilon)}{\epsilon^2})$. Theorem 3.7.1 shows that such a result is impossible for the E-objective, for which the results in [ALSW17b] cannot be improved.

Our integrality gap instance comes from a natural connection to spectral graph theory. Let us first describe the instance for any given d. We first define $n = \binom{d+1}{2}$ vectors in $\mathbb{R}^{d+1}$, one for each unordered pair $(i,j) \in \binom{[d+1]}{2}$. The vector corresponding to (i,j), $i < j$, is u_{ij} and has value 1 in the i-th coordinate, -1 in the j-th coordinate, and 0 everywhere else. In other words, the u_{ij} vectors are the columns of the vertex by edge incidence matrix U of the complete graph K_{d+1}, and $UU^\top = (d+1)I_{d+1} - J_{d+1}$ is the (unnormalized) Laplacian of K_{d+1}. (We use I_m for the $m \times m$ identity matrix, and J_m for the $m \times m$ all-ones matrix.) All the u_{ij} are orthogonal to the all-ones vector 1; we define our instance by writing u_{ij} in an orthonormal basis of this subspace: pick any orthonormal basis $b_1, \dots, b_d$ of the subspace of $\mathbb{R}^{d+1}$ orthogonal to 1, and define $v_{ij} = B^\top u_{ij}$ for $B = (b_i)_{i=1}^d$. Thus

$$M = \sum_{i=1}^{d+1} \sum_{j=i+1}^{d+1} v_{ij} v_{ij}^\top = (d+1)I_d.$$

We consider the fractional solution $x = \frac{k}{\binom{d+1}{2}} 1$, i.e. each coordinate of x is $k/\binom{d+1}{2}$. Then $M(x) = \sum_{i=1}^{d+1} \sum_{j=i+1}^{d+1} x_{ij} v_{ij} v_{ij}^\top = \frac{2k}{d} I_d$, and the objective value of the solution is $\frac{2k}{d}$.

Consider now any integral solution $S \subseteq \binom{[d+1]}{2}$ of the E-optimal design problem. We can treat S as the edges of a graph $G = ([d+1], S)$, and the Laplacian L_G of this graph is $L_G = \sum_{(i,j) \in S} u_{ij} u_{ij}^\top$. If the objective value of S is at most $(1+\epsilon)\mathsf{CP}$, then the smallest eigenvalue of

$M(S) = \sum_{(i,j)\in S} v_{ij}v_{ij}^\top$ is at least $\frac{2k}{d(1+\epsilon)} \geq (1-\epsilon)\frac{2k}{d}$. Since $M(S) = B^\top L_G B$, this means that the second smallest eigenvalue of L_G is at least $(1-\epsilon)\frac{2k}{d}$. The average degree Δ of G is $\frac{2k}{d+1}$. So, we have a graph G on $d+1$ vertices with average degree Δ for which the second smallest eigenvalue of its Laplacian is at least $(1-\epsilon)(1-\frac{1}{d+1})\Delta \geq (1-2\epsilon)\Delta$, where the inequality holds for d large enough. The classical Alon-Boppana bound ([Alo86, Nil91]) shows that, up to lower order terms, the second smallest eigenvalue of the Laplacian of a Δ-regular graph is at most $\Delta - 2\sqrt{\Delta}$. If our graph G were regular, this would imply that $\frac{2k}{d+1} = \Delta \geq \frac{1}{\epsilon^2}$. In order to prove Theorem 3.7.1, we extend the Alon-Boppana bound to not necessarily regular graphs, but with worse constants. There is an extensive body of work on extending the Alon-Boppana bound to non-regular graphs: see the recent preprint [ST17] for an overview of prior work on this subject. However, most of the work focuses either on the normalized Laplacian or the adjacency matrix of G, and we were unable to find the statement below in the literature.

Theorem 3.7.2. *Let $G = (V, E)$ be a graph with average degree $\Delta = \frac{2|E|}{|V|}$, and let L_G be its unnormalized Laplacian matrix. Then, as long as Δ is large enough, and $|V|$ is large enough with respect to Δ,*

$$\lambda_2(L_G) \leq \Delta - c\sqrt{\Delta},$$

where $\lambda_2(L_G)$ is the second smallest eigenvalue of L_G, and $c > 0$ is an absolute constant.

Proof. By the variational characterization of eigenvalues, we need to find a unit vector x, orthogonal to 1, such that $x^\top L_G x \leq \Delta - c\sqrt{\Delta}$. Our goal is to use a vector x similar to the one used in the lower bound on the number of edges of a spectral sparsifier in [BSS12b]. However, to apply this strategy we need to make sure that G has a low degree vertex most of whose neighbors have low degree. This requires most of the work in the proof.

So that we don't have to worry about making our "test vector" orthogonal to 1, observe that

$$\lambda_2(L_G) = \min_{x\in\mathbb{R}^V} \frac{x^\top L_G x}{x^\top x - (1^\top x)^2/|V|}. \tag{3.43}$$

Indeed, the denominator equals $y^\top y$ for the projection y of x orthogonal to 1, and the numerator is equal to $y^\top L_G y$. Here, and in the remainder of the proof, we work in $\mathbb{R}^V$, the space of $|V|$-dimensional real vectors indexed by V, and think of L_G as being indexed by V as well.

Observe that if G has a vertex u of degree $\Delta(u)$ at most $\Delta - \frac{1}{10}\sqrt{\Delta}$, we are done. In that case we can pick $x \in \mathbb{R}^V$ such that $x_u = 1$ and $x_v = 0$ for all $v \neq u$. Then

$$\frac{x^\top L_G x}{x^\top x - (1^\top x)^2/n} = \frac{\sum_{(u,v)\in E} (x_u - x_v)^2}{1 - \frac{1}{|V|}} \leq \frac{\Delta - \frac{1}{10}\sqrt{\Delta}}{1 - \frac{1}{|V|}},$$

which, by (3.43), implies the theorem for all large enough $|V|$. Therefore, for the rest of the proof we will assume that $\Delta(u) \geq \Delta - \frac{1}{10}\sqrt{\Delta}$ for all $u \in V$.

Define $T = \{u \in V : \Delta(u) \geq \Delta + \frac{1}{2}\sqrt{\Delta}\}$ to be the set of large degree vertices, and let $S = V \setminus T$. Observe that

$$\begin{aligned} |V|\Delta &\geq |T|\big(\Delta + \frac{1}{2}\sqrt{\Delta}\big) + |S|\big(\delta - \frac{1}{10}\sqrt{\Delta}\big) \\ &= |V|\Delta + \big(\frac{1}{2}|T| - \frac{1}{10}|S|\big)\sqrt{\Delta}. \end{aligned}$$

Therefore, $|S| \geq 5|T|$, and, since T and S partition V, we have $|S| \geq \frac{5}{6}|V|$.

Define

$$\alpha = \min\left\{\frac{|\{v \sim u : v \in T\}|}{\Delta - \frac{1}{10}\sqrt{\Delta}} : u \in S\right\},$$

where $v \sim u$ means that v is a neighbor of u. We need to find a vertex in S such that only a small fraction of its neighbors are in T, i.e. we need an upper bound on α. To show such an upper bound, let us define $E(S,T)$ to be the set of edges between S and T; then

$$\frac{1}{2}\Delta|V| = |E| \geq |E(S,T)| \geq |S|\alpha\left(\Delta - \frac{1}{10}\sqrt{\Delta}\right) \geq \frac{5}{6}|V|\alpha\Delta\left(1 - \frac{1}{10\sqrt{\Delta}}\right).$$

Therefore, $\alpha \leq \frac{3}{5}\big(1 - \frac{1}{10\sqrt{\Delta}}\big)^{-1}$.

Let $u \in S$ be a vertex with at most $\alpha\Delta - \frac{\alpha}{10}\sqrt{\Delta}$ neighbors in T, and let $\delta = |\{v \sim u : v \in S\}|$. By the choice of u,

$$\delta \geq \Delta(u) - \alpha\Delta + \frac{\alpha}{10}\sqrt{\Delta} \geq (1-\alpha)\Delta\left(1 - \frac{1}{10\sqrt{\Delta}}\right).$$

Assume that Δ is large enough so that $\left(1 - \frac{1}{10\sqrt{\Delta}}\right) \geq \frac{16}{25}$. Then, $\delta \geq \frac{16}{25}(1-\alpha)\Delta$.

We are now ready to define our test vector x and complete the proof. Let $x_u = 1$, $x_v = \frac{1}{\sqrt{\delta}}$ for any neighbor v of u which is in S, and $x_w = 0$ for any w which is in T or is not a neighbor of u. We calculate

$$\begin{aligned} x^\top L_G x &= |\{v \sim u : v \in S\}| \left(1 - \frac{1}{\sqrt{\delta}}\right)^2 + |\{v \sim u : v \in T\}| + \sum_{v \sim u, v \in S} \sum_{w \sim v, w \neq u} \frac{1}{\delta} \\ &\leq \delta \left(1 - \frac{1}{\sqrt{\delta}}\right)^2 + \Delta(u) - \delta + \Delta + \frac{1}{2}\sqrt{\Delta} - 1, \end{aligned}$$

where we used the fact for any $v \in S$, $\Delta(v) \leq \Delta + \frac{1}{2}\sqrt{\Delta}$ by definition of S. The right hand side simplifies to

$$\Delta(u) - 2\sqrt{\delta} + \Delta + \frac{1}{2}\sqrt{\Delta} \leq 2\Delta - \left(\frac{8}{5}\sqrt{(1-\alpha)} - \frac{1}{2}\right)\sqrt{\Delta}.$$

Since $\alpha \leq \frac{3}{5}\left(1 - \frac{1}{10\sqrt{\Delta}}\right)^{-1}$, $\frac{8}{5}\sqrt{(1-\alpha)} - \frac{1}{2} \geq \frac{1}{2}$ for all large enough Δ, and by (3.43), we have

$$\lambda_2(G) \leq \frac{x^\top L_G x}{x^\top x - (1^\top x)^2} \leq \frac{2\Delta - \frac{1}{2}\sqrt{\Delta}}{2\left(1 - \frac{1+\sqrt{\Delta}}{2|V|}\right)} = \left(\Delta - \frac{1}{4}\sqrt{\Delta}\right)\left(1 - \frac{1+\sqrt{\Delta}}{2|V|}\right)^{-1}.$$

The theorem now follows as long as $|V| \geq C\Delta$ for a sufficiently large constant C. □

To finish the proof of Theorem 3.7.1, recall that the existence of a $(1+\epsilon)$-approximate solution S to our instance implies that, for all large enough d, the graph $G = ([d+1], S)$ with average degree $\Delta = \frac{2k}{d+1}$ satisfies $\lambda_2(L_G) \geq (1-2\epsilon)\Delta$. By Theorem 3.7.2, $\lambda_2(L_G) \leq \Delta - c\sqrt{\Delta}$ for large enough d with respect to Δ. We have $\Delta \geq \frac{c^2}{4\epsilon^2}$, and re-arranging the terms proves the theorem.

Note that the proof of Theorem 3.7.2 does not require the graph G to be simple, i.e. parallel edges are allowed. This means that the integrality gap in Theorem 3.7.1 holds for the E-optimal design problem with repetitions as well.

3.7.2 Integrality Gap for A-optimality

Theorem 3.7.3. *There exists an instance of the A-optimal design $v_1, \ldots, v_n$ such that*

$$\text{OPT} \geq (\frac{k}{k-d+1} - \delta)\mathsf{CP}$$

for any $\delta > 0$.

This implies that the gap is at least $\frac{k}{k-d+1}$. The theorem statement applies to both with and without repetitions.

Proof. The example $v_1, \ldots, v_n$ will be the same for the problem either with or without repetitions. Pick v_i to be paralleled to axis i for each $1 \leq i \leq d$. We will set the rest $v_i, i > d$ to be at least k copies of each of these v_i for $i \leq d$, as we can pick n as big as needed. Hence, we may assume that we are allowed to pick only $v_i, i \leq d$, but with repetition.

Choose $v_i = N \cdot e_i$ for each $i = 1, \ldots, d-1$, and $v_d = e_d$. As $N \to \infty$, the fractional optimal solution (can be calculated by Lagrange's multiplier technique) is $y^* = (\delta_0, \delta_0, \ldots, \delta_0, k - (d-1)\delta_0)$ for a very small $\delta_0 = \frac{k}{\sqrt{N}+d-1}$. The optimal integral solution is $x^* = (1, 1, \ldots, 1, k-d+1)$. We have $\mathsf{CP} = \frac{d-1}{\delta_0 N} + \frac{1}{k-(d-1)\delta_0} \to \frac{1}{k}$, and OPT $= \frac{d-1}{N} + \frac{1}{k-d+1} \to \frac{1}{k-d+1}$. Hence,

$$\frac{\text{OPT}}{\mathsf{CP}} \to \frac{k}{k-d+1}.$$

□

3.8 Hardness of Approximation

In this section we show that the A-optimal design problem is NP-hard to approximate within a fixed constant when $k = d$. To the best of our knowledge, no hardness results for this problem were previously known. Our reduction is inspired by the hardness of approximation for D-optimal design proved in [SEFM15]. The hard problem we reduce from is an approximation version of Partition into Triangles.

Before we prove our main hardness result, Theorem 3.1.8, we describe the class of instances we consider, and prove some basic properties. Given a graph $G = ([d], E)$, we define a vector v_e for each edge $e = (i, j)$ so that its i-th and j-th coordinates are equal to 1, and all its other coordinates are equal to 0. Then the matrix $V = (v_e)_{e \in E}$ is the undirected vertex by edge incidence matrix of G. The main technical lemma needed for our reduction follows.

Lemma 3.8.1. *Let V be the vertex by edge incidence matrix of a graph $G = ([d], E)$, as described above. Let $S \subseteq E$ be a set of d edges of G so that the submatrix V_S is invertible. Then each connected component of the subgraph $H = ([d], S)$ is the disjoint union of a spanning tree and an edge. Moreover, if t of the connected components of H are triangles, then*

- *for $t = \frac{d}{3}$, $\operatorname{tr}((V_S V_S^\top)^{-1}) = \frac{3d}{4}$;*
- *for any t, $\operatorname{tr}((V_S V_S^\top)^{-1}) \geq d - \frac{3t}{4}$.*

Proof. Let $H_1, \dots, H_c$ be the connected components of H. First we claim that the invertibility of V_S implies that none of the H_ℓ is bipartite. Indeed, if some H_ℓ were bipartite, with bipartition $L \cup R$, then the nonzero vector x defined by

$$x_i = \begin{cases} 1 & i \in L \\ -1 & i \in R \\ 0 & \text{otherwise,} \end{cases}$$

is in the kernel of V_S. In particular, each H_ℓ must have at least as many edges as vertices. Because the number of edges of H equals the number of vertices, it follows that *every* connected component H_ℓ must have exactly as many edges as vertices, too. In particular, this means that every H_ℓ is the disjoint union of a spanning tree and an edge, and the edge creates an odd-length cycle.

Let us explicitly describe the inverse V_S^{-1}. For each $e \in S$ we need to give a vector $u_e \in \mathbb{R}^d$ so that $u_e^\top v_e = 1$ and $u_e^\top v_f = 0$ for every $f \in S$, $f \neq e$. Then $U^\top = V_S^{-1}$, where $U = (u_e)_{e \in S}$ is the matrix whose columns are the u_e vectors. Let H_ℓ be, as above, one of the connected components of H. We will define the vectors u_e for all edges e in H_ℓ; the vectors for edges in the other connected components are defined analogously. Let C_ℓ be the unique cycle of H_ℓ. Recall that C_ℓ must be an odd cycle. For any $e = (i, j)$ in C_ℓ, we set the i-th and the j-th coordinate of u_e to $\frac{1}{2}$. Let T be the spanning tree of H_ℓ derived from removing the edge e. We set the coordinates of u_e corresponding to vertices of H_ℓ other than i and j to either $-\frac{1}{2}$ or $+\frac{1}{2}$, so that the vertices of any edge of T receive values with opposite signs. This can be done by setting the coordinate of u_e corresponding to vertex k in H_ℓ to $\frac{1}{2}(-1)^{\delta_T(i,k)}$, where $\delta_T(i, k)$ is the distance in T between i and k. Because C_ℓ is an odd cycle, $\delta_T(i, j)$ is even, and this assignment is consistent with the values we already determined for i and j. Finally, the coordinates of u_e which do not correspond to vertices of H_ℓ are set to 0. See Figure 3.1 for an example. It is easy to verify that $u_e^\top v_e = 1$ and $u_e^\top v_f = 0$ for any edge $f \neq e$. Notice that $\|u_e\|_2^2 = \frac{d_\ell}{4}$, where d_ℓ is the number of vertices (and also the number of edges) of H_ℓ.

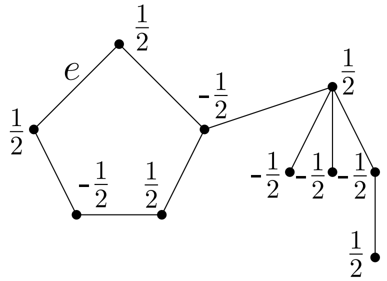

Figure 3.1: The values of the coordinates of u_e for $e \in C_\ell$.

It remains to describe u_e when $e = (i, j) \notin C_\ell$. Let T be the tree derived from H_ℓ by contracting

C_ℓ to a vertex r, and set r as the root of T. Without loss of generality, assume that j is the endpoint of e which is further from r in T. We set the j-th coordinate of u_e equal to 1. We set the coordinates of u_e corresponding to vertices in the subtree of T below j to either -1 or $+1$ so that the signs alternate down each path from j to a leaf of T below j. This can be achieved by setting the coordinate of u_e corresponding to vertex k to $(-1)^{\delta_T(j,k)}$, where $\delta_T(j,k)$ is the distance between j and k in T. All other coordinates of u_e are set equal to 0. See Figure 3.2 for an example. Notice that $\|u_e\|_2^2 \geq 1$ (and in fact equals the number of nodes in the subtree of T below the node j).

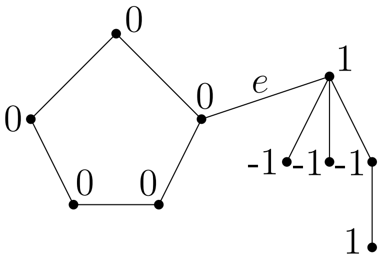

Figure 3.2: The values of the coordinates of u_e for $e \notin C_\ell$.

We are now ready to finish the proof. Clearly if $[d]$ can be partitioned into $t = \frac{d}{3}$ disjoint triangles, and the union of their edges is S, then

$$\operatorname{tr}((V_S V_S^\top)^{-1}) = \operatorname{tr}(UU^\top) = \sum_{e \in S} \|u_e\|_2^2 = \frac{3|S|}{4} = \frac{3d}{4}.$$

In the general case, we have

$$\begin{aligned}\operatorname{tr}((V_S V_S^\top)^{-1}) = tr(UU^\top) &= \sum_{e \in S} \|u_e\|_2^2 \\ &\geq \sum_{\ell=1}^{c} \frac{|C_\ell| \cdot d_\ell}{4} + d_\ell - |C_\ell| \\ &\geq \frac{9t}{4} + d - 3t = d - \frac{3t}{4},\end{aligned}$$

where $|C_\ell|$ is the length of C_ℓ, and d_ℓ is the number of edges (and also the number of vertices) in H_ℓ. The final inequality follows because any connected component H_ℓ which is not a triangle contributes at least d_ℓ to the sum. □

Recall that in the Partition into Triangles problem we are given a graph $G = (W, E)$, and need to decide if W can be partitioned into $\frac{|W|}{3}$ vertex-disjoint triangles. This problem is NP-complete ([GJ79] present a proof in Chapter 3 and cite personal communication with Schaeffer), and this, together with Lemma 3.8.1, suffice to show that the A-optimal design problem is NP-hard when $k = d$. To prove hardness of approximation, we prove hardness of a gap version of Partition into Triangles. In fact, we just observe that the reduction from 3-Dimensional Matching to Partition into Triangles in [GJ79] and known hardness of approximation of 3-Dimensional Matching give the result we need.

Lemma 3.8.2. *Given a graph $G = (W, E)$, it is* NP*-hard to distinguish the two cases:*

1. *W can be partitioned into $\frac{|W|}{3}$ vertex-disjoint triangles;*
2. *every set of vertex-disjoint triangles in G has cardinality at most $\alpha\frac{|W|}{3}$,*

where $\alpha \in (0, 1)$ is an absolute constant.

To prove Lemma 3.8.2 we use a theorem of Petrank.

Theorem 3.8.3 ([Pet94])**.** *Given a collection of triples $F \subseteq X \times Y \times Z$, where X, Y, and Z are three disjoint sets of size m each, and each element of $X \cup Y \cup Z$ appears in at most 3 triples of F, it is* NP*-hard to distinguish the two cases*

1. *there is a set of disjoint triples $M \subseteq F$ of cardinality m;*
2. *every set of disjoint triples $M \subseteq F$ has cardinality at most βm,*

where $\beta \in (0, 1)$ is an absolute constant.

We note that Petrank gives a slightly different version of the problem, in which the set M is allowed to have intersecting triples, and the goal is to maximize the number of elements $X \cup Y \cup Z$

that are covered exactly once. Petrank shows that it is hard to distinguish between the cases when every element is covered exactly once, and the case when at most $3\beta m$ elements are covered exactly once. It is immediate that this also implies Theorem 3.8.3.

Proof of Lemma 3.8.2: We will show that the reduction in [GJ79] from 3-Dimensional Matching to Partition into Triangles is approximation preserving. This follows in a straightforward way from the argument in [GJ79], but we repeat the reduction and its analysis for the sake of completeness.

Given $F \subseteq X \cup Y \cup Z$ such that each element of $X \cup Y \cup Z$ appears in at most 3 tripes of F, we construct a graph $G = (W, E)$ on the vertices $X \cup Y \cup Z$ and $9|F|$ additional vertices: $a_{f1}, \ldots a_{f9}$ for each $f \in F$. For each triple $f \in F$, we include in E the edges E_f shown in Figure 3.3. Note that the subgraphs spanned by the sets E_f, E_g for two different triples f and g are edge-disjoint, and the only vertices they share are in $X \cup Y \cup Z$.

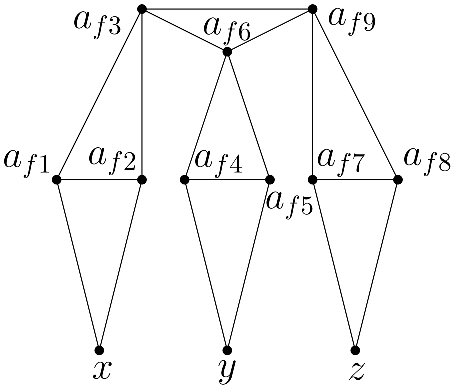

Figure 3.3: The subgraph with edges E_f for the triple $f = \{x, y, z\}$. (Adapted from [GJ79])

First we show that if F has a matching M covering all elements of $X \cup Y \cup Z$, then G can be partitioned into vertex-disjoint triangles. Indeed, for each $f = \{x, y, z\} \in M$ we can take the triangles $\{x, a_{f1}, a_{f2}\}$, $\{y, a_{f4}, a_{f5}\}$, $\{z, a_{f7}, a_{f8}\}$, and $\{a_{f3}, a_{f6}, a_{f9}\}$. For each $f \notin M$ we can take the triangles $\{a_{f1}, a_{f2}, a_{f3}\}$, $\{a_{f4}, a_{f5}, a_{f6}\}$, and $\{a_{f7}, a_{f8}, a_{f9}\}$.

In the other direction, assume there exists a set T of at least $\alpha \frac{|W|}{3}$ vertex disjoint triangles in G, for a value of α to be chosen shortly. We need to show that F contains a matching of at least βm triples. To this end, we construct a set M which contains all triples f, for each E_f which contains at least 4 triangles of T. Notice that the only way to pick three vertex disjoint triangles from E_f is to include the lower three triangles (see Figure), so any two triples f and g in M must be disjoint. The cardinality of T is at most $4|M| + 3(|F| - |M|) = |M| + 3|F|$. Therefore,

$$|M| + 3|F| \geq \alpha \frac{|W|}{3} = \alpha(m + 3|F|),$$

and we have $|M| \geq \alpha m - (1-\alpha)3|F| \geq (10\alpha - 9)m$, where we used the fact that $|F| \leq 3m$ because each element of X appears in at most 3 triples of F. Then, if $\alpha \geq \frac{9+\beta}{10}$ we have $|M| \geq \beta m$. This finishes the proof of the lemma. □

We now have everything in place to finish the proof of our main hardness result.

Proof of Theorem 3.1.8: We use a reduction from (the gap version of) Partition into Triangles to the A-optimal design problem. In fact the reduction was already described in the beginning of the section: given a graph $G = ([d], E)$, it outputs the columns v_e of the vertex by edge incidence matrix V of G.

Consider the case in which the vertices of G can be partitioned into vertex-disjoint triangles. Let S be the union of the edges of the triangles. Then, by Lemma 3.8.1, $\operatorname{tr}((V_S V_S^\top)^{-1}) = \frac{3d}{4}$.

Next, consider the case in which every set of vertex-disjoint triangles in G has cardinality at most $\alpha \frac{d}{3}$. Let S be any set of d edges in E such that V_S is invertible. The subgraph $H = ([d], S)$ of G can have at most $\alpha \frac{d}{3}$ connected components that are triangles, because any two triangles in distinct connected components are necessarily vertex-disjoint. Therefore, by Lemma 3.8.1, $\operatorname{tr}((V_S V_S^\top)^{-1}) \geq \frac{(4-\alpha)d}{4}$.

It follows that a c-approximation algorithm for the A-optimal design problem, for any $c < \frac{4-\alpha}{3}$, can be used to distinguish between the two cases of Lemma 3.8.2, and, therefore, the A-optimal design problem is NP-hard to c-approximate. □

3.9 Regularized Proportional Volume Sampling for Ridge Regression

In this section, we consider the problem of optimal design with a regularizer called *ridge regression*, and extend the sampling algorithms for *A*-optimal design to ridge regression. We first start with the background and motivation of ridge regression.

3.9.1 Background

Notations

We recall notations used throughout this thesis. Let $V = [v_1 \dots v_n]$ be the d-by-n matrix of vectors $v_i \in \mathbb{R}^d$. These v_i's are also called datapoints. We want to select a subset $S \subseteq [n]$ of size k so that learning the model with label on S is as efficient as possible. Let $V_S = [v_i]_{i \in S}$ be a matrix with columns $v_i, i \in S$. Let y be the label column vector, and y_S is the $k \times 1$ column vector $(y_i)_{i \in S}$. We denote X as the datapoints we want to predict, which is most cases is the same as V.

Linear Model Assumption

In optimal design throughout the thesis, we assume that $y_i = x_i^\top w^* + \eta_i$ where η_i are independent Gaussian noise with mean zero and same variance. In this section, we note that we may also assume η is a random Gaussian vector $N\left(0, \mathrm{Cov}\left(\eta\right)\right)$ with $\mathrm{Cov}\left(\eta\right) \preceq \sigma^2 I$. Under this assumption, the errors to be presented in this section is upper bounded by in the setting where $\eta \sim N\left(0, \sigma^2 I\right)$. Hence, for simplicity we assume $\eta \sim N\left(0, \sigma^2 I\right)$ as earlier.

After obtaining labels y_S, we are interested in fitting linear model $\hat{w}_S$ by minimizing square loss with a regularizer with parameter λ:

$$\hat{w}_S = \operatorname*{argmin}_{w \in \mathbb{R}^d} \left\{ \|y_S - V_S^\top w\|_2^2 + \lambda \|w\|_2^2 \right\} \tag{3.44}$$

This problem is called *ridge regression*, and when $\lambda = 0$, the problem reverts to linear regression.

Table 3.4: Distributions of model and prediction errors in ridge regression

Errors	$\hat{w}_S - w^*$	$X^\top \left(\hat{w}_S - w^*\right)$
$\lambda = 0$	$= N\left(0, \sigma^2 \left(V_S V_S^\top\right)^{-1}\right)$	$= N\left(0, \sigma^2 X^\top \left(V_S V_S^\top\right)^{-1} X\right)$
$\lambda \geq 0$	$= N(-\lambda Z_S(\lambda)^{-1} w^*,$ $\sigma^2 \left[Z_S(\lambda)^{-1} - \lambda Z_S(\lambda)^{-2}\right])$	$= N(-\lambda X^\top Z_S(\lambda)^{-1} w^*,$ $\sigma^2 X^\top \left[Z_S(\lambda)^{-1} - \lambda Z_S(\lambda)^{-2}\right] X)$

It is also known that the above ridge regression is equivalent to linear regression under Gaussian prior assumption. Ridge regression with $\lambda > 0$ increases the stability the linear regression against the outliar, and forces the optimization problem to have unique solution even when datapoints in V do not span full-rank d.

Model Error and Prediction Error

In order to motivate a good objective for subset selection problem, we calculate the model error $\hat{w}_S - w^*$ and prediction error $X^\top \left(\hat{w}_S - w^*\right)$ when the predictor is used to predict datapoints X. In many applications, the matric of error concerns X to be the same as V. These errors are random with distributions summarized in Table 3.4.

The calculations used to obtain distribution in Table 3.4 is similar in each of four cases. Here we will compute only one example, $X^\top \left(\hat{w}_S - w^*\right)$. This example is the most complicated one of the four, and enough to guide the reader to obtain other three.

Denote $Z_S(\lambda) := \left(V_S V_S^\top + \lambda I\right)$. We first state a simple claim that will help in this calculation.

Claim 2. *For a fixed matrix A and a random vector Z, we have* $\operatorname{Cov}\left(AZ\right) = A \operatorname{Cov}\left(Z\right) A^\top$.

Proof. Denote $m = \mathbb{E}[Z]$, the mean vector of Z. Then, the mean of AZ is Am. We now have

$$\begin{aligned}
\mathrm{Cov}\,(AZ) &= \mathbb{E}\big[(AZ - Am)(AZ - Am)^\top\big] \\
&= \mathbb{E}\big[A(Z - m)(Z - m)^\top A^\top\big] \\
&= A\mathbb{E}\big[(Z - m)(Z - m)^\top\big]A^\top \\
&= A\,\mathrm{Cov}\,(Z)\,A^\top
\end{aligned}$$

□

We now show how to obtain the distribution of $X^\top\,(\hat{w}_S - w^*)$.

Claim 3. *We have*

$$X^\top\,(\hat{w}_S - w^*) = N(-\lambda X^\top Z_S(\lambda)^{-1}w^*, \sigma^2 X^\top\left[Z_S(\lambda)^{-1} - \lambda Z_S(\lambda)^{-2}\right]X)$$

Proof. We split calculations into the following steps.

1. Find closed-form solution of $\hat{w}_S$ by taking the gradient:

$$\hat{w}_S = Z_S(\lambda)^{-1}V_S y_S \tag{3.45}$$

2. Substituting y_i from the linear model assumption. This finishes obtaining the distribution of model error.

$$\begin{aligned}
\hat{w}_S - w^* &= Z_S(\lambda)^{-1}V_S y_S - w^* \\
&= Z_S(\lambda)^{-1}V_S\left(V_S^\top w^* + \eta_S\right) - w^* \\
&= Z_S(\lambda)^{-1}\left[Z_S(\lambda)w^* - (\lambda I)w^* + V_S\eta_S\right] - w^* \\
&= -\lambda Z_S(\lambda)^{-1}w^* + Z_S(\lambda)^{-1}V_S\eta_S
\end{aligned}$$

Table 3.5: Expected square loss of model and prediction errors in ridge regression

Errors	$\underset{\eta_S}{\mathbb{E}}\left[\|\hat{w}_S - w^*\|_2^2\right]$	$\underset{\eta_S}{\mathbb{E}}\left[\|X^\top(\hat{w}_S - w^*)\|_2^2\right]$
$\lambda = 0$	$= \sigma^2 \operatorname{tr} V_S V_S^{\top -1}$	$= \sigma^2 \operatorname{tr} X^\top \left(V_S V_S^\top\right)^{-1} X$
$\lambda \geq 0$	$= \sigma^2 \operatorname{tr} Z_S(\lambda)^{-1}$ $-\lambda \left\langle Z_S(\lambda)^{-2}, \sigma^2 I - \lambda w^* w^{*\top} \right\rangle$	$= \sigma^2 \operatorname{tr} X^\top Z_S(\lambda)^{-1} X$ $-\lambda \left\langle Z_S(\lambda)^{-1} X X^\top Z_S(\lambda)^{-1}, \sigma^2 I - \lambda w^* w^{*\top} \right\rangle$

3. To obtain prediction error, we simply left multiply by the data matrix:

$$X^\top (\hat{w}_S - w^*) = -\lambda X^\top Z_S(\lambda)^{-1} w^* + X^\top Z_S(\lambda)^{-1} V_S \eta_S$$

4. Linear transformation of random Gaussian vectors is Gaussian, so we use the claim above to get that the mean of prediction error is

$$\mu_{X^\top(\hat{w}_S - w^*)} = -\lambda X^\top Z_S(\lambda)^{-1} w^* \tag{3.46}$$

and the covariance is

$$\begin{aligned}
\operatorname{Cov}\left(X^\top (\hat{w}_S - w^*)\right) &= X^\top Z_S(\lambda)^{-1} V_S \operatorname{Cov}(\eta_S) \left(X^\top Z_S(\lambda)^{-1} V_S\right)^\top \\
&= X^\top Z_S(\lambda)^{-1} V_S \operatorname{Cov}(\eta_S)\, V_S^\top Z_S(\lambda)^{-1} X \\
&\preceq \sigma^2 X^\top Z_S(\lambda)^{-1} V_S V_S^\top Z_S(\lambda)^{-1} X \\
&= \sigma^2 X^\top \left[Z_S(\lambda)^{-1} - \lambda Z_S(\lambda)^{-2}\right] X
\end{aligned}$$

□

Expected Square Loss of Ridge Regression Predictor

There are several metric to minimize the error distribution. One common metric is expected square loss under the distribution. We can find this expectation as follow(s). First, we calculate expected square-loss of model error:

$$\begin{aligned}\mathop{\mathbb{E}}_{\eta}\left[\|\hat{w}_S - w^*\|_2^2\right] &= \sum_{i=1}^{d} \mathop{\mathbb{E}}_{\eta}\left[((\hat{w}_S)_i - (w^*)_i)^2\right] \\ &= \sum_{i=1}^{d}\left(\mathop{\mathbb{E}}_{\eta}\left[(\hat{w}_S)_i - (w^*)_i\right]^2 + \operatorname{Var}\left((\hat{w}_S)_i - (w^*)_i\right)\right) \\ &= \|\mathop{\mathbb{E}}_{\eta}\left[\hat{w}_S - w^*\right]\|_2^2 + \operatorname{tr}\operatorname{Cov}\left(\hat{w}_S - w^*\right)\end{aligned}$$

where we use $\mathbb{E}\left[X^2\right] = \mathbb{E}\left[X\right]^2 + \operatorname{Var}\left(X\right)$ (bias-variance decomposition). Similarly, for prediction error,

$$\begin{aligned}\mathop{\mathbb{E}}_{\eta}\left[\|X^\top(\hat{w}_S - w^*)\|_2^2\right] &= \sum_{i=1}^{d} \mathop{\mathbb{E}}_{\eta}\left[\|(X^\top(\hat{w}_S)_i - (w^*)_i)\|_2^2\right] \\ &= \sum_{i=1}^{d}\left(\mathop{\mathbb{E}}_{\eta}\left[X^\top((\hat{w}_S)_i - (w^*)_i)\right]^2 + \operatorname{Var}\left(X^\top((\hat{w}_S)_i - (w^*)_i)\right)\right) \\ &= \|\mathop{\mathbb{E}}_{\eta}\left[X^\top(\hat{w}_S - w^*)\right]\|_2^2 + \operatorname{tr}\operatorname{Cov}\left(X^\top(\hat{w}_S - w^*)\right)\end{aligned}$$

As we know mean and variance of the model and prediction errors (Table 3.4), we can substitute those means and variances:

$$\begin{aligned}\mathop{\mathbb{E}}_{\eta_S}\left[\|\hat{w}_S - w^*\|_2^2\right] &= \|-\lambda Z_S(\lambda)^{-1}w^*\|_2^2 + \operatorname{tr}\sigma^2\left[Z_S(\lambda)^{-1} - \lambda Z_S(\lambda)^{-2}\right] \\ &= \lambda^2\left\langle Z_S(\lambda)^{-2}, w^*{w^*}^\top\right\rangle + \sigma^2\operatorname{tr}Z_S(\lambda)^{-1} - \lambda\sigma^2\operatorname{tr}Z_S(\lambda)^{-2} \\ &= \sigma^2\operatorname{tr}Z_S(\lambda)^{-1} - \lambda\left\langle Z_S(\lambda)^{-2}, \sigma^2 I - \lambda w^*{w^*}^\top\right\rangle\end{aligned}$$

The challenge is the second-order term $Z_S(\lambda)^{-2}$. One way to address is to consider only the first-order term $\operatorname{tr} Z_S(\lambda)^{-1}$. For example, [DW17a] assume that the regularization λ is sufficiently small: that $\lambda \leq \frac{\sigma^2}{\|w^*\|_2^2}$. In such case, we have $\lambda w^* {w^*}^\top \preceq \sigma^2 I$. Then,

$$\mathop{\mathbb{E}}_{\eta_S}\left[\|\hat{w}_S - w^*\|_2^2\right] \leq \sigma^2 \operatorname{tr} Z_S(\lambda)^{-1} \tag{3.47}$$

where the right-hand side now contains only the first-order term $\operatorname{tr} Z_S(\lambda)^{-1}$ which can be optimized by sampling-based algorithms. Note that it is an open question to directly bound the expected loss without any assumption on λ.

For prediction error,

$$\begin{aligned}
\mathop{\mathbb{E}}_{\eta}\left[\|X^\top(\hat{w}_S - w^*)\|_2^2\right] &= \|-\lambda X^\top Z_S(\lambda)^{-1} w^*\|_2^2 + \operatorname{tr} \sigma^2 X^\top \left[Z_S(\lambda)^{-1} - \lambda Z_S(\lambda)^{-2}\right] X \\
&= \lambda^2 \left\langle Z_S(\lambda)^{-1} X X^\top Z_S(\lambda)^{-1}, w^* {w^*}^\top \right\rangle + \sigma^2 \operatorname{tr} X^\top Z_S(\lambda)^{-1} X - \lambda \sigma^2 \operatorname{tr} X^\top Z_S(\lambda)^{-2} X \\
&= \sigma^2 \operatorname{tr} X^\top Z_S(\lambda)^{-1} X - \lambda \left\langle Z_S(\lambda)^{-1} X X^\top Z_S(\lambda)^{-1}, \sigma^2 I - \lambda w^* {w^*}^\top \right\rangle
\end{aligned}$$

Again, if we assume that $\lambda \leq \frac{\sigma^2}{\|w^*\|_2^2}$, then we have $\lambda w^* {w^*}^\top \preceq \sigma^2 I$. Then,

$$\mathop{\mathbb{E}}_{\eta}\left[\|X^\top(\hat{w}_S - w^*)\|_2^2\right] \leq \sigma^2 \operatorname{tr} X^\top Z_S(\lambda)^{-1} X \tag{3.48}$$

The bound (3.47) is the analog of the *A*-optimal design objective, and is the motivation for the ridge regression objective to be considered in the next subsection.

3.9.2 λ-Regularized A-Optimal Design and λ-Regularized Proportional Volume Sampling

In this section, we consider an approximation algorithm to the optimization problem that, given $V = [v_1 \ldots v_n] \in \mathbb{R}^{d \times n}$, integer $k \geq d$, and $\lambda \in \mathbb{R}^+$, solve

$$\min_{S \subseteq [n], |S|=k} \frac{E_{d-1} Z_S(\lambda)}{E_d Z_S(\lambda)} \tag{3.49}$$

where $Z_S(\lambda) := V_S V_S^\top + \lambda I$. Though this objective is motivated from the square loss from ridge regression (see Table 3.5), it is not the same objective. However, if we assume $\lambda \leq \frac{\sigma^2}{\|w^*\|_2^2}$, then the loss is bounded above by $\frac{E_{d-1} Z_S(\lambda)}{E_d Z_S(\lambda)}$. Due to this motivation and its similarity with objective from A-optimal design, we call problem (3.49) λ-regularized A-optimal design.

Denote $\mathcal{U}_k$ ($\mathcal{U}_{\leq k}$) the set of all subsets $S \subseteq [n]$ of size k (of size $\leq k$). Given $\lambda \geq 0, y \in \mathbb{R}^n$,$\mathcal{U} \in \{\mathcal{U}_k,\mathcal{U}_{\leq k}\}$, and μ a distribution over $\mathcal{U}$, we define the *λ-regularized proportional volume sampling with measure* μ to be the distribution μ' over $\mathcal{U}$ where $\mu'(S) \propto \mu(S) \det Z_S(\lambda)$ for all $S \in \mathcal{U}$. Given $y \in \mathbb{R}^n$, we say a distribution μ over $\mathcal{U}$ is *hard-core with parameter* z if $\mu(S) \propto z^S := \prod_{i \in S} z_i$ for all $S \in \mathcal{U}$. Denote $\|A\|_2$ the spectral norm of matrix A.

To solve λ-regularized A-optimal design, we solve the convex relaxation of the optimization problem

$$\min_{x \in \mathbb{R}^n} \frac{E_{d-1}(V(x)V(x)^\top + \lambda I)}{E_d(V(x)V(x)^\top + \lambda I)} \text{ subject to} \tag{3.50}$$

$$\sum_{i=1}^n x_i = k, \tag{3.51}$$

$$1 \geq x_i \geq 0 \tag{3.52}$$

where $V(x) := [\sqrt{x_1} v_1 \ldots \sqrt{x_n} v_n]$, to get a fractional solution $x \in \mathbb{R}^n$. Note that convexity follows from the convexity of function $\frac{E_{d-1}(M)}{E_d(M)}$ over the set of all PSD matrices $M \in \mathbb{R}^{n \times n}$. Then, we use λ-regularized proportional volume sampling with hard-core measure μ with some parameter $y \in \mathbb{R}^n$ which depends on x to sample output $S \in \mathcal{U}_{\leq k}$. The summary of the algorithm is in Algorithm 3.4. The overall goal is to show that with $k = \Omega\left(\frac{d}{\epsilon} + \frac{\log(1/\epsilon)}{\epsilon^2}\right)$, Algorithm 3.4 has $(1 + \frac{\epsilon}{\sqrt{1+\frac{\lambda}{\|V(x)V(x)\|_2}}})$-approximation guarantee to solving λ-regularized A-optimal design.

Algorithm 3.4 Solving $\min_{S\subseteq[n],|S|=k} \frac{E_{d-1}Z_S(\lambda)}{E_d Z_S(\lambda)}$ with convex relaxation and λ-regularized proportional volume sampling

```
1: Given an input $V = [v_1, \dots, v_n]$ where $v_i \in \mathbb{R}^d$, $k$ a positive integer, $\lambda \geq 0$
2: Solve to get a fractional solution $x \in \operatorname{argmin}_{x\in[0,1]^n, 1^\top x=k} \frac{E_{d-1}\left(V(x)V(x)^\top+\lambda I\right)}{E_d\left(V(x)V(x)^\top+\lambda I\right)}$
3: Let $z_i = \frac{x_i}{\beta - x_i}$ where $\beta = 1 + \frac{\epsilon}{4}\sqrt{1+\frac{\lambda}{\|V(x)V(x)\|_2}}$.
4: Sample $\mathcal{S}$ from $\mu'(S) \propto z^S \det Z_S(\lambda)$ for each $S \in \mathcal{U}_{\leq k}$
5: Output $\mathcal{S}$ (If $|\mathcal{S}| < k$, add $k - |\mathcal{S}|$ arbitrary vectors to $\mathcal{S}$ first).
```

Theorem 3.9.1. *Given $V = [v_1 \dots v_n] \in \mathbb{R}^{d\times n}$, integer $k \geq d$, and $\lambda \in \mathbb{R}^+$, Algorithm 3.4 has $(1+\epsilon)$-approximation guarantee to solving λ-regularized A-optimal design.*

We note that the approximation ratio is in fact a slightly tighter factor $1 + \frac{\epsilon}{\sqrt{1+\frac{\lambda}{\|V(x)V(x)\|_2}}}$, as will be shown later in this section. This ratio shows that the algorithm's performance improves as λ increases, and is asymptotically optimal as $\lambda \to \infty$.

The proof of Theorem 3.9.1 relies on showing that proving an approximation guarantee of a λ-regularized proportional volume sampling with measure μ reduces to showing a property on μ which we called *near-pairwise independence*. This reduction is explained in Theorem 3.9.4. We then construct μ based on fractional solution x and prove that μ has such property in Section 3.9.5. Finally, we note that our constructed μ is hardcore, and show that we can efficiently implement λ-regularized proportional volume sampling with any *hard-core* measure μ.

3.9.3 Related Work

Ridge regression or regularized regression is introduced by [HK70] to ensure a unique solution of linear regression when data matrix is singular, i.e. when labeled datapoints do not span full d

dimensions. Ridge regression has been applied to many practical problems [MS75] and is one of classical linear methods for regression in machine learning [HTF09].

λ-regularized volume sampling. [DW17a] introduced λ-regularized volume sampling and gave theoretical guarantee bound for the model error $\mathop{\mathbb{E}}_{\eta_S}[\|\hat{w}_S - w^*\|_2^2]$, which equals to $\mathrm{tr}\left(V_S V_S^\top + \lambda I\right)^{-1}$, the objective of focus in this section. We explain the similarity and difference of their guarantees here. [DW17a] showed that for $\mathrm{Cov}\,(\eta) \preceq \sigma^2 I$ and $\lambda \leq \frac{\sigma^2}{\|w^*\|^2}$,

$$\mathop{\mathbb{E}}_{S}\left[\mathrm{tr}\left(V_S V_S^\top + \lambda I\right)^{-1}\right] \leq \frac{\sigma^2 n\ \mathrm{tr}((VV^\top + \lambda I)^{-1})}{k - d_\lambda + 1} \tag{3.53}$$

where $d_\lambda = \mathrm{tr}(V^\top(VV^\top + \lambda I)^{-1}V)$ (recall that n is the number of vectors to choose from). For $\lambda = 0$, $d_\lambda = d$, and d_λ decreases as λ increases.

The bound (3.53) is different from our goal of approximation ratio in this thesis. Indeed, suppose that S^* is an optimal subset of the problem, then in expectation over the run of the our algorithm,

$$\mathop{\mathbb{E}}_{S}\left[\mathrm{tr}\left(V_S V_S^\top + \lambda I\right)^{-1}\right] \leq \left(1 + c\frac{d-1}{(k-d+1)\sqrt{1 + \frac{\lambda}{\|V(x)V(x)\|_2}}}\right)\sigma^2\,\mathrm{tr}((V_{S^*}V_{S^*}^\top + \lambda I)^{-1}) \tag{3.54}$$

for some fixed constant c (we assume d is large compared to $\frac{1}{\epsilon}$ so that $\frac{d}{\epsilon} + \frac{\log(1/\epsilon)}{\epsilon^2} = O\left(\frac{d}{\epsilon}\right)$). When $\lambda = 0$, our bound (3.54) simplifies to a bound similar to (3.53):

$$\mathop{\mathbb{E}}_{S}\left[\mathrm{tr}\left(V_S V_S^\top + \lambda I\right)^{-1}\right] \leq \frac{\sigma^2 k\ \mathrm{tr}((V_{S^*}V_{S^*} + \lambda I)^{-1})}{k - d_\lambda + 1}$$

The main difference between our guarantee and ones by [DW17a] is that ours is in comparison to the best possible subset S^*, whereas (3.53) compares the performance to labelling the whole original dataset.

3.9.4 Reduction of Approxibility to Near-Pairwise Independence

In this section, we show that an approximation guarantee of a λ-regularized proportional volume sampling with measure μ to λ-regularized A-optimal design reduces to showing a property on μ which we called *near-pairwise independence*. We first define *near-pairwise independence* of a distribution.

Definition 3.9.2. Let μ be a distribution on $\mathcal{U} \in \{\mathcal{U}_k, \mathcal{U}_{\leq k}\}$. Let $x \in \mathbb{R}^n_+$. We say μ is *(c, α)-near-pairwise independent* with respect to x if for all $T, R \subseteq [n]$ each of size at most d,

$$\frac{\Pr_{\mathcal{S}\sim\mu}[S \supseteq T]}{\Pr_{\mathcal{S}\sim\mu}[S \supseteq R]} \leq c\alpha^{|R|-|T|}\frac{x^T}{x^R} \tag{3.55}$$

We omit the phrase "with respect to x" when the context is clear. Before we prove the main result, we make some calculation which will be used later.

Lemma 3.9.3. *For any matrix PSD $X \in \mathbb{R}^{d\times d}$ and $a \in \mathbb{R}$,*

$$E_d\left(X + aI\right) = \sum_{i=0}^{d} E_i(X)a^{d-i} \tag{3.56}$$

and

$$E_{d-1}\left(X + aI\right) = \sum_{i=0}^{d-1}(d-i)E_i(X)a^{d-1-i} \tag{3.57}$$

Proof. Let λ be eigenvalues of X. Then,

$$E_d\left(X + aI\right) = \prod_{i=1}^{d}(\lambda_i + a) = \sum_{i=0}^{d} e_i(\lambda)a^{d-i} = \sum_{i=0}^{d} E_i(X)a^{d-i}$$

proving the first equality. Next, we have

$$\begin{aligned} E_{d-1}\left(X+aI\right) &= \sum_{j=1}^{d} \prod_{i\in[d],i\neq j} (\lambda_i + a) \\ &= \sum_{j=1}^{d}\sum_{i=0}^{d-1} e_i(\lambda_{-j})a^{d-1-i} = \sum_{i=0}^{d-1}\left(\sum_{j=1}^{d} e_i(\lambda_{-j})\right)a^{d-1-i} \end{aligned}$$

where λ_{-j} is λ with one element λ_j deleted. For each fixed $i\in\{0,\ldots,d-1\}$, we have

$$\sum_{j=1}^{d} e_i(\lambda_{-j}) = (d-i)e_i(\lambda) \tag{3.58}$$

by counting the number of each monomial in $e_i(\lambda)$. Noting $e_i(\lambda)=E_i(X)$ finishes the proof.

□

Now we are ready to prove the main result.

Theorem 3.9.4. *Let $x\in[0,1]^n$. Let μ be a distribution on $\mathcal{U}\in\{\mathcal{U}_k,\mathcal{U}_{\leq k}\}$ that is (c,α)-near-pairwise independent. Then the λ-regularized proportional volume sampling μ' with measure μ satisfies*

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\frac{E_{d-1}\left(Z_{\mathcal{S}}(\lambda)\right)}{E_d\left(Z_{\mathcal{S}}(\lambda)\right)}\right] \leq c\alpha\frac{E_{d-1}\left(V(x)V(x)^\top+\alpha\lambda I\right)}{E_d\left(V(x)V(x)^\top+\alpha\lambda I\right)} \tag{3.59}$$

That is, the sampling gives $c\alpha$-approximation guarantee to $\alpha\lambda$-regularized A-optimal design in expectation.

Note that by $\frac{E_{d-1}\left(V(x)V(x)^\top+\alpha\lambda I\right)}{E_d\left(V(x)V(x)^\top+\alpha\lambda I\right)} \leq \frac{E_{d-1}\left(V(x)V(x)^\top+\lambda I\right)}{E_d\left(V(x)V(x)^\top+\lambda I\right)}$, (3.59) also implies $c\alpha$-approximation guarantee to the original λ-regularized A-optimal design. However, we can exploit the gap of these two quantities to get a better approximation ratio which converges to 1 as $\lambda\to\infty$. This is done in Theorem 3.9.6.

Proof. We apply Lemma 3.9.3 to RHS of (3.59) to get

$$\begin{aligned}\frac{E_{d-1}\left(V(x)V(x)^\top+\alpha\lambda I\right)}{E_d\left(V(x)V(x)^\top+\alpha\lambda I\right)} &= \frac{\sum_{h=0}^{d-1}(d-h)E_h(V(x)V(x)^\top)(\alpha\lambda)^{d-1-h}}{\sum_{\ell=0}^{d}E_\ell(V(x)V(x)^\top)(\alpha\lambda)^{d-\ell}}\\ &= \frac{\sum_{h=0}^{d-1}\sum_{|T|=h}(d-h)(\alpha\lambda)^{d-1-h}x^T\det\left(V_T^\top V_T\right)}{\sum_{\ell=0}^{d}\sum_{|R|=\ell}(\alpha\lambda)^{d-\ell}x^R\det\left(V_R^\top V_R\right)}\end{aligned}$$

where we apply Cauchy-Binet to the last equality. Next, we apply Lemma 3.9.3 to LHS of (3.59) to get

$$\begin{aligned}\mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\frac{E_{d-1}\left(Z_{\mathcal{S}}(\lambda)\right)}{E_d\left(Z_{\mathcal{S}}(\lambda)\right)}\right] &= \frac{\sum_{S\in\mathcal{U}}\mu(S)E_d(Z_S(\lambda))\frac{E_{d-1}(Z_{\mathcal{S}}(\lambda))}{E_d(Z_{\mathcal{S}}(\lambda))}}{\sum_{S\in\mathcal{U}}\mu(S)E_dZ_S(\lambda)} = \frac{\sum_{S\in\mathcal{U}}\mu(S)E_{d-1}Z_S(\lambda)}{\sum_{S\in\mathcal{U}}\mu(S)E_dZ_S(\lambda)}\\ &= \frac{\sum_{S\in\mathcal{U}}\mu(S)\sum_{h=0}^{d-1}(d-h)E_h(V_SV_S^\top)\lambda^{d-1-h}}{\sum_{S\in\mathcal{U}}\mu(S)\sum_{\ell=0}^{d}E_\ell(V_SV_S^\top)\lambda^{d-\ell}}\\ &= \frac{\sum_{S\in\mathcal{U}}\mu(S)\sum_{h=0}^{d-1}\sum_{|T|=h,T\subseteq S}(d-h)\lambda^{d-1-h}\det\left(V_T^\top V_T\right)}{\sum_{S\in\mathcal{U}}\mu(S)\sum_{\ell=0}^{d}\sum_{|R|=\ell,R\subseteq S}\lambda^{d-\ell}\det\left(V_R^\top V_R\right)}\\ &= \frac{\sum_{h=0}^{d-1}\sum_{|T|=h}\sum_{S\in\mathcal{U},S\supseteq T}\mu(S)(d-h)\lambda^{d-1-h}\det\left(V_T^\top V_T\right)}{\sum_{\ell=0}^{d}\sum_{|R|=\ell}\sum_{S\in\mathcal{U},S\supseteq R}\mu(S)\lambda^{d-\ell}\det\left(V_R^\top V_R\right)}\\ &= \frac{\sum_{h=0}^{d-1}\sum_{|T|=h}(d-h)\lambda^{d-1-h}\det\left(V_T^\top V_T\right)\Pr_{\mathcal{S}\sim\mu}\left[\mathcal{S}\supseteq T\right]}{\sum_{\ell=0}^{d}\sum_{|R|=\ell}\lambda^{d-\ell}\det\left(V_R^\top V_R\right)\Pr_{\mathcal{S}\sim\mu}\left[\mathcal{S}\supseteq R\right]}\end{aligned}$$

Therefore, by cross-multiplying the numerator and denominator, the ratio $\frac{\mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\frac{E_{d-1}(Z_{\mathcal{S}}(\lambda))}{E_d(Z_{\mathcal{S}}(\lambda))}\right]}{\frac{E_{d-1}\left(V(x)V(x)^\top+\alpha\lambda I\right)}{E_d\left(V(x)V(x)^\top+\alpha\lambda I\right)}}$ is

$$= \frac{\sum_{h=0}^{d-1}\sum_{|T|=h}\sum_{\ell=0}^{d}\sum_{|R|=\ell}(d-h)\det\left(V_T^\top V_T\right)\det\left(V_RV_R^\top\right)\lambda^{d-1-h}(\alpha\lambda)^{d-\ell}x^R\Pr_{\mu}\left[\mathcal{S}\supseteq T\right]}{\sum_{h=0}^{d-1}\sum_{|T|=h}\sum_{\ell=0}^{d}\sum_{|R|=\ell}(d-h)\det\left(V_T^\top V_T\right)\det\left(V_RV_R^\top\right)\lambda^{d-\ell}(\alpha\lambda)^{d-1-h}x^T\Pr_{\mu}\left[\mathcal{S}\supseteq R\right]}$$

For each fixed h, T, ℓ, R, we want to upper bound $\frac{\lambda^{d-1-h}(\alpha\lambda)^{d-\ell}x^R\Pr_{\mu}[\mathcal{S}\supseteq T]}{\lambda^{d-\ell}(\alpha\lambda)^{d-1-h}x^T\Pr_{\mu}[\mathcal{S}\supseteq R]}$. By the definition of near-pairwise independence (3.55),

$$\frac{\lambda^{d-1-h}(\alpha\lambda)^{d-\ell}x^R \Pr_{\mu}\left[\mathcal{S} \supseteq T\right]}{\lambda^{d-\ell}(\alpha\lambda)^{d-1-h}x^T \Pr_{\mu}\left[\mathcal{S} \supseteq R\right]} \leq \frac{\lambda^{d-1-h}(\alpha\lambda)^{d-\ell}}{\lambda^{d-\ell}(\alpha\lambda)^{d-1-h}} c\alpha^{\ell-h} \tag{3.60}$$

$$= \alpha^{h-\ell+1} \cdot c\alpha^{\ell-h} = c\alpha \tag{3.61}$$

Therefore, the ratio $\frac{\mathbb{E}_{\mathcal{S}\sim\mu'}\left[\frac{E_{d-1}(Z_{\mathcal{S}}(\lambda))}{E_d(Z_{\mathcal{S}}(\lambda))}\right]}{\frac{E_{d-1}(V(x)V(x)^\top+\alpha\lambda I)}{E_d(V(x)V(x)^\top+\alpha\lambda I)}}$ is also bounded above by $c\alpha$. □

3.9.5 Constructing a Near-Pairwise-Independent Distribution

In this section, we want to construct a distribution μ on $\mathcal{U}_{\leq k}$ and prove its (c, α)-near-pairwise-independent property. Our proposed μ is hard-core with parameter $z \in \mathbb{R}^n$ defined by $z_i := \frac{x_i}{\beta - x_i}$ (coordinate-wise) for some $\beta \in (1, 2]$. With this choice of μ, we upper bound the ratio $\frac{\Pr_{\mathcal{S}\sim\mu}[\mathcal{S}\supseteq T]}{\Pr_{\mathcal{S}\sim\mu}[\mathcal{S}\supseteq R]}$ in terms of β. Later in Section 3.9.6, after getting an explicit approximation ratio in term of β here, we found that under the assumption $k = \Omega\left(\frac{d}{\epsilon} + \frac{\log(1/\epsilon)}{\epsilon^2}\right)$, the choice $\beta = 1 + \frac{\epsilon}{4}\sqrt{1 + \frac{\lambda}{\|V(x)V(x)\|_2}}$ gives $(1 + \frac{\epsilon}{\sqrt{1+\frac{\lambda}{\|V(x)V(x)\|_2}}})$-approximation guarantee to Algorithm 3.4.

Lemma 3.9.5. *Let $x \in [0,1]^n$ such that $\sum_{i=1}^n x_i = k$. Let μ be a distribution on $\mathcal{U}_{\leq k}$ that is hard-core with parameter $z \in \mathbb{R}^n$ defined by $z_i := \frac{x_i}{\beta - x_i}$ (coordinate-wise) for some $\beta \in (1, 2]$. Then, for all $T, R \subseteq [n]$ of size h, ℓ between 0 and d, we have*

$$\frac{\Pr_{\mathcal{S}\sim\mu}\left[\mathcal{S} \supseteq T\right]}{\Pr_{\mathcal{S}\sim\mu}\left[\mathcal{S} \supseteq R\right]} \leq \frac{\beta^{\ell-h}}{1 - \exp\left(-\frac{(\beta-1)k-\beta d)^2}{3\beta k}\right)} \cdot \frac{x^T}{x^R} \tag{3.62}$$

That is, μ is $\left(\frac{1}{1-\exp\left(-\frac{(\beta-1)k-\beta d)^2}{3\beta k}\right)}, \beta\right)$-near-pairwise independent.

Proof. Fix T, R of size $0 \leq h, \ell \leq d$. Define $\mathcal{B} \subseteq [n]$ to be the random set that includes each $i \in [n]$ independently with probability x_i/β. Let $Y_i = \mathbb{1}\left[i \in \mathcal{B}\right]$ and $Y = \sum_{i \notin R} Y_i$. Then, noting that $z_i = \frac{x_i/\beta}{1-x_i/\beta}$, we have

$$\frac{\underset{\mathcal{S}\sim\mu}{\mathsf{Pr}}\left[\mathcal{S} \supseteq T\right]}{\underset{\mathcal{S}\sim\mu}{\mathsf{Pr}}\left[\mathcal{S} \supseteq R\right]} = \frac{\mathsf{Pr}\left[\mathcal{B} \supseteq T, |\mathcal{B}| \leq k\right]}{\mathsf{Pr}\left[\mathcal{B} \supseteq R, |\mathcal{B}| \leq k\right]} \leq \frac{\mathsf{Pr}\left[\mathcal{B} \supseteq T\right]}{\mathsf{Pr}\left[\mathcal{B} \supseteq R, |\mathcal{B}| \leq k\right]}$$

$$= \beta^{\ell-h}\frac{x^T}{x^R}\frac{1}{\mathsf{Pr}\left[\sum_{i\notin R} Y_i \leq k-\ell\right]}$$

Let $x(R) = \sum_{i\in R} x_i$. Then by Chernoff bound,

$$\mathsf{Pr}\left[Y > k-\ell\right] \leq \exp\left(-\frac{\left((\beta-1)k + x(R) - \beta\ell\right)^2}{3\beta(k - x(R))}\right) \leq \exp\left(-\frac{\left((\beta-1)k - \beta d\right)^2}{3\beta k}\right) \qquad (3.63)$$

which finishes the proof. □

3.9.6 The Proof of the Main Result

The main aim of this section is prove the $(1+\frac{\epsilon}{\sqrt{1+\frac{\lambda}{\|V(x)V(x)\|_2}}})$-approximation guarantee of the λ-regularized proportional volume sampling algorithm for λ-regularized A-optimal design. The main result is stated formally in Theorem 3.9.6.

Lemma 3.9.5 shows that our constructed μ is (c,β)-near-pairwise independent for some c dependent on β. Theorem 3.9.4 translates this property to the $(c\beta)$-approximation guarantee to $\beta\lambda$-regularized A-optimal design problem. However, this is a gap between the optimum of $\beta\lambda$-regularized A-optimal design and that of λ-regularized A-optimal design. This gap obviously depends on β and is quantified in Claim 5. Therefore, we want to pick β small enough to bound $(c\beta)$-approximation guarantee but also big enough to exploit this gap. The optimization of β is done formally in Theorem 3.9.6, giving the $(1+\frac{\epsilon}{\sqrt{1+\frac{\lambda}{\|V(x)V(x)\|_2}}})$-approximation guarantee.

Before proving the main theorem, we first simplify the parameter c of (c,β)-near-pairwise independent μ that we constructed. The calculation shows that $k = \Omega\left(\frac{d}{\epsilon}+\frac{\log(1/\epsilon)}{\epsilon^2}\right)$ is a right condition to obtain $c \leq 1+\epsilon$.

Claim 4. *Let $\epsilon' > 0, \beta > 1$. Suppose*

$$k \geq \frac{2\beta d}{\beta - 1} + \frac{3\beta}{(\beta - 1)^2} \log(1/\epsilon') \tag{3.64}$$

Then

$$\exp\left(-\frac{(\beta - 1)k - \beta d)^2}{3\beta k}\right) \leq \epsilon' \tag{3.65}$$

Proof. (3.65) is equivalent to

$$(\beta - 1)k - \beta d \geq \sqrt{3\beta \log(1/\epsilon')k}$$

which, by solving the quadratic equation in $\sqrt{k}$, is further equivalent to

$$\sqrt{k} \geq \frac{\sqrt{3\beta \log(1/\epsilon')} + \sqrt{3\beta \log(1/\epsilon') + 4(\beta - 1)\beta d}}{2(\beta - 1)}$$

Using inequality $\sqrt{a} + \sqrt{b} \leq \sqrt{2(a + b)}$, we have

$$\begin{aligned} \frac{\sqrt{3\beta \log(1/\epsilon')} + \sqrt{3\beta \log(1/\epsilon') + 4(\beta - 1)\beta d}}{2(\beta - 1)} &\leq \frac{\sqrt{3\beta \log(1/\epsilon') + 2(\beta - 1)\beta d}}{\beta - 1} \\ &= \sqrt{\frac{3\beta}{(\beta - 1)^2} \log(1/\epsilon') + \frac{2\beta d}{\beta - 1}} \end{aligned}$$

so the result follows. □

Next, we quantify the gap of the optimum of $\beta\lambda$-regularized A-optimal design and that of λ-regularized A-optimal design.

Claim 5. *Let $M \in \mathbb{R}^{d \times d}$ be a PSD matrix, and let $\beta, \lambda \geq 0$. Then,*

$$\frac{E_{d-1}(M + \beta\lambda I)}{E_d(M + \beta\lambda I)} \leq \frac{1 + \frac{\lambda}{\|M\|_2}}{1 + \beta\frac{\lambda}{\|M\|_2}} \frac{E_{d-1}(M + \lambda I)}{E_d(M + \lambda I)}$$

Proof. Let γ be eigenvalues of M. Then, $\frac{\gamma_i+\lambda}{\gamma_i+\beta\lambda} \leq \frac{\|M\|_2+\lambda}{\|M\|_2+\beta\lambda} = \frac{1+\frac{\lambda}{\|M\|_2}}{1+\beta\frac{\lambda}{\|M\|_2}}$ for all $i \in [d]$. Therefore,

$$\begin{aligned}\frac{E_{d-1}\left(M+\beta\lambda I\right)}{E_d\left(M+\beta\lambda I\right)} &= \sum_{i=1}^{d} \frac{1}{\gamma_i+\beta\lambda} \\ &\leq \frac{1+\frac{\lambda}{\|M\|_2}}{1+\beta\frac{\lambda}{\|M\|_2}} \sum_{i=1}^{d} \frac{1}{\gamma_i+\lambda} = \frac{1+\frac{\lambda}{\|M\|_2}}{1+\beta\frac{\lambda}{\|M\|_2}} \frac{E_{d-1}\left(M+\lambda I\right)}{E_d\left(M+\lambda I\right)}\end{aligned}$$

as desired. □

Now we are ready to state and prove the main result of this section.

Theorem 3.9.6. *Let* $V = [v_1, \ldots, v_n] \in \mathbb{R}^{d\times n}, \epsilon \in (0,1), \lambda \geq 0$, $x \in [0,1]^n$ *and suppose*

$$k \geq \frac{10d}{\epsilon} + \frac{60}{\epsilon^2}\log(4/\epsilon) \tag{3.66}$$

Denote $\lambda' = \frac{\lambda}{\|V(x)V(x)^\top\|_2}$. *Then the* λ*-proportional volume sampling* μ' *with hard-core measure* μ *with parameter* $z_i := \frac{x_i}{\beta - x_i}$ *(coordinate-wise) with* $\beta = 1 + \frac{\epsilon}{4}\sqrt{1+\lambda'}$ *satisfies*

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\frac{E_{d-1}\left(Z_{\mathcal{S}}(\lambda)\right)}{E_d\left(Z_{\mathcal{S}}(\lambda)\right)}\right] \leq \left(1+\frac{\epsilon}{\sqrt{1+\lambda'}}\right)\frac{E_{d-1}\left(V(x)V(x)^\top+\lambda I\right)}{E_d\left(V(x)V(x)^\top+\lambda I\right)} \tag{3.67}$$

Therefore, Algorithm 3.4 gives $(1+\frac{\epsilon}{\sqrt{1+\lambda'}})$*-approximation ratio to* λ*-regularized A-optimal design problem.*

The approximation guarantee of Algorithm 3.4 follows from x being a convex solution to the λ-regularized *A*-optimal design, so the objective given by x is at most the optimal integral solution of the λ-regularized *A*-optimal design problem.

Proof. Denote $\beta_{\lambda'} = 1 + \frac{\epsilon\sqrt{1+\lambda'}}{4}$ and $\beta_0 = 1 + \frac{\epsilon}{4}$. By inequality (3.66),

$$k \geq \frac{10d}{\epsilon} + \frac{60}{\epsilon^2}\log(4/\epsilon) = \frac{5d}{2(\beta_0 - 1)} + \frac{15}{4(\beta_0 - 1)^2}\log(4/\epsilon) \tag{3.68}$$

$$\geq \frac{2\beta_0 d}{\beta_0 - 1} + \frac{3\beta_0}{(\beta_0 - 1)^2}\log(4/\epsilon) \tag{3.69}$$

The last inequality is by $\beta_0 = 1 + \frac{\epsilon}{4} \leq \frac{5}{4}$. We have $\frac{\beta_0}{\beta_0 - 1} \geq \frac{\beta_{\lambda'}}{\beta_{\lambda'}-1}$ and

$$\frac{\beta_0}{(\beta_0-1)^2} = \frac{1}{\beta_0 - 1} + \frac{1}{(\beta_0-1)^2} = \frac{\sqrt{1+\lambda'}}{\beta_{\lambda'}-1} + \frac{(\sqrt{1+\lambda'})^2}{(\beta_{\lambda'}-1)^2} \geq \frac{\sqrt{1+\lambda'}}{\beta_{\lambda'}-1} + \frac{\sqrt{1+\lambda'}}{(\beta_{\lambda'}-1)^2}$$

$$= \sqrt{1+\lambda'}\frac{\beta_{\lambda'}}{(\beta_{\lambda'}-1)^2}$$

Therefore, (3.69) implies

$$k \geq \frac{2\beta_{\lambda'} d}{\beta_{\lambda'} - 1} + \frac{3\beta_{\lambda'}}{(\beta_{\lambda'}-1)^2}\sqrt{1+\lambda'}\log(4/\epsilon) \tag{3.70}$$

By Lemmas 3.9.5, μ is (c, β)-near-pairwise independent for $c = \frac{1}{1-\exp\left(-\frac{(\beta-1)k-\beta d)^2}{3\beta k}\right)}$. We now use Claim 4 to bound c: with the choice of $\beta = \beta_{\lambda'}$ and $\epsilon' = \left(\frac{\epsilon}{4}\right)^{\sqrt{1+\lambda'}}$ in Claim 4, we have $c \leq \frac{1}{1-\epsilon'}$. Therefore, by Theorem 3.9.4, Algorithm 3.4 guarantees objective with factor at most $c\beta = \frac{\beta}{1-\epsilon'}$ from optimum of $\beta\lambda$-regularized *A*-optimal design, i.e.

$$\mathop{\mathbb{E}}_{\mathcal{S}\sim\mu'}\left[\frac{E_{d-1}\left(Z_{\mathcal{S}}(\lambda)\right)}{E_d\left(Z_{\mathcal{S}}(\lambda)\right)}\right] \leq \frac{\beta}{1-\epsilon'}\frac{E_{d-1}\left(V(x)V(x)^\top + \beta\lambda I\right)}{E_d\left(V(x)V(x)^\top + \beta\lambda I\right)} \tag{3.71}$$

Now we apply Claim 5 to exploit the gap between λ- and $\beta\lambda$-regularized *A*-optimal design:

$$\frac{E_{d-1}\left(V(x)V(x)^\top + \beta\lambda I\right)}{E_d\left(V(x)V(x)^\top + \beta\lambda I\right)} \leq \frac{1+\lambda'}{1+\beta\lambda'} \cdot \frac{E_{d-1}\left(V(x)V(x)^\top + \beta\lambda I\right)}{E_d\left(V(x)V(x)^\top + \beta\lambda I\right)}$$

Therefore, Algorithm 3.4 gives approximation ratio of

$$\begin{aligned}\frac{\beta}{1-\epsilon'} \cdot \frac{1+\lambda'}{1+\beta\lambda'} &= \left(1+\frac{\beta-1}{1+\beta\lambda'}\right)(1-\epsilon')^{-1} \leq \left(1+\frac{\beta-1}{1+\lambda'}\right)(1-\epsilon')^{-1} \\ &= \left(1+\frac{\epsilon}{4\sqrt{1+\lambda'}}\right)(1-\epsilon')^{-1}\end{aligned}$$

As $\epsilon/4 < 1/e$, we have the inequality $\epsilon' = \left(\frac{\epsilon}{4}\right)^{\sqrt{1+\lambda'}} \leq \frac{\epsilon}{4\sqrt{1+\lambda'}}$. Hence, $(1-\epsilon')^{-1} \leq \left(1-\frac{\epsilon}{4\sqrt{1+\lambda'}}\right)^{-1}$. Thus, the approximation factor is at most

$$\left(1+\frac{\epsilon}{4\sqrt{1+\lambda'}}\right)\left(1-\frac{\epsilon}{4\sqrt{1+\lambda'}}\right)^{-1} \leq 1+\frac{\epsilon}{\sqrt{1+\lambda'}} \tag{3.72}$$

where the inequality is by $\epsilon \leq 1$. □

Note that we could have used fractional solution x from solving convex relaxation with regularizer $\beta\lambda$ instead of λ in Algorithm 3.4. This does not change the approximation ratio of the algorithm nor the proof, but in practice this gives a smaller value of $\frac{E_{d-1}\left(V(x)V(x)^\top+\beta\lambda I\right)}{E_d\left(V(x)V(x)^\top+\beta\lambda I\right)}$ to more tightly bound the objective of the algorithm using (3.59).

3.9.7 Efficient Implementation of λ-Regularized Proportional Volume Sampling

In this section, we show that λ-regularized proportional volume sampling can be implemented in polynomial time. The deterministic counterpart and its generalized version that naturally follows Section 3.5.3 (λ-regularized proportional ℓ-volume sampling – sampling S with $\mu'(S) \propto z^S E_\ell(V_S V_S^\top + \lambda I)$ to solve the generalized ratio objective with regularizer) can also be implemented in polynomial time by following a similar argument.

The following is the main statement for efficient implementation of λ-regularized version of proportional volume sampling. The standard counterpart was stated in Theorem 3.6.2.

Theorem 3.9.7. *Let $z \in \mathbb{R}^n_+, v_1, \ldots, v_n \in \mathbb{R}^d, \lambda \geq 0, 1 \leq k \leq n, \mathcal{U} \in \{\mathcal{U}_k, \mathcal{U}_{\leq k}\}$, and $V = [v_1, \ldots, v_n]$. Then there is a randomized algorithm $\mathcal{A}$ that runs in $\operatorname{poly}(n,d)$ time which outputs*

$\mathcal{S} \in \mathcal{U}$ such that

$$\Pr_{\mathcal{S}\sim\mathcal{A}}[\mathcal{S}=S] = \frac{z^S \det(V_S V_S^\top + \lambda I)}{\sum_{S'\in\mathcal{U}} z^{S'} \det(V'_S V'^\top_S + \lambda I)} =: \mu'(S)$$

That is, the algorithm correctly implements λ-regularized proportional volume sampling μ' with hard-core measure μ on $\mathcal{U}$ with parameter z. The algorithm runs in $O\left(n^4 dk^2 \log(dk)\right)$ number of arithmetic operations.

Moreover, there is an efficient derandomization of the algorithm. The algorithm also runs in the same time complexity $O\left(n^4 dk^2 \log(dk)\right)$ number of arithmetic operations.

Proof. The argument follows similarly with one in Theorem 3.6.2, with some modification of calculation later in the proof. We sample by starting with an empty set $S = \emptyset$. Then, in each step $i = 1, 2, \ldots, n$, decide with the correct probability

$$\Pr_{\mathcal{S}\sim\mu'}[i \in \mathcal{S} | I \subseteq \mathcal{S}, J \cap \mathcal{S} = \emptyset]$$

whether to include i in S or not, given the previous outcome. Let $I' = I \cup \{i\}$. This probability equals to

$$\begin{aligned}\Pr_{\mathcal{S}\sim\mu'}[i \in \mathcal{S} | I \subseteq \mathcal{S}, J \cap \mathcal{S} = \emptyset] &= \frac{\Pr_{\mathcal{S}\sim\mu'}[I' \subseteq \mathcal{S}, J \cap \mathcal{S} = \emptyset]}{\Pr_{\mathcal{S}\sim\mu'}[I \subseteq \mathcal{S}, J \cap \mathcal{S} = \emptyset]} \\ &= \frac{\sum_{S\in\mathcal{U}, I'\subseteq S, J\cap S=\emptyset} z^S \det(V_S V_S^\top + \lambda I)}{\sum_{S\in\mathcal{U}, I\subseteq S, J\cap S=\emptyset} z^S \det(V_S V_S^\top + \lambda I)} \\ &= \frac{\sum_{S\in\mathcal{U}, I'\subseteq S, J\cap S=\emptyset} z^S \sum_{h=0}^d \lambda^{d-h} \sum_{|R|=h, R\subset S} \det(V_R^\top V_R)}{\sum_{S\in\mathcal{U}, I\subseteq S, J\cap S=\emptyset} z^S \sum_{h=0}^d \lambda^{d-h} \sum_{|R|=h, R\subset S} \det(V_R^\top V_R)} \\ &= \frac{\sum_{h=0}^d \lambda^{d-h} \sum_{S\in\mathcal{U}, I'\subseteq S, J\cap S=\emptyset} z^S \sum_{|R|=h, R\subset S} \det(V_R^\top V_R)}{\sum_{h=0}^d \lambda^{d-h} \sum_{S\in\mathcal{U}, I\subseteq S, J\cap S=\emptyset} z^S \sum_{|R|=h, R\subset S} \det(V_R^\top V_R)}\end{aligned}$$

where we apply Lemma 3.9.3 and the Cauchy-Binet formula in the third equality. Both the numerator and denominator are sums over terms in the form $\sum_{S\in\mathcal{U}, A\subseteq S, J\cap S=\emptyset} z^S \sum_{|R|=h, R\subset S} \det(V_R^\top V_R)$ for some set $A \subseteq \mathcal{U}$ and $h = 0, 1, \ldots, d$. We have shown in the proof of Theorem 3.6.2 that a term in such form can be computed in polynomial time. More specifically, for each $A \in \{I', I\}$, we compute polynomial $F(t_1, t_2, t_3)$ in Lemma 3.6.1 only once to find the coefficients of the all monomials $t_1^{k_0} t_2^{d_0} t_3^{|I|}$ for $k_0 = 0.1. \dots, k$ and $d_0 = 0, 1, \dots, d$, giving the value of $\sum_{S \subseteq \mathcal{U}, |S| = k_0, A \subseteq S, J \cap S = \emptyset} z^S \sum_{|R| = d_0, R \subset S} \det(V_R^\top V_R)$ for each k_0, d_0. Hence, the sampling can be done both for $\mathcal{U} = \mathcal{U}_k$ (when we just need $k_0 = k$), and for $\mathcal{U} = \mathcal{U}_{\leq k}$ when we need values for $k_0 = 1, 2, \dots k$. Computing polynomial $F(t_1, t_2, t_3)$ takes $O\left(n^3 dk^2 \log(dk)\right)$ number of arithmetic operations by Lemma 3.6.1 and is the bottleneck in each of the n sampling steps, and hence the total runtime is $O\left(n^4 dk^2 \log(dk)\right)$ number of arithmetic operations.

Derandomization can be done identically to obtain the same result as in Theorem 3.6.5. Generlization to ℓ-volume sampling can be done identically to Theorem 3.6.11. The runtimes for λ-regularized counterpart are the same for both theorems. The modifications of proofs to obtain the results are identical to the proof of Theorem 3.9.7. That is, to expand any terms in the form $E_d(V_S V_S^\top + \lambda I)$ (or $E_h(V_S V_S^\top + \lambda I)$ for other h's) into polynomial in λ with coefficients in the form $E_{d_0}(V_S V_S^\top)$, and use Lemma 3.6.1 to calculate all terms of interests for all $d_0 = 0, 1, \dots, d$. $\square$

CHAPTER 4
COMBINATORIAL ALGORITHMS FOR OPTIMAL DESIGN

4.1 Introduction

One of the classical optimization methods that is used for optimal design problems is the local search heuristic which is also called the Fedorov's exchange method [Fed72] (see also [MMJ70]). The method starts with any set of k experiments from the given set of n experiments and aims to exchange one of the design vectors if it improves the objective. The ease in implementing the method as well as its efficacy in practice makes the method widely used [NM92] and implemented in statistics softwares such as SAS (see [ADT07], Chapter 13). Moreover, there has been considerable study on heuristically improving the performance of the algorithm. Surprisingly, theoretical analysis of this classical algorithm has not been performed despite its wide usage. In this thesis, we bridge this gap and give theoretical guarantees on the performance of local search heuristic for D and A-optimal design problems. In addition to local search, we analyze the greedy heuristic for the D and A-optimal design problems.

4.1.1 Main Approximation Results of Combinatorial Algorithms

Our main contribution is to prove worst case bounds on the performance of simple local search algorithm (also known as Fedorov Exchange method) and greedy algorithms. Our results also give worst case performance guarantee on the variants of local search algorithm.

Our first result is for the D-optimal design problem where we show the following guarantee. We consider both settings when the design vectors are allowed to be repeated in the solution and when they are not allowed to be repeated.

Theorem 4.1.1. *For any $\epsilon > 0$, the local search algorithm returns a $(1+\epsilon)$-approximate solution for D-*DESIGN *with or without repetitions whenever $k \geq d + \frac{d}{\epsilon}$.*

Our analysis method crucially uses the convex relaxation for the D-DESIGN problem. In recent works, the convex relaxation has been studied extensively and various rounding algorithms have been designed ([WYS16, ALSW17b, SX18, NST19]). Solving the convex relaxation is usually the bottleneck in the running time of all these algorithms. Our results differ from this literature in that we only use the convex relaxation for the analysis of the local search heuristic. The algorithm does not need to solve the convex program (or even formulate it). We use the *dual-fitting* approach to prove the guarantee. We also remark the above guarantee improves on the best previous bound, that gave $(1+\epsilon)$-approximation for $k = \Omega\left(\frac{d}{\epsilon} + \frac{1}{\epsilon^2}\log\frac{1}{\epsilon}\right)$ and so had an additional additive term of $\frac{1}{\epsilon^2}\log\frac{1}{\epsilon}$ in the requirement on the size of k.

We also consider the natural greedy algorithm for D-DESIGN problem. Indeed this algorithm has also been implemented and tested in empirical studies (see for example [ADT07], Chapter 12) and is referred to as the forward procedure algorithm. The algorithm is initialized to a small set of experiments and new experiments are added greedily. We show that the guarantee is slightly specific to the initialized set. If the initialized set is a local optimum set of size d, we obtain the following result. Again we employ the dual-fitting approach to prove the bounds.

Theorem 4.1.2. *For any $\epsilon > 0$, the greedy algorithm for D-*DESIGN *with repetitions returns a $(1+\epsilon)$-approximate solution whenever $k \geq \Omega\left(\frac{d}{\epsilon}\left(\log\frac{1}{\epsilon} + \log\log d\right)\right)$.*

A-DESIGN. While the simple combinatorial algorithms have tight asymptotic guarantee for D-DESIGN, we show that a similar guarantee *cannot* be proven for A-DESIGN. Indeed, there are examples where local optimum can be arbitrarily bad as compared to the optimum solution as we show in Section 4.3.3. We note that the bad local optima arise due to presence of long vectors among design vectors. In particular, we show that this is the *only* bottleneck to obtain an asymptotic guarantee on the performance of the local search algorithm. Moreover, we show a combinatorial iterative procedure to truncate the length of all the vectors while ensuring that the value of the optimal solution does not change significantly. This allows us to obtain a modified local search procedure with the following guarantee.

Theorem 4.1.3. *The modified local search algorithm for A-DESIGN with repetitions returns a $(1+\epsilon)$-approximate solution whenever $k = \Omega\left(\frac{d}{\epsilon^4}\right)$.*

We note that the above asymptotic guarantee does not match the best approximation algorithms [NST19] for A-DESIGN as was the case of D-DESIGN. Nonetheless, it specifically points why local search algorithm performs well in practice as has been noted widely [ADT07].

We also consider the natural greedy algorithm for the A-DESIGN problem, which again requires truncating the length of all vectors. As in D-DESIGN problem, the guarantee depends on the initialized set. If the initialized set is a local optimum set of size cd for an absolute constant c, we obtain the following guarantee.

Theorem 4.1.4. *The modified greedy algorithm for A-DESIGN with repetitions returns a $(1+\epsilon)$-approximate solution whenever $k \geq \Omega\left(\frac{d}{\epsilon^3}\log^2\frac{1}{\epsilon}\right)$.*

Approximate Local Search: Theorem 4.1.1 and 4.1.3 show that the local search for D-DESIGN and modified local search for A-DESIGN yield $(1+\epsilon)$-approximation algorithm. But, as are typical of local search algorithms, they are usually not polynomial time algorithms. However, the standard fix is to make local improvements only when the objectives improves by a factor of $1+\delta$. With appropriately chosen δ, this implies a polynomial running time at the cost of a slight degradation in the approximation guarantee. We show that under the same assumption on parameter k, approximate local search for D-DESIGN and modified approximate local search for A-DESIGN yield $(1+2\epsilon)$-approximation when δ is small enough and take polynomially many iterations.

Theorem 4.1.5. *The $(1+\delta)$-approximate local search algorithm for D-DESIGN with repetitions returns a $(1+2\epsilon)$-approximate solution whenever $k \geq d + \frac{d}{\epsilon}$ and $\delta < \frac{\epsilon d}{2k}$, and the algorithm runs in polynomial time.*

Theorem 4.1.6. *The modified $(1+\delta)$-approximate local search algorithm for A-DESIGN with repetitions returns a $(1+2\epsilon)$-approximate solution whenever $k = \Omega\left(\frac{d}{\epsilon^4}\right)$ and $\delta < \frac{\epsilon d}{2k}$, and the algorithm runs in polynomial time.*

Runtime of approximate local search algorithms are $O\left(\frac{Lknd^3 \log d + knd^2 \log k}{\delta}\right)$ where L is the bit biggest bit complexity of entries in input vectors (details are in Sections 4.6 and 4.7). We note that approximate local optimum sets are sufficient for initialization of greedy algorithms, implying that greedy algorithms run in polynomial time.

4.1.2 Related Work

Please refer to Related Work from previous chapter in Section 3.1.2.

4.1.3 Organization

In Section 4.2, we analyze the local search algorithm for D-DESIGN and prove Theorem 4.1.1. In Section 4.3, we analyze the modified local search algorithm for A-DESIGN and prove Theorem 4.1.3. Sections 4.4 and 4.5 include details and proofs deferred from the main body of the paper. We present approximate local search algorithms for D-DESIGN and A-DESIGN and their analysis in Sections 4.6 and 4.7, respectively, proving Theorems 4.1.5 and 4.1.6. Greedy algorithms and their analysis for D-DESIGN and A-DESIGN are presented in Sections 4.8 and 4.9, respectively, which prove Theorems 4.1.2 and 4.1.4.

4.2 Local Search for D-DESIGN

We first give the local search algorithm for D-DESIGN with repetitions.

Algorithm 4.1 Local search algorithm for D-DESIGN

```
Input: $V = \{v_1, \dots, v_n\}$ where $v_i \in \mathbb{R}^d, d \leq k \in \mathbb{N}$.
Let $I$ be any (multi)-subset of $[1, n]$ of size $k$ such that $X = \sum_{i \in I} v_i v_i^\top$ is non-singular matrix.
While $\exists i \in I, j \in [1, n]$ such that $\det\left(X - v_i v_i^\top + v_j v_j^\top\right) > \det(X)$:
    $X \leftarrow X - v_i v_i^\top + v_j v_j^\top$
    $I \leftarrow I \setminus \{i\} \cup \{j\}$
Return $(I, X)$
```

4.2.1 Local Search Algorithm

4.2.2 Relaxations

To prove the performance of local search algorithm, presented earlier as Theorem 4.1.1, we use the convex programming relaxation for the D-DESIGN problem. We first describe these relaxations in Figure 2.2c in Preliminaries. (see Chapter 7 of [BV04]). Let OPT denote the be the common optimum value of (D-REL) and its dual (D-REL-DUAL). Let $I^\star$ denote the indices of the vector in the optimal solution and let $\phi^{\mathsf{D}} = \det\left(\sum_{i \in I^\star} v_i v_i^\top\right)^{\frac{1}{d}}$ be its objective. Recall that $\phi_f^{\mathsf{D}} \geq \log \phi^{\mathsf{D}}$. Theorem 4.1.1 now follows from the following result.

Theorem 4.2.1. *Let X be the solution returned by Algorithm 4.1. Then,*

$$\det(X) \geq \left(\frac{k-d+1}{k}\right)^d e^{d \cdot \phi_f^{\mathsf{D}}}$$

and therefore,

$$\det(X)^{\frac{1}{d}} \geq \frac{k-d+1}{k} \cdot \phi^{\mathsf{D}}.$$

Before we prove Theorem 4.2.1, we begin with a few definitions. Let (I, X) be the returned solution of the algorithm. Let V_I be the $d \times |I|$ matrix whose columns are v_i for each $i \in I$. Observe that $X = V_I V_I^\top$ and X is invertible since $\det(X) > 0$ at the beginning of the algorithm and $\det(X)$ only increases in later iterations. We let $\tau_i = v_i^\top X^{-1} v_i$ for any $1 \leq i \leq n$. Observe that if $i \in I$, then τ_i is the leverage score of row v_i with respect to the matrix $V_I^\top$. We also let

$\tau_{ij} = v_i^\top X^{-1} v_j$ for any $1 \leq i, j \leq n$.

Notations: For convenience, we summarize the notations used in this section.

- OPT is the common optimum value of (D-REL) and its dual (D-REL-DUAL).
- $I^\star \subseteq [1, n]$ is the set of indices of the vectors in the optimal solution.
- $\phi^{\mathrm{D}} = \det\left(\sum_{i \in I^\star} v_i v_i^\top\right)^{\frac{1}{d}}$, the integral optimum value of D-DESIGN
- $I \subseteq [1, n], X = \sum_{i \in I} v_i v_i^\top$ is the solution returned by the algorithm.
- For $1 \leq i \leq n$, $\tau_i = v_i^\top X^{-1} v_i$.
- For $1 \leq i, j \leq n$, $\tau_{ij} = v_i^\top X^{-1} v_j$.

The following lemma states standard properties about leverage scores of vectors with respect to the PSD matrix $X = \sum_{i \in I} v_i v_i^\top$ (see for example [DMIMW12]). These results hold even when X is not an output from a local search algorithm and the proof is included in the appendix.

Lemma 4.2.2. *Let $v_1, \ldots, v_n \in \mathbb{R}^d$ and $I \subseteq [n]$. For any matrix $X = \sum_{i \in I} v_i v_i^\top$, we have:*

1. *For any $i \in I$, we have $\tau_i \leq 1$. Moreover, for any $i \in I$, $\tau_i = 1$ if and only if $X - v_i v_i^\top$ is singular.*
2. *We have $\sum_{i \in I} \tau_i = d$.*
3. *For any $1 \leq j \leq n$, we have $\sum_{i \in I} \tau_{ij} \tau_{ji} = \tau_j$.*
4. *For any $1 \leq i, j \leq n$, we have $\tau_{ij} = \tau_{ji}$ and $\tau_{ij} \leq \sqrt{\tau_i \tau_j}$.*

We now prove an upper bound on τ_j for the local optimal solution. This lemma utilizes the local optimality condition crucially.

Lemma 4.2.3. *For any $j \in [1, n]$, $\tau_j \leq \frac{d}{k-d+1}$.*

Before we prove the lemma, we complete the proof of Theorem 4.2.1 using Lemma 4.2.3.

Theorem 4.2.1. We construct a feasible solution to the (D-REL-DUAL) of the objective value at most $\frac{1}{d}\log\det(X) + \log\frac{k}{k-d+1}$. This would imply that

$$\phi_f^{\mathsf{D}} \leq \frac{1}{d}\log\det(X) + \log\frac{k}{k-d+1}$$

which proves the first part of the theorem. The second part follows since $\phi_f^{\mathsf{D}} \geq \log\phi^{\mathsf{D}}$.

Let $Y = \alpha X, \mu = \max_{1\leq j\leq n} v_j^\top Y^{-1} v_j = \frac{1}{\alpha}\max_{j\in[1,n]} v_j^\top X^{-1} v_j$ where $\alpha > 0$ will be fixed later. Then, (Y,μ) is a feasible solution of (D-REL-DUAL). Hence,

$$\begin{aligned}\phi_f^{\mathsf{D}} &\leq \frac{1}{d}\log\det(\alpha X) + \frac{k}{d}\cdot\frac{1}{\alpha}\max_{j\in[1,n]} v_j^\top X^{-1} v_j - 1\\ &\leq \log\alpha + \frac{1}{d}\log\det(X) + \frac{k}{d\alpha}\cdot\frac{d}{k-d+1} - 1 \qquad \text{(Lemma 4.2.3)}\end{aligned}$$

Setting $\alpha = \frac{k}{k-d+1}$, we get

$$\phi_f^{\mathsf{D}} \leq \log\frac{k}{k-d+1} + \frac{1}{d}\log\det(X) + 1 - 1 = \log\frac{k}{k-d+1} + \frac{1}{d}\log\det(X)$$

as required. □

We now prove Lemma 4.2.3.

Lemma 4.2.3. Since X is a symmetric matrix, X^{-1} is also a symmetric matrix and therefore $\tau_{ij} = \tau_{ji}$ for each i, j. We first show that the local optimality condition implies the following claim:

Claim 6. *For any $i \in I$ and $1 \leq j \leq n$, we have $\tau_j - \tau_i\tau_j + \tau_{ij}\tau_{ji} \leq \tau_i$.*

Proof. Let $i \in I, j \in [1,n]$. By local optimality of I,

$$\det(X - v_i v_i^\top + v_j v_j^\top) \leq \det(X).$$

Next we cite the following lemma for a determinant formula.

Lemma 4.2.4. *(Matrix Determinant Lemma, [Har97]) For any invertible matrix $A \in \mathbb{R}^{d\times d}$ and $a, b \in \mathbb{R}^d$,*

$$\det(A + ab^\top) = \det(A)(1 + b^\top A^{-1} a)$$

Applying the Lemma twice to $\det(X - v_i v_i^\top + v_j v_j^\top)$, the local optimality condition implies that

$$\begin{aligned}\det(X) \geq \det(X - v_i v_i^\top + v_j v_j^\top) &= \det(X + v_j v_j^\top)(1 - v_i^\top (X + v_j v_j^\top)^{-1} v_i) \\ &= \det(X)(1 + v_j^\top X^{-1} v_j)(1 - v_i^\top (X + v_j v_j^\top)^{-1} v_i)\end{aligned}$$

Hence, $(1 + v_j^\top X^{-1} v_j)(1 - v_i^\top (X + v_j v_j^\top)^{-1} v_i) \leq 1$. Applying Sherman-Morrison formula, we get

$$\begin{aligned}(1 + v_j^\top X^{-1} v_j)\left(1 - v_i^\top \left(X^{-1} - \frac{X^{-1} v_j v_j^\top X^{-1}}{1 + v_j^\top X^{-1} v_j}\right) v_i\right) &\leq 1 \\ (1 + \tau_j)\left(1 - \tau_i + \frac{\tau_{ij}\tau_{ji}}{1 + \tau_j}\right) &\leq 1 \\ (1 - \tau_i)(1 + \tau_j) + \tau_{ij}\tau_{ji} &\leq 1 \\ \tau_j - \tau_i\tau_j + \tau_{ij}\tau_{ji} &\leq \tau_i.\end{aligned}$$

This finishes the proof of Claim 6. □

Now summing the inequality in Claim 6 over all $i \in I$, we get

$$\sum_{i \in I} (\tau_j - \tau_i\tau_j + \tau_{ij}\tau_{ji}) \leq \sum_{i \in I} \tau_i.$$

Applying Lemma 4.2.2, we obtain that $k\tau_j - d\tau_j + \tau_j \leq d$. Rearranging, we obtain that

$$\tau_j \leq \frac{d}{k-d+1}$$

as desired. □

4.2.3 D-DESIGN without Repetitions

We defer the proof of local search for D-DESIGN without repetitions to Section 4.4.

4.3 Local Search for A-DESIGN

In this section, we prove the performance of modified local search, presented earlier as Theorem 4.1.3. As remarked earlier, we need to modify the instance to cap the length of the vectors before applying the local search procedure. This is done in Section 4.3.1. We show that the value of any feasible solution only increases after capping. Moreover, the value of the natural convex programming relaxation increases by at most a small factor. We then analyze that the local search algorithm applied to vectors of short length returns a near optimal solution. Combining these facts give a complete analysis of modified local search for A-DESIGN in Section 4.3.2 which implies Theorem 4.1.3.

4.3.1 Capping Vectors

Algorithm 4.2 Capping vectors length for A-DESIGN

```
Input: $V = \{v_1, \dots, v_n\} \subseteq \mathbb{R}^d$, parameter $\Delta$.
While $\exists i \in [1, n], ||v_i||_2^2 > \Delta$:
$t = \text{argmax}_{i \in [n]} ||v_i||_2$.
For $j \in [1, n], v_j = \left( I_d - \frac{1}{2} \frac{v_t v_t^\top}{||v_t||_2^2} \right) v_j$
For $j \in [1, n], u_j = v_j$.
Return $U = \{u_1, \dots, u_n\} \subseteq \mathbb{R}^d$
```

The algorithm to cap the length of input vectors is given in Algorithm 4.2. In each iteration, it considers the longest vector v_t. If the length of this vector (and thus every vector) is at most Δ, then it returns the current updated vectors. Else, it scales down all the vectors along the direction of the longest vector. Here, I_d denotes the d-by-d identity matrix.

Before we give the guarantee about the algorithm, we introduce the convex program for the A-DESIGN problem in Figure 2.1c (see Chapter 7 of [BV04]) in Preliminaries. For any input vectors $V = \{v_1, \ldots, v_n\}$, the primal program is A-REL(V) and the dual program is A-REL-DUAL(V). We index these convex programs by input vectors V as we will analyze their objectives when the input vectors change by the capping algorithm. We let $\phi_f^{\mathsf{A}}(V)$ denote the (common) optimal objective value of both convex programs with input vectors V.

We prove the following guarantee about Algorithm 4.2. The proof along with some intuition of Algorithm 4.2 appears in the appendix.

Lemma 4.3.1. *For any input vectors $V = \{v_1, \ldots, v_n\} \subseteq \mathbb{R}^d$ and $k \geq d$, if $k \geq 15$ then the capping algorithm returns a set of vectors $U = \{u_1, \ldots u_n\}$ such that*

1. $\|u_i\|_2^2 \leq \Delta$ *for all* $i \in [n]$.
2. *For any (multi-)set* $S \subseteq [n]$, $\operatorname{tr}\left(\left(\sum_{i \in S} v_i v_i^\top\right)^{-1}\right) \leq \operatorname{tr}\left(\left(\sum_{i \in S} u_i u_i^\top\right)^{-1}\right)$.
3. $\phi_f^{\mathsf{A}}(U) \leq \left(1 + \frac{3000 \cdot d}{k}\right)\left(\phi_f^{\mathsf{A}}(V) + \frac{135 \cdot d}{\Delta}\right)$.

Lemma 4.3.1 states that if an algorithm returns a good solution from capped vectors, then the objective remains small after we map the solution back to the original (uncapped) input vectors. Moreover, by choosing a sufficiently large capping length Δ, we may bound the increase in optimal value of the natural convex programming relaxation after capping by a small factor. Optimizing for Δ is to be done later.

4.3.2 Local Search Algorithm

We now consider the local search algorithm with the capped vectors. The performance of the algorithm is stated as follows.

Algorithm 4.3 Local search algorithm for A-DESIGN with capped vectors

```
Input: $U = \{u_1, \ldots, u_n\} \subseteq \mathbb{R}^d, d \leq k \in \mathbb{N}$.
Let $I$ be any (multi)-subset of $[1, n]$ of size $k$ such that $X = \sum_{i \in I} u_i u_i^\top$ is nonsingular.
While $\exists i \in I, j \in [1, n]$ such that $\operatorname{tr}\left((X - u_i u_i^\top + u_j u_j^\top)^{-1}\right) < \operatorname{tr}(X^{-1})$:
$X = X - u_i u_i^\top + u_j u_j^\top$
$I = I \setminus \{i\} \cup \{j\}$
Return $(I, X)$
```

Theorem 4.3.2. *Let (I, X) be the solution returned by Algorithm 4.3. If $||u_i||_2^2 \leq \Delta$ for all $i \in [n]$,*

$$\operatorname{tr}(X^{-1}) \leq \phi_f^{\mathrm{A}}(U) \left(\left(1 - \frac{d-2}{k}\right) - \sqrt{\frac{\Delta \phi_f^{\mathrm{A}}(U)}{k}} \right)^{-1}.$$

The proof of Theorem 4.3.2 is deferred to the appendix. We now analyze the modified local search algorithm presented as Algorithm 4.4 with input vectors $V = \{v_1, \ldots, v_n\}$ which may contain vectors with long length using Theorem 4.3.2. Let $I^\star$ be the set of indices of the vectors in the optimal solution of A-DESIGN with input vector set V and let $\phi^{\mathrm{A}}(V) = \operatorname{tr}\left(\left(\sum_{i \in I^\star} v_i v_i^\top\right)^{-1}\right)$ be its objective. Observe that $\phi_f^{\mathrm{A}}(V) \leq \phi^{\mathrm{A}}(V)$.

Algorithm 4.4 Modified local search algorithm for A-DESIGN

```
Input: $V = \{v_1, \ldots, v_n\}, d \leq k \in \mathbb{N}$.
Let $\Delta = \frac{d}{\epsilon^2 \phi^{\mathrm{A}}(V)}$.
Let $U = \{u_1, \ldots, u_n\}$ be the output of Vector Capping Algorithm 4.2 with input $(V, \Delta)$.
Let $I \subseteq [1, n], X = \sum_{i \in I} u_i u_i^\top$ be the output of Local Search Algorithm 4.3 with input $(U, k)$.
Return $I$.
```

Theorem 4.3.3. *For input vectors $V = \{v_1, \ldots, v_n\}$ where $v_i \in \mathbb{R}^d$ and parameter k, let I be the* *solution returned by Algorithm 4.4. If $k \geq \frac{2d}{\epsilon^4}$ and $\epsilon \leq 0.001$, then*

$$\operatorname{tr}\left(\left(\sum_{i\in I} v_i v_i^\top\right)^{-1}\right) \leq (1+\epsilon)\phi^{\mathsf{A}}(V).$$

The $(1+\epsilon)$-approximation of Algorithm 4.4 is achieved by setting an appropriate capping length Δ and combining the guarantees from Lemma 4.3.1 and Theorem 4.3.2.

Proof. By Theorem 4.3.2,

$$\begin{aligned}\operatorname{tr}\left(\left(\sum_{i\in I} u_i u_i^\top\right)^{-1}\right) &\leq \phi_f^{\mathsf{A}}(U)\left(1-\frac{d-2}{k}-\sqrt{\frac{\Delta\phi_f^{\mathsf{A}}(U)}{k}}\right)^{-1}\\ &= \phi_f^{\mathsf{A}}(U)\left(1-\frac{\epsilon^4}{2}+\frac{\epsilon^4}{d}-\epsilon\sqrt{\frac{\phi_f^{\mathsf{A}}(U)}{2\phi^{\mathsf{A}}(V)}}\right)^{-1}\end{aligned}$$

The last inequality follows since $k \geq \frac{2d}{\epsilon^4}$ and $\Delta = \frac{d}{\epsilon^2\phi^{\mathsf{A}}(V)}$. By Lemma 4.3.1,

$$\phi_f^{\mathsf{A}}(U) \leq \left(1+1500\epsilon^4\right)\left(\phi_f^{\mathsf{A}}(V)+135\epsilon^2\phi_f^{\mathsf{A}}(V)\right).$$

Since $\phi_f^{\mathsf{A}}(V) \leq \phi^{\mathsf{A}}(V)$, we get $\phi_f^{\mathsf{A}}(U) \leq (1+1500\epsilon^4)(1+135\epsilon^2)\phi^{\mathsf{A}}(V)$. Substituting in the equation above, we get

$$\begin{aligned}\operatorname{tr}\left(\left(\sum_{i\in I} u_i u_i^\top\right)^{-1}\right) &\leq \phi^{\mathsf{A}}(V)\frac{(1+1500\epsilon^4)(1+135\epsilon^2)}{1-\frac{\epsilon^4}{2}+\epsilon^4/d-\epsilon\sqrt{(1+1500\epsilon^4)(1+135\epsilon^2)/2}}\\ &\leq (1+\epsilon)\phi^{\mathsf{A}}(V)\end{aligned}$$

where the last inequality follows from the fact that $\epsilon < 0.001$. By Lemma 4.3.1, we also have that

$\mathrm{tr}\left(\left(\sum_{i\in I} v_i v_i^\top\right)^{-1}\right) \leq \mathrm{tr}\left(\left(\sum_{i\in I} u_i u_i^\top\right)^{-1}\right)$. Hence,

$$\mathrm{tr}\left(\left(\sum_{i\in I} v_i v_i^\top\right)^{-1}\right) \leq (1+\epsilon)\phi^{\mathrm{A}}(V).$$

This finishes the proof of Theorem 4.3.3. □

Algorithm 4.4 requires the knowledge of the optimum solution value $\phi^{\mathrm{A}}(V)$. We can guess this value efficiently by performing a binary search. The details appear in the appendix.

4.3.3 Instances with Bad Local Optima

In this section, we show that preprocessing input vectors to the A-DESIGN problem is required for the local search algorithm to have any approximation guarantee. This is because a locally optimal solution can give an arbitrarily bad objective value compared to the optimum. Hence, this requirement applies regardless of implementations of the local search algorithm. We summarize the result as follows.

Theorem 4.3.4. *For any $k \geq d \geq 2$, there exists an instance of A-DESIGN, either with or without repetitions, such that a locally optimal solution has an arbitrarily bad approximation ratio.*

We note that any instance to A-DESIGN with repetitions can be used for A-DESIGN without repetitions by making k copies of each input vector. Therefore, it is enough to show example of instances only in A-DESIGN with repetitions. For each i, let e_i be the unit vector in the i^{th} dimension. In this section, N is a real number tending to infinity, and the $A(N) \sim B(N)$ notation indicates that $\lim_{N\to\infty} \frac{A(N)}{B(N)} = 1$. All asymptotic notions such as big-Oh are with respect to $N \to \infty$. We first show the bad instance when $k \geq d = 2$. Though $d = 2$ seems a small case to consider, the calculation presented is central to prove the main theorem later.

Lemma 4.3.5. *There exists an instance of A-DESIGN for $k \geq d = 2$, with repetitions, such that a locally optimal solution has an arbitrarily bad approximation ratio.*

The construction in Lemma 4.3.5 can be generalized to $d > 2$ dimensions by adding a vector with an appropriate length to each additional dimension. The proof of Theorem 4.3.4 appears in the appendix. We now prove the Lemma.

Proof. Let $v_1 = [1; \frac{1}{N^2}], v_2 = [1; -\frac{1}{N^2}]$, $w_1 = [N^4; \frac{1}{N}], w_2 = [N^4; -\frac{1}{N}]$, and let the input of A-DESIGN be these four vectors. We first make straightforward calculations, summarized as the following claim.

Claim 7. *Let p, q be positive integers. Then,*

$$\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top\right)^{-1}\right) = \frac{p+q}{4pq}N^4 + O(1) \tag{4.1}$$

$$\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top + w_1w_1^\top\right)^{-1}\right) = \frac{1}{p+q}N^4 + O(N) \tag{4.2}$$

$$\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top + w_2w_2^\top\right)^{-1}\right) = \frac{1}{p+q}N^4 + O(N) \tag{4.3}$$

$$\operatorname{tr}\left(\left(w_1w_1^\top + w_2w_2^\top\right)^{-1}\right) = \frac{N^2}{2} + O(N^{-8}) \tag{4.4}$$

Proof. We will repeatedly use the formula $\operatorname{tr}\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{-1}\right) = \frac{a+d}{ad-bc}$. We have

$$\begin{aligned}\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top\right)^{-1}\right) &= \operatorname{tr}\left(\begin{bmatrix} p+q & (p-q)N^{-2} \\ (p-q)N^{-2} & (p+q)N^{-4} \end{bmatrix}^{-1}\right) \\ &= \frac{p+q+(p+q)N^{-4}}{(p+q)^2N^{-4} - (p-q)^2N^{-4}} = \frac{p+q}{4pq}N^4 + O(1)\end{aligned}$$

$$\begin{aligned}\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top + w_1w_1^\top\right)^{-1}\right) &= \operatorname{tr}\left(\begin{bmatrix} N^8+p+q & N^3+(p-q)N^{-2} \\ N^3+(p-q)N^{-2} & N^{-2}+(p+q)N^{-4}\end{bmatrix}^{-1}\right) \\ &= \frac{N^8+O(1)}{(p+q)N^4+O(N)} = \frac{1}{p+q}N^4 + O(N)\end{aligned}$$

The calculation for $\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top + w_2w_2^\top\right)^{-1}\right)$ is symmetric. Finally, we have

$$\operatorname{tr}\left(w_1w_1^\top + w_2w_2^\top\right)^{-1} = \operatorname{tr}\left(\begin{bmatrix} 2N^8 & 0 \\ 0 & 2N^{-2}\end{bmatrix}^{-1}\right) = \frac{N^2}{2} + \frac{1}{2N^8}$$

finishing the proof. □

We now continue the proof of Lemma 4.3.5. Let $p = \lfloor \frac{k}{2} \rfloor, q = \lceil \frac{k}{2} \rceil$ and consider the solution S which has p and q copies of v_1 and v_2 respectively. By Claim 7, the current objective of S is $\operatorname{tr}\left(\left(pv_1v_1^\top + qv_2v_2^\top\right)^{-1}\right) \sim \frac{k}{4pq}N^4$ and the objective of $S \setminus \{v_i\} \cup \{w_j\}$ for any pair $i, j \in \{1,2\}$ is $\frac{1}{p+q-1}N^4 + O(N) \sim \frac{1}{k-1}N^4$. As $\frac{k}{4pq}N^4 \geq \frac{k}{k^2-1}N^4 > \frac{1}{k-1}N^4$ for $k \geq 2$, S is locally optimal.

However, consider another solution S^* which picks p and q copies of w_1 and w_2. Since $\operatorname{tr}\left(w_1w_1^\top + w_2w_2^\top\right)^{-1} = O(N^2)$, by monotonicity of $\operatorname{tr}((\cdot)^{-1})$ under Loewner ordering, we must have that the objective given by S^* is also at most $O(N^2)$, which is a $\Theta(N^2)$-factor smaller than the objective value of S. The result follows because N tends to infinity. □

4.4 Proofs from Section 4.2

We use the notation $\langle A, B\rangle$ for an inner product of two matrices A, B of the same size. We begin by stating the Sherman-Morrison formula that is important in our calculations. We instantiate it for symmetric matrices.

Theorem 4.4.1. *Let L be an $d \times d$ invertible matrix and $v \in \mathbb{R}^d$. Then*

$$\left(L + vv^\top\right)^{-1} = L^{-1} - \frac{L^{-1}vv^\top L^{-1}}{1 + v^\top L^{-1}v}$$

Lemma 4.4.2. *(Matrix Determinant Lemma, [Har97]) For any invertible matrix $L \in \mathbb{R}^{d\times d}$ and $v \in \mathbb{R}^d$,*

$$\det(L + vv^\top) = \det(L)(1 + v^\top L^{-1}v)$$

We now detail the missing proofs.

Lemma 4.2.2. Let $W = X_{-i} = X - v_i v_i^\top = \sum_{j\in I\setminus\{i\}} v_j v_j^\top$. To show $\tau_i \leq 1$, we make two cases depending on whether W is singular or not.

Case 1: W is non-singular.

$$\begin{aligned}
\tau_i &= v_i^\top (W + v_i v_i^\top)^{-1} v_i \\
&= v_i^\top \left(W^{-1} - \frac{W^{-1}v_i v_i^\top W^{-1}}{1 + v_i^\top W^{-1} v_i} \right) v_i \\
&= v_i^\top W^{-1} v_i - \frac{v_i^\top W^{-1} v_i v_i^\top W^{-1} v_i}{1 + v_i^\top W^{-1} v_i} \\
&= \frac{v_i^\top W^{-1} v_i + (v_i^\top W^{-1} v_i)^2 - (v_i^\top W^{-1} v_i)^2}{1 + v_i^\top W^{-1} v_i} \\
&= \frac{v_i^\top W^{-1} v_i}{1 + v_i^\top W^{-1} v_i} \\
&< 1.
\end{aligned}$$

Last inequality follows from the fact that $v_i^\top W^{-1} v_i > 0$ since W^{-1} is non-singular.

Case 2: W is singular. We have that X is non-singular and $W = X - v_i v_i^\top$ is a singular matrix. Let $Y^\dagger$ denote the Moore-Penrose pseudo-inverse of Y for any matrix Y. Observe that $X^\dagger = X^{-1}$. From Theorem 1 [Mey73], we have that

$$X^{-1} = W^\dagger - \frac{W^\dagger v_i v_i^\top (I - WW^\dagger)^\top}{\|(I - WW^\dagger)v_i\|_2^2} - \frac{(I - W^\dagger W)^\top v_i v_i^\top W^\dagger}{\|(I - W^\dagger W)^\top v_i\|_2^2} + \frac{(1 + v_i^\top W^\dagger v_i)(I - W^\dagger W)^\top v_i v_i^\top (I - WW^\dagger)^\top}{\|(I - W^\dagger W)^\top v_i\|_2^2 \|(I - WW^\dagger)v_i\|_2^2}$$

Now we use the fact that $(I - WW^\dagger)$ and $(I - W^\dagger W)$ are projection matrices. Since $v^\top P v = \|Pv\|_2^2$ for any projection matrix P and vector v, we obtain that

$$\begin{aligned} v_i^\top X^{-1} v_i &= v_i^\top W^\dagger v_i - \frac{\left(v_i^\top W^\dagger v_i\right)\left(v_i^\top (I - WW^\dagger)^\top v_i\right)}{\|(I - WW^\dagger)v_i\|_2^2} - \frac{\left(v_i^\top (I - W^\dagger W)^\top v_i\right) v_i^\top W^\dagger v_i}{\|(I - W^\dagger W)^\top v_i\|_2^2} \\ &\quad + \frac{(1 + v_i^\top W^\dagger v_i) v_i^\top (I - W^\dagger W)^\top v_i v_i^\top (I - WW^\dagger)^\top v_i}{\|(I - W^\dagger W)^\top v_i\|_2^2 \|(I - WW^\dagger)v_i\|_2^2} \\ &= v_i^\top W^\dagger v_i - v_i^\top W^\dagger v_i - v_i^\top W^\dagger v_i + (1 + v_i^\top W^\dagger v_i) \\ &= 1 \end{aligned}$$

as claimed.

We now show that $\sum_{i\in I} \tau_i = d$. Indeed

$$\sum_{i\in I} \tau_i = \sum_{i\in I} v_i^\top X^{-1} v_i = \sum_{i\in I} \langle X^{-1}, v_i v_i^\top \rangle = \langle X^{-1}, \sum_{i\in I} v_i v_i^\top \rangle = \langle X^{-1}, X \rangle = d$$

Similarly, we have

$$\begin{aligned} \sum_{i\in I} \tau_{ij}\tau_{ji} &= \sum_{i\in I} v_i^\top X^{-1} v_j v_j^\top X^{-1} v_i = \sum_{i\in I} \langle X^{-1} v_j v_j^\top X^{-1}, v_i v_i^\top \rangle = \langle X^{-1} v_j v_j^\top X^{-1}, \sum_{i\in I} v_i v_i^\top \rangle \\ &= \langle X^{-1} v_j v_j^\top X^{-1}, X \rangle = v_j^\top X^{-1} v_j \end{aligned}$$

For the last part, observe that X^{-1} is symmetric and thus $\tau_{ij} = \tau_{ji}$. Moreover,

$$\tau_{ij} = v_i^\top X^{-1} v_j = (X^{-\frac{1}{2}} v_i)^\top (X^{-\frac{1}{2}} v_j) \leq \|X^{-\frac{1}{2}} v_i\|_2 \|X^{-\frac{1}{2}} v_j\| = \sqrt{\tau_i \tau_j}$$

where the inequality follows from Cauchy-Schwarz. □

4.4.1 Local Search for D-DESIGN without Repetitions

In this section, we focus on the variant of D-DESIGN where repetitions of vectors are not allowed, and show the approximation guarantee of the local search in this setting. In comparison to D-DESIGN with repetitions, the relaxation now has an upper bound on x_i and extra nonnegative variables η_i on the dual. See the relaxation and its dual in Preliminaries.

The local search algorithm 4.1 is modified by considering a swap where elements to be included in the set must not be in the current set. We prove a similar approximation ratio of the local search algorithm for the without repetition setting.

Theorem 4.4.3. *Let X be the solution returned by the local search algorithm. Then for all $k \geq d+1$,*

$$\det(X) \geq \left(\frac{k-d}{k}\right)^d e^{d \cdot \phi_f^{\mathrm{D}}}$$

and therefore,

$$\det(X)^{\frac{1}{d}} \geq \frac{k-d}{k} \cdot \phi^{\mathrm{D}}.$$

We note that in the case $k = d$, the design problem without repetition is identical to with repetition since the optimal solution must be linearly independent, and thus the bound from with repetitions of Theorem 4.2.1 applies to obtain d-approximation.

The proof of Theorem 4.4.3 is similar to D design requires a different bound on τ_j from the setting with repetitions to set a feasible dual solution, since the local search condition no longer applies to all vectors $j \in [n]$ but only for those not in output set I. We first give a bound of τ_j for

$j \notin I$.

Lemma 4.4.4. *For any $j \notin S$ and any $i \in S$ such that $\tau_i < 1$,*

$$\tau_j \leq \frac{\tau_i}{1-\tau_i}.$$

Proof. We claim that the local search condition implies that for any $i \in I$ and $j \notin I$, we have

$$\tau_j - \tau_i \tau_j + \tau_{ij}\tau_{ji} \leq \tau_i. \tag{4.5}$$

The proof of the claim is identical to that of Claim 6. Hence, we have

$$\tau_i \geq \tau_j - \tau_i\tau_j + \tau_{ij}^2 \geq \tau_j - \tau_i\tau_j \tag{4.6}$$

which finishes the proof of the Lemma. □

We now prove the main Theorem.

Theorem 4.4.3. As in the proof of Theorem 4.2.1, we construct a feasible solution to the (D-REL-DUAL) of the objective value of at most $\frac{1}{d}\log\det(X) + \log\frac{k}{k-d}$ which is sufficient as a proof of the theorem. Denote $\tau_{\min} = \min_{j\in I} v_j^\top Y^{-1} v_j$. Let

$$Y = \alpha X, \qquad \mu = \frac{k}{\alpha(k-d)}\tau_{\min}, \qquad \eta_j = \begin{cases} 0, & j \notin I \\ \frac{\tau_j - \tau_{\min}}{\alpha} & j \in I \end{cases}$$

where $\alpha > 0$ will be fixed later. We first check the feasibility of the solution. It is clear by definition that $\mu, \eta_j \geq 0$. For $j \notin I$, by Lemma 4.4.4, we have

$$v_j^\top Y^{-1} v_j = \frac{1}{\alpha}\cdot\tau_j \leq \frac{1}{\alpha}\cdot\frac{\tau_{\min}}{1-\tau_{\min}} \leq \frac{1}{\alpha}\cdot\frac{k}{k-d}\tau_{\min} = \mu + \eta_j$$

where the second inequality follows from $\tau_{\min} \leq \frac{1}{k}\sum_{i\in I}\tau_i = \frac{d}{k}$. For $i \in I$, we have

$$\mu + \eta_i \geq \frac{1}{\alpha} \cdot (\tau_{\min} + \tau_i - \tau_{\min}) = v_i^\top Y^{-1} v_i$$

Therefore, the solution is dual feasible. This solution obtains the objective of $\frac{1}{d}\log\det(\alpha X) - 1 + \frac{k}{d}\mu + \frac{1}{d}\sum_{i=1}^n \eta_i$ which is equal to

$$\begin{aligned}
&= \frac{1}{d}\log\det(\alpha X) - 1 + \frac{k}{d}\frac{k}{\alpha(k-d)}\tau_{\min} + \frac{1}{\alpha d}\sum_{i\in I}(\tau_i - \tau_{\min}) \\
&= \frac{1}{d}\log\det(\alpha X) - 1 + \frac{k^2}{\alpha d(k-d)}\tau_{\min} + \frac{1}{\alpha d}(d - k\tau_{\min}) \\
&= \frac{1}{d}\log\det X + \log\alpha - 1 + \frac{1}{\alpha}\left(\frac{k}{k-d}\tau_{\min} + 1\right) \\
&\leq \frac{1}{d}\log\det X + \log\alpha - 1 + \frac{k}{\alpha(k-d)}
\end{aligned}$$

where the last inequality is by $\tau_{\min} \leq \frac{d}{k}$. Finally, we set $\alpha = \frac{k}{k-d}$ to obtain the objective value of dual

$$\frac{1}{d}\log\det(X) + \log\frac{k}{k-d} - 1 + 1 = \frac{1}{d}\log\det(X) + \log\frac{k}{k-d}$$

as required.

□

4.5 Proofs from Section 4.3

4.5.1 Proof of Performance of Modified Local Search Algorithm for A-DESIGN

Proof of Theorem 4.3.2

We first outline the proof of Theorem 4.3.2. Let (I, X) be the returned solution of the Algorithm 4.3. Observe that X is invertible since X is invertible at the beginning and $\mathrm{tr}(X^{-1})$ only decreases in the later iterations. Let $\tau_{ij} = u_i^\top X^{-1} u_j, h_{ij} = u_i^\top X^{-2} u_j, \tau_i = \tau_{ii}, h_i = h_{ii}$, and $\beta = tr(X^{-1})$. Since, X is a symmetric matrix, X^{-1} is also a symmetric matrix and therefore $\tau_{ij} = \tau_{ji}$ for each $i, j \in [n]$.

Notations For convenience, we restate the notations used in this section.

- V : Input to Modified Local Search Algorithm 4.4.
- $I^\star$: indices of the vectors in the optimal solution of A-DESIGN with input vector set V.
- $\phi^{\mathsf{A}}(V) = \operatorname{tr}\left(\left(\sum_{i \in I^\star} v_i v_i^\top\right)^{-1}\right)$.
- U : Output of Vector Capping Algorithm 4.2 and input to Local Search Algorithm with capped vectors 4.3.
- Δ : For every $i \in [1, n], ||u_i||_2^2 \leq \Delta$.
- (I, X) : Output of Local Search Algorithm with capped vectors 4.3 on input (U, k).
- $\phi_f^{\mathsf{A}}(U)$, and $\phi_f^{\mathsf{A}}(V)$ denote the (common) optimal value of objective values of the convex program with input vectors from V and U respectively.
- For $i, j \in [1, n], \tau_{ij} = u_i^\top X^{-1} u_j, h_{ij} = u_i^\top X^{-2} u_j$.
- For $i \in [n], \tau_i = \tau_{ii}, h_i = h_{ii}$.

Following lemma shows some standard connections between $\tau_{ij}, \tau_i, h_{ij}$ and h_i's. Proof of the lemma is presented in Section 4.5.1.

Lemma 4.5.1. *We have the following.*

1. *For any $i \in I$, we have $\tau_i \leq 1$. Moreover, for any $i \in I$, $\tau_i = 1$ if and only if $X - v_i v_i^\top$ is singular.*

2. *We have* $\sum_{i \in I} \tau_i = d$.

3. *For any* $i, j \in [n]$, $h_i(1 + \tau_j) - 2\tau_{ij}h_{ij} \geq 0$.

4. *For any* $j \in [n]$, *we have* $\sum_{i \in I} \tau_{ij}^2 = h_j$.

5. *We have* $\sum_{i \in I} h_i = \beta$.

6. *For any* $j \in [n]$, *we have* $\sum_{i \in I} \tau_{ij}h_{ij} = h_j$.

7. *For any* $j \in [n]$, *we have* $\tau_j \leq \sqrt{h_j}||u_j||_2$.

8. *For any* $i \in [n]$, *let* $X_{-i} = X - u_i u_i^\top$. *If* X_{-i} *is invertible, then for any* $j \in [n]$, *we have*

   - $u_j^\top X_{-i}^{-1} u_j = \frac{\tau_j + \tau_{ij}^2 - \tau_i \tau_j}{1 - \tau_i}$, *and*
   - $u_j^\top X_{-i}^{-2} u_j = h_j + \frac{h_i \tau_{ij}^2}{(1-\tau_i)^2} + \frac{2\tau_{ij}h_{ij}}{1-\tau_i}$.

Next lemma shows a lower bound on h_j in terms of β and $\phi_f^{\mathrm{A}}(U)$ by constructing a dual feasible solution.

Lemma 4.5.2. *We have* $\max_{j \in [n]} h_j \geq \frac{\beta^2}{k \cdot \phi_f^{\mathrm{A}}(U)}$.

Next lemma shows an upper bound on h_j in terms of β and τ_j using the local optimality condition.

Lemma 4.5.3. *For any* $j \in [n]$, $\frac{h_j}{1+\tau_j} \leq \frac{\beta}{k-d+2}$.

Before we prove these lemmas, we complete the proof of Theorem 4.3.2.

Theorem 4.3.2. By Lemma 4.5.3, for any $j \in [n]$, $\frac{h_j}{1+\tau_j} \leq \frac{\beta}{k-d+2}$. By Lemma 4.5.1, $\tau_j \leq \sqrt{h_j}||u_j||_2 \leq \sqrt{h_j \Delta}$. Hence, for any $j \in [n]$,

$$\frac{h_j}{1 + \sqrt{h_j \Delta}} \leq \frac{\beta}{k - d + 2}.$$

By Lemma 4.5.2, there exists $j \in [n]$ such that $h_j \geq \frac{\beta^2}{k \cdot \phi_f^{\mathrm{A}}(U)}$. Now we note the following claim.

Claim 8. *$f(x) = \frac{x}{1+c\sqrt{x}}$ is a monotonically increasing function for $x \geq 0$ if $c \geq 0$.*

Proof. $f'(x) = \frac{1}{1+c\sqrt{x}} + x \cdot \frac{-1}{(1+c\sqrt{x})^2} \cdot \frac{c}{2\sqrt{x}} = \frac{2+c\sqrt{x}}{(1+c\sqrt{x})^2}$ which is always positive for $x \geq 0$ if $c \geq 0$. □

Hence, we have

$$\frac{\frac{\beta^2}{k\cdot\phi_f^{\mathrm{A}}(U)}}{1+\sqrt{\frac{\beta^2}{k\cdot\phi_f^{\mathrm{A}}(U)}\Delta}} \leq \frac{\beta}{k-d+2}$$

$$\frac{k-d+2}{k}\frac{\beta}{\phi_f^{\mathrm{A}}(U)} \leq 1+\sqrt{\frac{\Delta\phi_f^{\mathrm{A}}(U)}{k}}\frac{\beta}{\phi_f^{\mathrm{A}}(U)}$$

$$\left(1-\frac{d-2}{k}-\sqrt{\frac{\Delta\phi_f^{\mathrm{A}}(U)}{k}}\right)\frac{\beta}{\phi_f^{\mathrm{A}}(U)} \leq 1$$

$$\mathrm{tr}(X^{-1}) = \beta \leq \phi_f^{\mathrm{A}}(U)\left(1-\frac{d-2}{k}-\sqrt{\frac{\Delta\phi_f^{\mathrm{A}}(U)}{k}}\right)^{-1}.$$

This finishes the proof of Theorem 4.3.2. □

Next, we prove Lemma 4.5.2 and Lemma 4.5.3.

Lemma 4.5.2. We prove the lemma by constructing a feasible solution to A-REL-DUAL(U). Let

$$Y = \gamma X^{-2}, \qquad \lambda = \max_{j\in[n]} u_j^\top Y u_j = \gamma \max_{j\in[n]} h_j$$

where $\gamma > 0$ will be fixed later. Then, (Y, λ) is a feasible solution to A-REL-DUAL(U). Hence,

$$\phi_f^{\mathrm{A}}(U) \geq 2\,\mathrm{tr}\left(\left(\gamma X^{-2}\right)^{1/2}\right) - k\gamma \max_{j\in[n]} h_j = 2\sqrt{\gamma}\beta - k\gamma \max_{j\in[n]} h_j.$$

Substituting $\gamma = \left(\frac{\beta}{k\max_{j\in[n]} h_j}\right)^2$, we get $\phi_f^{\mathrm{A}}(U) \geq \frac{\beta^2}{k\max_{j\in[n]} h_j}$. This gives us $\max_{j\in[n]} h_j \geq \frac{\beta^2}{k\phi_f^{\mathrm{A}}(U)}$ which is the desired inequality in Lemma 4.5.2. □

Lemma 4.5.3. We start the proof by showing an inequality implied by the local optimality of the solution.

Claim 9. *For any* $i \in I, j \in [n]$,

$$h_i(1+\tau_j) - h_j(1-\tau_i) - 2\tau_{ij}h_{ij} \geq 0 \tag{4.7}$$

Proof. For $i \in I$, let $X_{-i} = X - u_i u_i^\top$. First consider the case when X_{-i} is singular. From Lemma 4.5.1, $\tau_i = 1$ and $h_i(1+\tau_j) - 2\tau_{ij}h_{ij} \geq 0$. Hence,

$$h_i(1+\tau_j) - h_j(1-\tau_i) - 2\tau_{ij}h_{ij} \geq 0.$$

Now, consider the case when X_{-i} is non-singular. By local optimality condition, we have that for any $i \in I, j \in [n]$,

$$\beta \leq \operatorname{tr}\left(\left(X_{-i} + u_j u_j^\top\right)^{-1}\right)$$

By Sherman-Morrison formula,

$$\operatorname{tr}\left(\left(X_{-i} + u_j u_j^\top\right)^{-1}\right) = \operatorname{tr}(X_{-i}^{-1}) - \frac{u_j^\top X_{-i}^{-2} u_j}{1 + u_j^\top X_{-i} u_j} = \operatorname{tr}(X^{-1}) + \frac{u_i^\top X^{-2} u_i}{1 - u_i^\top X^{-1} u_i} - \frac{u_j^\top X_{-i}^{-2} u_j}{1 + u_j^\top X_{-i} u_j}$$

Hence, local optimality of I implies that for any $i \in I, j \in [n]$,

$$\beta \leq \operatorname{tr}(X^{-1}) + \frac{u_i^\top X^{-2} u_i}{1 - u_i^\top X^{-1} u_i} - \frac{u_j^\top X_{-i}^{-2} u_j}{1 + u_j^\top X_{-i} u_j} \tag{4.8}$$

By Lemma 4.5.1, we have $u_j^\top X_{-i}^{-1} u_j = \frac{\tau_j + \tau_{ij}^2 - \tau_i\tau_j}{1-\tau_i}$ and $u_j^\top X_{-i}^{-2} u_j = h_j + \frac{h_i \tau_{ij}^2}{(1-\tau_i)^2} + \frac{2\tau_{ij}h_{ij}}{1-\tau_i}$.

Substituting these and $\mathrm{tr}(X^{-1}) = \beta, u_j^\top X^{-2} u_j = h_j$, and $u_j^\top X^{-1} u_j = \tau_j$ in equation (4.8), we get

$$\begin{aligned}
\beta &\leq \beta + \frac{h_i}{1-\tau_i} - \frac{h_j + \frac{h_i \tau_{ij}^2}{(1-\tau_i)^2} + \frac{2\tau_{ij} h_{ij}}{1-\tau_i}}{1 + \frac{\tau_j + \tau_{ij}^2 - \tau_i \tau_j}{1-\tau_i}} \\
0 &\leq \frac{h_i}{1-\tau_i} - \frac{h_j(1-\tau_i)^2 + h_i \tau_{ij}^2 + 2(1-\tau_i)\tau_{ij} h_{ij}}{(1-\tau_i)(1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j)} \\
0 &\leq \frac{h_i}{1-\tau_i} - \frac{h_i \tau_{ij}^2}{(1-\tau_i)(1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j)} - \frac{h_j(1-\tau_i)^2 + 2(1-\tau_i)\tau_{ij} h_{ij}}{(1-\tau_i)(1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j)} \\
0 &\leq \frac{h_i(1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j-\tau_{ij}^2)}{(1-\tau_i)(1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j)} - \frac{h_j(1-\tau_i) + 2\tau_{ij} h_{ij}}{1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j} \\
0 &\leq \frac{h_i(1+\tau_j)}{1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j} - \frac{h_j(1-\tau_i) + 2\tau_{ij} h_{ij}}{1-\tau_i+\tau_j+\tau_{ij}^2-\tau_i\tau_j} \\
0 &\leq h_i(1+\tau_j) - h_j(1-\tau_i) - 2\tau_{ij} h_{ij}
\end{aligned}$$

Last inequality follows from the fact that $1-\tau_i+\tau_j-\tau_i\tau_j+\tau_{ij}^2 = (1-\tau_i)(1+\tau_j)+\tau_{ij}^2 > 0$ which follows from the fact that $\tau_i < 1$ (Lemma 4.5.1 and X_{-i} is invertible). This concludes the proof of claim 9. □

Next, we sum up equation (4.7) from claim 9 for all $i \in Z$ and get

$$(1+\tau_j)\sum_{i\in I} h_i - h_j(|I| - \sum_{i\in I}\tau_i) - 2\sum_{i\in I}\tau_{ij} h_{ij} \geq 0$$

By Lemma 4.5.1, $\sum_{i\in I} h_i = \beta, \sum_{i\in I}\tau_i = d$, and $\sum_{i\in I}\tau_{ij} h_{ij} = h_j$. We also know that $|I| = k$ throughout the algorithm. Substituting these in the equation above we get, $(1+\tau_j)\beta - h_j(k-d) - 2h_j \geq 0$ or equivalently,

$$\frac{h_j}{1+\tau_j} \leq \frac{\beta}{k-d+2}.$$

This finishes the proof of Lemma 4.5.3. □

The Capping Algorithm and the Proof of Lemma 4.3.1

Some intuition of the capping algorithm. Section 4.3.3 shows an example where local search outputs a solution with very large cost, thus showing that local search does not provide any approximation algorithm. The failure of local search algorithm is the presence of extremely long vectors ($||v||_2^2$ much larger than A-optimum) which leads to “skewed" eigenvectors and eigenvalues. Moreover, we were able to show that this is the only bottleneck. That is, if all vector norms are small (compared to A-optimum), solution output by the local search algorithm has cost at most $(1+\epsilon)$ times the fractional optimum.

The capping algorithm should then satisfy the following(s): Given an instance with arbitrary length vectors, output a new instance such that

1. All vectors in the new instance have small length

2. Fractional optimum of the new instance does not increase by more than $1+\epsilon$ factor of the old fractional optimum

3. Any integral solution in the new instance can be translated into an integral solution in the old instance with the same or lower cost.

If we can get such a procedure, we run the local search on the new instance and get an integral solution with cost at most $(1+\epsilon)$ times the fractional optimum of the new solution. Combining with the properties above, we can then get an integral solution in the old instance with cost at most $(1+\epsilon)^2$ of the old fractional optimum.

We note that a more natural capping algorithm where we pick the longest vector, scale this vector down, and project all other vectors into the space orthogonal to the large vector satisfies properties (1) and (2) but not (3). That is, given an integral solution in the new instance, we can not always find an integral solution in the old instance with roughly the same cost.

We now proof of Lemma 4.3.1, which says that our capping algorithm satisfies three properties we want.

Lemma 4.3.1. For ease of notation, we consider the equivalent algorithm of Algorithm 4.2.

Algorithm 4.5 Capping vectors length for A-DESIGN

```
Input: $V = \{v_1, \dots, v_n\} \subseteq \mathbb{R}^d$, parameter $\Delta$.
For $i \in [1, n], w_i^0 := v_i, \ell = 0$.
While$\exists i \in [1, n], ||w_i^l||_2^2 > \Delta$:
    $t_\ell = \mathrm{argmax}_{i \in [1,n]} ||w_i^l||_2$.
    % For all vectors, scale the component along with $w_t$ direction.
    For $j \in [1, n], w_j^{\ell+1} = \left( I_d - \frac{1}{2} \frac{w_{t_\ell}^\ell (w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2} \right) w_j^\ell$
    $\ell = \ell + 1$.
For $j \in [1, n], u_j = w_j^\ell$.
Return $U = \{u_1, \dots, u_n\} \subseteq \mathbb{R}^d$
```

First observe that the length of the largest vector reduces by a constant factor and length of any vector does not increase. Thus the algorithm ends in a finite number of iterations. Observe that the first property is trivially true when the algorithm returns a solution. For the second property, we show that the objective value of any set S only increases over the iterations. In particular, we show the following claim.

Claim 10. *For any set $S \subset [n]$ and any $\ell \geq 0$,*

$$\mathrm{tr}\left(\left(\sum_{i \in S} w_i^\ell (w_i^\ell)^\top\right)^{-1}\right) \leq \mathrm{tr}\left(\left(\sum_{i \in S} w_i^{\ell+1} (w_i^{\ell+1})^\top\right)^{-1}\right)$$

Proof. Let $Z = \left(I_{d\times d} - \frac{1}{2}\frac{w_{t_\ell}^\ell (w_{t_\ell}^\ell)^\top}{||w_{t_\ell}^\ell||_2^2}\right)$

$$\begin{aligned}
\operatorname{tr}\left(\left(\sum_{i\in S} w_i^{\ell+1}(w_i^{\ell+1})^\top\right)^{-1}\right) &= \operatorname{tr}\left(\left(Z\sum_{i\in S} w_i^{\ell}(w_i^{\ell})^\top Z^\top\right)^{-1}\right)\\
&= \operatorname{tr}\left(Z^{-1}\left(\sum_{i\in S} w_i^{\ell}(w_i^{\ell})^\top\right)^{-1} Z^{-1}\right)\\
&= \left\langle Z^{-2}, \left(\sum_{i\in S} w_i^{\ell}(w_i^{\ell})^\top\right)^{-1}\right\rangle
\end{aligned}$$

Observe that Z has all eigenvalues 1 except for one which is $\frac{1}{2}$. Thus Z^{-1} and Z^{-2} have all eigenvalues at least one and in particular $Z^{-2} \succeq I$. Hence,

$$\operatorname{tr}\left(\left(\sum_{i\in S} w_i^{\ell+1}(w_i^{\ell+1})^\top\right)^{-1}\right) \geq \operatorname{tr}\left(\left(\sum_{i\in S} w_i^{\ell}(w_i^{\ell})^\top\right)^{-1}\right)$$

as required. □

To prove the last property, we aim to obtain a recursion on the objective value of the convex program over the iterations. Let $W^\ell = \{w_1^\ell, \ldots, w_n^\ell\}$ be the set of vectors at the end of ℓ^{th} iteration and let $\alpha_\ell^\star = \phi_f^{\mathsf{A}}(W^\ell)$ denote the objective value of the convex program with the vectors obtained at the end of ℓ^{th} iteration. We divide the iterations in to epochs where in each epoch the length of the maximum vector drops by a factor of 2. For ease of notation, we let $p = 0$ be the last epoch and $p = 1$ to be the second last epoch and so on. For any integer $p \geq 0$, we let $r_p :=$ argmin$_\ell \max_{i\in[n]} ||w_i^\ell||_2^2 \leq 2^p \cdot \Delta$ be the last iteration of p^{th} epoch. Thus in the p^{th} epoch the length of the largest vector is in the interval $[2^p\cdot\Delta, 2^{p+1}\cdot\Delta)$. Let T denote the first epoch and thus $r_T = 0$. Next lemma bounds the increase in the relaxation value in each iteration. The bound depends on which epoch does the iteration lies in.

Lemma 4.5.4. *For every $\ell \in [r_p, r_{p-1})$, we have*

$$\alpha^\star_{\ell+1} \le \left(1 + \frac{2^{-3p/4}}{k}\right)\left(\alpha^\star_l + \frac{8}{2^{p/4}\Delta}\right).$$

Next lemma bounds the number of iterations in the p^{th} epoch.

Lemma 4.5.5. *For every $p \ge 1$, we have $r_{p-1} - r_p + 1 \le \frac{8}{3}d$.*

We first see the proof of last claim of Lemma 4.3.1 using Lemma 4.5.4 and Lemma 4.5.5 and then prove these lemmas.

Using Lemmas 4.5.4 and 4.5.5, we bound the increase in relaxation value in each epoch.

Claim 11. *For every $p \ge 1$, we have*

$$\alpha^\star_{r_{p-1}} \le \left(1 + \frac{2^{-3p/4}}{k}\right)^{\frac{8}{3}d}\left(\alpha^\star_{r_p} + \frac{64d}{3 \cdot 2^{p/4}\Delta}\right).$$

Proof. From Lemma 4.5.4, we have

$$\begin{aligned}
\alpha^\star_{r_{p-1}} &\le \left(1 + \frac{2^{-3p/4}}{k}\right)^{r_{p-1}-r_p+1} \alpha^\star_{r_p} + \frac{8}{2^{p/4}\Delta}\left(\sum_{i=1}^{r_{p-1}-r_p+1}\left(1 + \frac{2^{-3p/4}}{k}\right)^i\right) \\
&\le \left(1 + \frac{2^{-3p/4}}{k}\right)^{r_{p-1}-r_p+1}\left(\alpha^\star_{r_p} + \frac{8}{2^{p/4}\Delta}(r_{p-1} - r_p + 1)\right) \\
&\le \left(1 + \frac{2^{-3p/4}}{k}\right)^{r_{p-1}-r_p+1}\left(\alpha^\star_{r_p} + \frac{8}{2^{p/4}\Delta}(r_{p-1} - r_p + 1)\right) \\
&\le \left(1 + \frac{2^{-3p/4}}{k}\right)^{\frac{8}{3}d}\left(\alpha^\star_{r_p} + \frac{64d}{3 \cdot 2^{p/4}\Delta}\right) \qquad \text{(Lemma 4.5.5)}
\end{aligned}$$

as required. □

Solving the recurrence in Claim 11, we get a bound on the total increase in the relaxation cost throughout the algorithm.

$$\begin{aligned}\alpha^{\star}_{r_0} &\leq \left(\Pi_{p=0}^{\top}\left(1+\frac{2^{-3p/4}}{k}\right)^{\frac{8}{3}d}\right)\left(\alpha^{\star}_{r_T}+\sum_{p=0}^{T}\frac{64d}{3\cdot 2^{p/4}\Delta}\right)\\ &\leq \left(\Pi_{p=0}^{\top}\left(1+\frac{2^{-3p/4}}{k}\right)\right)^{\frac{8}{3}d}\left(\alpha^{\star}_{r_T}+\frac{2^{1/4}}{2^{1/4}-1}\frac{64d}{3\Delta}\right)\\ &\leq \left(\Pi_{p=0}^{\top}\left(1+\frac{2^{-p/2}}{k}\right)\right)^{\frac{8}{3}d}\left(\alpha^{\star}_{r_T}+\frac{135d}{\Delta}\right) \qquad (4.9)\end{aligned}$$

Claim 12. *For any* $k \geq 15$*,*

$$\Pi_{p=0}^{\infty}\left(1+\frac{2^{-3p/4}}{k}\right) \leq 1+\frac{3}{k}.$$

Proof.

$$\begin{aligned}\Pi_{p=0}^{\infty}\left(1+\frac{2^{-3p/4}}{k}\right) &= 1+\frac{1}{k}\sum_{p=0}^{\infty}2^{-3p/4}+\frac{1}{k^2}\sum_{p_1=0}^{\infty}\sum_{p_2=0}^{\infty}2^{-3p_1/4}2^{-3p_2/4}\\ &\quad+\frac{1}{k^3}\sum_{p_1=0}^{\infty}\sum_{p_2=0}^{\infty}\sum_{p_3=0}^{\infty}2^{-3p_1/4-3p_2/4-3p_3/4}\dots\\ &= 1+\frac{\sum_{p=0}^{\infty}2^{-3p/4}}{k}+\left(\frac{\sum_{p=0}^{\infty}2^{-3p/4}}{k}\right)^2+\left(\frac{\sum_{p=0}^{\infty}2^{-3p/4}}{k}\right)^3+\dots\\ &\leq 1+\frac{2.47}{k}+\left(\frac{2.47}{k}\right)^2+\left(\frac{2.47}{k}\right)^3+\dots\\ &= \frac{1}{1-2.47/k}\\ &\leq 1+\frac{3}{k}\end{aligned}$$

Last inequality follows since $k \geq 15$. □

Substituting bound from claim 12 in Equation (4.9), we get

$$\alpha^{\star}_{r_0} \leq \left(1+\frac{3}{k}\right)^{\frac{8}{3}d}\left(\alpha^{\star}_{r_T}+\frac{135d}{k}\right) \leq \left(1+e^{8}\frac{d}{k}\right)\left(\alpha^{\star}_{r_T}+\frac{135d}{k}\right)$$

Last inequality follows from the fact that $(1+a/x)^y \leq 1 + e^a \frac{x}{y}$ if $x > y > 0$ and $a \geq 1$.

By definition, $r_T = 0$. Hence, $\alpha_0^\star = \alpha_{r_T}^\star = \phi_f^{\mathrm{A}}(V)$. Also, by definition $\alpha_{r_0}^\star = \phi_f^{\mathrm{A}}(U)$. Hence,

$$\phi_f^{\mathrm{A}}(U) \leq \left(1 + e^8 \frac{d}{k}\right)\left(\phi_f^{\mathrm{A}}(V) + \frac{135d}{\Delta}\right) \leq \left(1 + 3000\frac{d}{k}\right)\left(\phi_f^{\mathrm{A}}(V) + 135\frac{d}{\Delta}\right).$$

This finishes the proof of Lemma 4.3.1. □

To complete the missing details in the proof of Lemma 4.3.1, we now prove Lemmas 4.5.4 and 4.5.5.

Lemma 4.5.4. For simplicity of exposition, we make some simplifying assumptions. Without loss of generality, we assume that $t_\ell = 1$, i.e., the longest vector is the first vector in this iteration. Also, since trace is invariant under rotation of basis, we may assume that $w_1^\ell = \sqrt{\gamma} e_1$ for some non-negative number γ where $e_1 = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}^\top$ is the first standard vector. Hence,

$$w_j^{\ell+1} = \left(I_{d\times d} - \frac{1}{2} e_1 e_1^\top\right) w_j^\ell.$$

Since, w_1^ℓ is the largest vector in this iteration and $\ell \in [r_p, r_{p-1})$, we have

$$2^p \Delta \geq \gamma > 2^{p-1}\Delta. \tag{4.10}$$

Let $\mathbf{x}$ be the optimal solution for $A\text{-}\textsc{Rel}(w_1^\ell, \dots, w_n^\ell)$. We construct a feasible solution $\mathbf{y}$ for $A\text{-}\textsc{Rel}(w_1^{\ell+1}, \dots, w_n^{\ell+1})$ with objective at most as required in the lemma. Let $\delta \geq 0$ be a constant that will be fixed later. Let

$$y_i = \begin{cases} \frac{k}{k+\delta}(\delta + x_1) & i = 1 \\ \frac{k}{k+\delta} x_i & i \in [2, n] \end{cases}$$

Claim 13. *$\mathbf{y}$ is a feasible solution to $A\text{-}\textsc{Rel}(w_1^{\ell+1}, \dots, w_n^{\ell+1})$.*

Proof. Since, **x** is a feasible solution of A-$\textsc{Rel}(w_1^\ell, \ldots, w_n^\ell)$, we know that $\sum_{i=1}^n x_i \leq k$. Thus

$$\sum_{i=1}^n y_i = \frac{k}{k+\delta}\delta + \frac{k}{k+\delta}\sum_{i=1}^n x_i \leq \frac{k}{k+\delta}\delta + \frac{k}{k+\delta}k \leq k.$$

Clearly **y** ≥ 0 and thus it is feasible. □

Now we bound the objective value of the solution **y**. Let

$$X = \sum_{i=1}^n x_i w_i^\ell (w_i^\ell)^\top, Y = \sum_{i=1}^n y_i w_i^{\ell+1} (w_i^{\ell+1})^\top.$$

Claim 14. *For any* $\delta > 0$, $\mathrm{tr}(Y^{-1}) \leq \frac{k+\delta}{k}\left(\mathrm{tr}(X^{-1}) + \frac{4}{\delta\gamma}\right)$.

Before we prove Claim 14, we complete the proof of Lemma 11.

From Equation (4.10), we have $\gamma \geq 2^{p-1}\Delta$ and substituting $\delta = 2^{-p/2}$ in Claim 14 we get,

$$\mathrm{tr}(Y^{-1}) \leq \left(1 + \frac{2^{-p/2}}{k}\right)\left(\mathrm{tr}(X^{-1}) + \frac{8}{2^{p/2}\Delta}\right).$$

Since, **x** is an optimal solution to A-$\textsc{Rel}(w_1^\ell, \ldots, w_n^\ell)$, we have $\alpha_\ell^\star = \phi_f^{\mathrm{A}}(w_1^\ell, \ldots, w_n^\ell) = \mathrm{tr}(X^{-1})$. Moreover, since y is a feasible solution to A-$\textsc{Rel}(w_1^{\ell+1}, \ldots, w_n^{\ell+1})$, we have

$$\alpha_{\ell+1}^\star = \phi_f^{\mathrm{A}}(w_1^{\ell+1}, \ldots, w_n^{\ell+1}) \leq \mathrm{tr}(Y^{-1}) \leq \left(1 + \frac{2^{-p/2}}{k}\right)\left(\alpha_\ell^\star + \frac{8}{2^{p/2}\Delta}\right).$$

Hence, it only remains to show the proof of Claim 14.

Claim 14. Let $X = \sum_{i=1}^{n} x_i w_i^{\ell}(w_i^{\ell})^\top = \begin{bmatrix} p & \bar{q}^\top \\ \bar{q} & R \end{bmatrix}$ where $p \in \mathbb{R}, \bar{q} \in \mathbb{R}^d, R \in \mathbb{R}^{d-1 \times d-1}$. Then

$$\begin{aligned}
\frac{k+\delta}{k} Y &= \delta w_1^{\ell+1}(w_1^{\ell+1})^\top + \sum_{i=1}^{n} x_i w_i^{\ell+1}(w_i^{\ell+1})^\top \\
&= \left(I_{d\times d} - \frac{1}{2} e_1 e_1^\top\right) \left(\delta w_1^{\ell}(w_1^{\ell})^\top + \sum_{i=1}^{n} w_i^{\ell}(w_i^{\ell})^\top\right) \left(I_{d\times d} - \frac{1}{2} e_1 e_1^\top\right)^\top \\
&= \begin{bmatrix} \frac{1}{2} & \bar{0}^\top \\ \bar{0} & I_{(d-1)\times(d-1)} \end{bmatrix} \begin{bmatrix} p + \delta\gamma & \bar{q}^\top \\ \bar{q} & R \end{bmatrix} \begin{bmatrix} \frac{1}{2} & \bar{0}^\top \\ \bar{0} & I_{(d-1)\times(d-1)} \end{bmatrix} \\
&= \begin{bmatrix} \frac{1}{4}(p + \delta\gamma) & \frac{1}{2}\bar{q}^\top \\ \frac{1}{2}\bar{q} & R \end{bmatrix}
\end{aligned}$$

Since X is positive definite, we must have $p > 0$, R is also positive definite and more over $p - \bar{q}^\top R^{-1} \bar{q} > 0$ (see Proposition 2.8.4 [Ber05]).

Fact 4.5.6. *(Block Inversion formula) For $A \in \mathbb{R}^{a\times a}, D \in \mathbb{R}^{d\times d}, B \in \mathbb{R}^{a\times d}, C \in \mathbb{R}^{d\times a}$ such that $\begin{bmatrix} A & B \\ C & D \end{bmatrix}$ is invertible, we have*

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1} = \begin{bmatrix} (A - BD^{-1}C)^{-1} & -(A - BD^{-1}C)^{-1}BD^{-1} \\ -(D - CA^{-1}B)^{-1}CA^{-1} & (D - CA^{-1}B)^{-1} \end{bmatrix}$$

Applying block inversion formula on X, we get

$$X^{-1} = \begin{bmatrix} \frac{1}{p - \bar{q}^\top R^{-1} \bar{q}} & \dots \\ \dots & \left(R - \frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1} \end{bmatrix}$$

Since, X is a positive semi-definite matrix, X^{-1} is also a positive semi-definite matrix. Hence, principle submatrices are positive semidefinite. In particular,

$$p - \bar{q}^\top R^{-1} \bar{q} \geq 0. \tag{4.11}$$

and,

$$R - \frac{1}{p}\bar{q}\bar{q}^\top \succeq 0_{(d-1)\times(d-1)} \tag{4.12}$$

Next, let us compute $\mathrm{tr}(X^{-1})$.

$$\mathrm{tr}(X^{-1}) = \frac{1}{p - \bar{q}^\top R^{-1}\bar{q}} + \mathrm{tr}\left(\left(R - \frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1}\right) \geq \mathrm{tr}\left(\left(R - \frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1}\right). \tag{4.13}$$

Applying block-inversion formula to $\frac{k+\delta}{k}Y$, we get

$$\left(\frac{k+\delta}{k}Y\right)^{-1} = \begin{bmatrix} \left(\frac{1}{4}(p+\delta\gamma) - \frac{1}{4}\bar{q}^\top R^{-1}\bar{q}\right)^{-1} & \cdots \\ \cdots & \left(R - \frac{1}{(p+\delta\gamma)/4}\frac{1}{4}\bar{q}\bar{q}^\top\right)^{-1} \end{bmatrix}$$

Hence,

$$\frac{k}{k+\delta}\,\mathrm{tr}\left(Y^{-1}\right) = \frac{4}{\delta\gamma + p - \bar{q}^\top R^{-1}\bar{q}} + \mathrm{tr}\left(\left(R - \frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top\right)^{-1}\right)$$

Claim 15.

$$\frac{4}{\delta\gamma + p - \bar{q}^\top R^{-1}\bar{q}} \leq \frac{4}{\delta\gamma}$$

Proof. By Equation (4.11), $p - \bar{q}^\top R^{-1}\bar{q} \geq 0$. Hence, the inequality trivially follows. □

Claim 16.

$$\mathrm{tr}\left(\left(R - \frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top\right)^{-1}\right) \leq \mathrm{tr}\left(\left(R - \frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1}\right)$$

Proof. Since, $\delta, \alpha \geq 0$, $\frac{1}{p+\delta\gamma} \leq \frac{1}{p}$. Hence,

$$\begin{aligned}
\frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top &\preceq \frac{1}{p}\bar{q}\bar{q}^\top \\
-\frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top &\succeq -\frac{1}{p}\bar{q}\bar{q}^\top \\
R - \frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top &\succeq R - \frac{1}{p}\bar{q}\bar{q}^\top \\
\left(R - \frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top\right)^{-1} &\preceq \left(R - \frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1} \\
\mathrm{tr}\left(\left(R - \frac{1}{p+\delta\gamma}\bar{q}\bar{q}^\top\right)^{-1}\right) &\leq \mathrm{tr}\left(\left(R - \frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1}\right)
\end{aligned}$$

□

Applying the above two claims, we get

$$\begin{aligned}
\frac{k}{k+\delta}\,\mathrm{tr}\left(Y^{-1}\right) &\leq \frac{10^4}{\delta\gamma} + \mathrm{tr}\left(\left(R - \frac{1}{p}\bar{q}\bar{q}^\top\right)^{-1}\right) \\
\frac{k}{k+\delta}\,\mathrm{tr}\left(Y^{-1}\right) &\leq \frac{10^4}{\delta\gamma} + \mathrm{tr}(X^{-1}) \qquad\qquad (eq\ (4.13)) \\
\mathrm{tr}(Y^{-1}) &\leq \frac{k+\delta}{k}\left(\mathrm{tr}(X^{-1}) + \frac{10^4}{\delta\gamma}\right).
\end{aligned}$$

This finishes the proof of Claim 14. □

Proof of Claim 14 also finishes the proof of Lemma 4.5.4.

□

Proof. (Lemma 4.5.5) By definition of r_p and r_{p-1}, we know that for any $\ell \in [r_p, r_{p-1})$,

$$2^{p-1}\Delta \leq \max_{i\in[n]} ||w_i^\ell||_2^2 \leq 2^p\Delta$$

Let $M_{r_p} = I_{d\times d}, R_{r_p} = I_{d\times d}$ and for $\ell \in [r_p, r_{p-1})$, let

$$M_{\ell+1} = \left(I_{d\times d} - \frac{1}{2}\frac{w^\ell_{t_\ell}(w^\ell_{t_\ell})^\top}{||w^\ell_{t_\ell}||_2^2}\right)M_\ell, \qquad R_{\ell+1} = M_{\ell+1}^T M_{\ell+1}.$$

For $\ell \in [r_p, r_{p-1})$, consider the potential function $\mathrm{tr}(R_\ell)$. We show the following properties about this potential function:

Claim 17. *Let M_ℓ, R_ℓ be as defined above for $\ell \in [r_p, r_{p-1})$. Then, $\mathrm{tr}(R_{r_p}) = d$ and for $\ell \in [r_p, r_{p-1})$,*

- $\mathrm{tr}(R_\ell) \geq 0$*, and*
- $\mathrm{tr}(R_{\ell+1}) \leq \mathrm{tr}(R_\ell) - \frac{3}{8}$.

Using Claim 17, it is easy to see that $r_{p-1} - r_p + 1 \leq \frac{8}{3}d$. Hence, to prove Lemma 4.5.5, it is enough to prove Claim 17.

Proof. (Claim 17) Since, $R_{r_p} = I_{d\times d}$, $\mathrm{tr}(R_{r_p}) = d$ is trivially true. Also, for any $\ell \in [r_p, r_{p-1})$, $R_\ell = M_\ell^\top M_\ell$ which is positive semidefinite. Hence, $\mathrm{tr}(R_\ell) \geq 0$ for any $\ell \in [r_p, r_{p-1})$. For $\ell \in [r_p, r_{p-1})$,

$$R_{\ell+1} = M_{\ell+1}^\top M_{\ell+1} = M_\ell^\top \left(I_{d\times d} - \frac{1}{2}\frac{w^\ell_{t_\ell}(w^\ell_{t_\ell})^\top}{||w^\ell_{t_\ell}||_2^2}\right)^\top \left(I_{d\times d} - \frac{1}{2}\frac{w^\ell_{t_\ell}(w^\ell_{t_\ell})^\top}{||w^\ell_{t_\ell}||_2^2}\right)M_\ell$$

Matrix $\left(I_{d\times d} - \frac{1}{2}\frac{w^\ell_{t_\ell}(w^\ell_{t_\ell})^\top}{||w^\ell_{t_\ell}||_2^2}\right)$ is symmetric. Hence,

$$\begin{aligned} R_{\ell+1} &= M_\ell^\top \left(I_{d\times d} - \frac{w^\ell_{t_\ell}(w^\ell_{t_\ell})^\top}{||w^\ell_{t_\ell}||_2^2} + \frac{1}{4}\frac{w^\ell_{t_\ell}(w^\ell_{t_\ell})^\top}{||w^\ell_{t_\ell}||_2^2}\frac{w^\ell_{t_\ell}(w^\ell_{t_\ell})^\top}{||w^\ell_{t_\ell}||_2^2}\right)M_\ell \\ &= M_\ell^\top \left(I_{d\times d} - \frac{w^\ell_{t_\ell}(w^\ell_{t_\ell})^\top}{||w^\ell_{t_\ell}||_2^2} + \frac{1}{4}\frac{w^\ell_{t_\ell}(w^\ell_{t_\ell})^\top}{||w^\ell_{t_\ell}||_2^2}\right)M_\ell \\ &= M_\ell^\top M_\ell - \frac{3}{4}\frac{(M_\ell^\top w^\ell_{t_\ell})(w^\ell_{t_\ell})^\top M_\ell}{||w^\ell_{t_\ell}||_2^2} = R_\ell - \frac{3}{4}\frac{(M_\ell^\top w^\ell_{t_\ell})(M_\ell^\top w^\ell_{t_\ell})^\top}{||w^\ell_{t_\ell}||_2^2} \end{aligned}$$

By definition $w_{t_\ell}^{\ell} = M_\ell w_{t_\ell}^{r_p}$. Hence,

$$R_{\ell+1} = R_\ell - \frac{3}{4}\frac{(M_\ell^\top M_\ell w_{t_\ell}^{r_p})(M_\ell^\top M_\ell w_{t_\ell}^{r_p})^\top}{||w_{t_\ell}^{\ell}||_2^2} = R_\ell - \frac{3}{4}\frac{(R_\ell w_{t_\ell}^{r_p})(R_\ell w_{t_\ell}^{r_p})^\top}{||w_{t_\ell}^{\ell}||_2^2}$$

And the trace is

$$\operatorname{tr}(R_{\ell+1}) = \operatorname{tr}\left(R_\ell - \frac{3}{4}\frac{(R_\ell w_{t_\ell}^{r_p})(R_\ell w_{t_\ell}^{r_p})^\top}{||w_{t_\ell}^{\ell}||_2^2}\right) = \operatorname{tr}(R_\ell) - \frac{3}{4}\frac{||R_\ell w_{t_\ell}^{r_p}||_2^2}{||w_{t_\ell}^{\ell}||_2^2}$$

By Cauchy-Shwarz inequality, $||u||_2^2 \geq (v^T u)^2/||v||_2^2$. Substituting $u = R_\ell w_{t_\ell}^{r_p}$ and $v = w_{t_\ell}^{r_p}$, we get

$$\begin{aligned}\operatorname{tr}(R_{\ell+1}) &\leq \operatorname{tr}(R_\ell) - \frac{3}{4}\frac{\left((w_{t_\ell}^{r_p})^\top R_\ell w_{t_\ell}^{r_p}\right)^2}{||w_{t_\ell}^{r_p}||_2^2 \cdot ||w_{t_\ell}^{\ell}||_2^2} = \operatorname{tr}(R_\ell) - \frac{3}{4}\frac{\left((w_{t_\ell}^{r_p})^\top M_\ell^\top M_\ell w_{t_\ell}^{r_p}\right)^2}{||w_{t_\ell}^{r_p}||_2^2 \cdot ||w_{t_\ell}^{\ell}||_2^2}\\ &= \operatorname{tr}(R_\ell) - \frac{3}{4}\frac{||M_\ell w_{t_\ell}^{r_p}||_2^4}{||w_{t_\ell}^{r_p}||_2^2||w_{t_\ell}^{\ell}||_2^2} = \operatorname{tr}(R_\ell) - \frac{3}{4}\frac{||w_{t_\ell}^{\ell}||_2^4}{||w_{t_\ell}^{r_p}||_2^2||w_{t_\ell}^{\ell}||_2^2}\\ &= \operatorname{tr}(R_\ell) - \frac{3}{4}\frac{||w_{t_\ell}^{\ell}||_2^2}{||w_{t_\ell}^{r_p}||_2^2}\end{aligned}$$

Since, $\ell \in [r_p, r_{p-1})$, $||w_{t_\ell}^{\ell}||_2^2 = \max_{i\in[n]} ||w_i^{\ell}||_2^2 \geq 2^{p-1}\Delta$. Also, by definition of r_p, $||w_{t_\ell}^{r_p}||_2^2 \leq \max_{i\in[n]} ||w_i^{r_p}||_2^2 \leq 2^p\Delta$. Hence,

$$\operatorname{tr}(R_{\ell+1}) \leq \operatorname{tr}(R_\ell) - \frac{3}{4}\frac{2^{p-1}\Delta}{2^p\Delta} = \operatorname{tr}(R_\ell) - \frac{3}{8}.$$

as desired. □

Hence, the proof of Lemma 4.5.5 is completed. □

Proof of Lemma 4.5.1

Lemma 4.5.1. Proof of first and second statement is same as that in Lemma 4.2.2. So, we start by proving that $h_i(1+\tau_j) - 2\tau_{ij}h_{ij} \geq 0$.

Claim 18. *For any $j \in [n]$, $X^{-1/2} u_j u_j^\top X^{-1/2} \preceq \tau_j I_d$.*

Proof. Since, X is a symmetric matrix, X^{-1} and $X^{-1/2}$ are also symmetric matrices. Hence, if $q = X^{-1/2} u_j$, then $X^{-1/2} u_j u_j^\top X^{-1/2} = qq^\top$. Such a matrix has one non-zero eigenvalue equal to $||q||_2^2 = u_j^\top X^{-1} u_j = \tau_j$. Hence, $X^{-1/2} u_j u_j^\top X^{-1/2} \preceq \tau_j I_d$. □

Next, we use this to derive further inequalities.

$$\begin{aligned}
X^{-1/2} u_j u_j^\top X^{-1/2} &\preceq \tau_j I_d \\
2X^{-1/2} u_j u_j^\top X^{-1/2} &\preceq 2\tau_j I_d \\
2X^{-1/2} u_j u_j^\top X^{-1/2} &\preceq (1+\tau_j) I_d & (\tau_j \leq 1, j \in [n]) \\
X^{-1/2} X^{-1/2} u_j u_j^\top X^{-1/2} X^{-3/2} &\preceq X^{-1/2} (1+\tau_j) X^{-3/2} & (X^{-1/2}, X^{-3/2} \text{ are PSD}) \\
2X^{-1} u_j u_j^\top X^{-2} &\preceq (1+\tau_j) X^{-2}
\end{aligned}$$

If $A \preceq B$, then $v^\top A v \leq v^\top B v$ for all v. Hence, $u_i^\top (2X^{-1} u_j u_j^\top X^{-2} \leq (1+\tau_j) X^{-2}) u_i \leq 0$. Or in other words, $h_i(1+\tau_i) - 2\tau_{ij} h_{ij} \geq 0$.

Next, we show that $\sum_{i \in I} \tau_{ij}^2 = h_j$.

$$\begin{aligned}
\sum_{i \in I} \tau_{ij}^2 &= \sum_{i \in I} u_i^\top X^{-1} u_j u_i^\top X^{-1} u_j = \sum_{i \in I} u_i^\top X^{-1} u_j u_j^\top X^{-1} u_i \\
&= \sum_{i \in u} \langle X^{-1} u_j u_j^\top X^{-1}, u_i u_i^\top \rangle \\
&= \langle X^{-1} u_j u_j^\top X^{-1}, \sum_{i \in Z} u_i u_i^\top \rangle \\
&= \langle X^{-1} u_j u_j^\top X^{-1}, X \rangle \\
&= \langle u_j^\top X^{-1}, X X^{-1} u_j^\top \rangle \\
&= \langle u_j^\top X^{-1}, u_j^\top \rangle = u_j^\top X^{-1} u_j = h_j
\end{aligned}$$

Next, we show that $\sum_{i\in I} h_i = \beta$.

$$\begin{aligned}\sum_{i\in I} h_i &= \sum_{i\in Z} u_i^\top X^{-2} u_i \\ &= \sum_{i\in I} \langle X^{-2}, u_i u_i^\top \rangle \\ &= \langle X^{-2}, \sum_{i\in I} u_i u_i^\top \rangle = \langle X^{-2}, X \rangle \\ &= \langle X^{-1}, X^{-1} X \rangle \\ &= \langle X^{-1}, I_d \rangle = \operatorname{tr}(X^{-1})\end{aligned}$$

Next, we show that $\sum_{i\in I} \tau_{ij} h_{ij} = h_j$.

$$\begin{aligned}\sum_{i\in I} \tau_{ij} h_{ij} &= \sum_{i\in I} u_i^\top X^{-1} u_j u_i^\top X^{-2} u_j = \sum_{i\in I} u_i^\top X^{-1} u_j u_j^\top X^{-2} u_i \\ &= \sum_{i\in I} \langle X^{-1} u_j u_j^\top X^{-2}, u_i u_i^\top \rangle \\ &= \langle X^{-1} u_j u_j^\top X^{-2}, \sum_{i\in Z} u_i u_i^\top \rangle = \langle X^{-1} u_j u_j^\top X^{-2}, X \rangle \\ &= \langle u_j^\top X^{-2}, u_j^\top X^{-1} X \rangle \\ &= \langle u_j^\top X^{-2}, u_j \rangle = h_j\end{aligned}$$

Next, we show that $\tau_j \le \sqrt{h_j} ||u_j||_2$.

$$\begin{aligned}\sqrt{h_j} ||u_j||_2 &= \sqrt{u_j^\top X^{-2} u_j} ||u_j||_2 \\ &= \sqrt{||X^{-1} u_j||_2^2} ||u_j||_2 = ||X^{-1} u_j||_2 ||u_j||_2 \\ &\ge u_j^\top X^{-1} u_j = \tau_j.\end{aligned}$$

Here, the last inequality follows from Cauchy-Schwarz inequality: for any $u, v \in \mathbb{R}^d$, $u^\top v \le$

$||u||_2||v||_2$.

Next, we show the last two equalities. For $i \in [n]$, $X_{-i} = X - u_i u_i^\top$. Let $j \in [n]$. By Sherman-Morrison formula,

$$X_{-i}^{-1} = X^{-1} + \frac{X^{-1}u_i u_i^\top X^{-1}}{1 - u_i^\top X^{-1} u_i} = X^{-1} + \frac{X^{-1}u_i u_i^\top X^{-1}}{1 - \tau_i} \tag{4.14}$$

Hence,

$$\begin{aligned} u_j^\top X_{-i}^{-1} u_j &= u_j^\top X^{-1} u_j + \frac{u_j^\top X^{-1} u_i u_i^\top X^{-1} u_j}{1 - \tau_i} \\ &= \tau_j + \frac{u_j^\top X^{-1} u_i u_i^\top X^{-1} u_j}{1 - \tau_i} \\ &= \tau_j + \frac{\tau_{ij} \cdot \tau_{ij}}{1 - \tau_i} = \frac{\tau_j + \tau_{ij}^2 - \tau_i \tau_j}{1 - \tau_i} \end{aligned}$$

Squaring the terms in equation (4.14), we get

$$\begin{aligned} X_{-i}^{-2} &= X^{-2} + \frac{X^{-1}u_i u_i^\top X^{-2} u_i u_i^\top X^{-1}}{(1 - \tau_i)^2} + \frac{X^{-1}u_i u_i^\top X^{-2}}{1 - \tau_i} + \frac{X^{-2}u_i u_i^\top X^{-1}}{1 - \tau_i} \\ &= X^{-2} + h_i \frac{X^{-1}u_i u_i^\top X^{-1}}{(1 - \tau_i)^2} + \frac{X^{-1}u_i u_i^\top X^{-2}}{1 - \tau_i} + \frac{X^{-2}u_i u_i^\top X^{-1}}{1 - \tau_i} \end{aligned}$$

Hence,

$$\begin{aligned} u_j^\top X_{-i}^{-2} u_j &= u_j^\top X^{-2} u_j + h_i \frac{u_j^\top X^{-1} u_i u_i^\top X^{-1} u_j}{(1 - \tau_i)^2} + \frac{u_j^\top X^{-1} u_i u_i^\top X^{-2} u_j}{1 - \tau_i} + \frac{u_j^\top X^{-2} u_i u_i^\top X^{-1} u_j}{1 - \tau_i} \\ &= h_j + h_i \frac{\tau_{ij} \cdot \tau_{ij}}{(1 - \tau_i)^2} + \frac{\tau_{ij} h_{ij}}{1 - \tau_i} + \frac{h_{ij} \tau_{ij}}{1 - \tau_i} \\ &= h_j + \frac{h_i \tau_{ij}^2}{(1 - \tau_i)^2} + \frac{2\tau_{ij} h_{ij}}{1 - \tau_i} \end{aligned}$$

□

4.5.2 Guessing A-Optimum Value $\phi^{\text{A}}(V)$

We remarked earlier that Algorithm 4.4 requires the knowledge of the optimum solution value $\phi^{\text{A}}(V)$. We can guess this value efficiently by performing a binary search. We explain the details and the proof of the polynomial runtime of the search in this section.

Let $\alpha = \operatorname{tr}\left(\left(\sum_{i=1}^n v_i v_i^T\right)^{-1}\right)$. Since we may pick at most k copies of each vector, we have that $\phi^{\text{A}}(V) \geq \operatorname{tr}\left(\left(k\sum_{i=1}^n v_i v_i^T\right)^{-1}\right) = \frac{1}{k}\alpha$. The fractional solution $x_i = \frac{k}{n}$ is feasible for A-$\textsc{Rel}(V)$. Hence, $\phi_f^{\text{A}}(V) \leq \operatorname{tr}\left(\left(\frac{k}{n}\sum_{i=1}^n v_i v_i^T\right)^{-1}\right) = \frac{n}{k}\alpha$. Using the result in [ALSW17b], we get that $\phi^{\text{A}}(V) \leq (1+\epsilon)\phi_f^{\text{A}}(V)$. Hence, $\phi^{\text{A}}(V) \in \left[\frac{1}{k}\alpha, \frac{n(1+\epsilon)}{k}\alpha)\right]$. Hence, given an instance, we first compute α and then perform a binary search for $\phi^{\text{A}}(V)$ in the interval $[\frac{1}{k}\alpha, \frac{n(1+\epsilon)}{k}\alpha]$.

Suppose the current range of the optimum is $[\ell, u]$. We guess OPT to be $\frac{\ell+u}{2}$ (use this as A-optimum $\phi^{\text{A}}(V)$) and run the modified local search algorithm. We claim that if it outputs a solution with cost at most $(1+\epsilon)\frac{\ell+u}{2}$ then $\phi^{\text{A}}(V)$ lies in the range $[\ell, (1+\epsilon)\frac{\ell+u}{2}]$. If it outputs a solution with cost more than $(1+\epsilon)\frac{\ell+u}{2}$, then $\phi^{\text{A}}(V)$ lies in the range $[\frac{\ell+u}{2}, u]$. The first statement is trivially true. The second statement is equivalent to the following: If $\phi^{\text{A}}(V)$ is less than $\frac{\ell+u}{2}$, then the algorithm outputs a solution of cost at most $(1+\epsilon)\frac{\ell+u}{2}$. Proof of this fact follows exactly the same way as the proof of Theorem 13 by substituting $\phi^{\text{A}}(V)$ with $\frac{\ell+u}{2}$ everywhere. The proof still follows, since the only place we use the meaning of the $\phi^{\text{A}}(V)$ value is in claiming that there exists a fractional solution with value $\phi^{\text{A}}(V)$. Because $\phi^{\text{A}}(V)$ is less than $\frac{\ell+u}{2}$, this statement is true with $\phi^{\text{A}}(V)$ replaced by $\frac{\ell+u}{2}$.

We can guess the value of $\phi^{\text{A}}(V)$ upto a factor of $1+\epsilon$ in $\log_{1+\epsilon}(n(1+\epsilon)) \leq \frac{\log(n(1+\epsilon))}{\epsilon}$ iterations. This introduces an additional multiplicative factor of $1+\epsilon$ in the approximation factor in Theorem 4.3.3. Hence, we get an approximation factor of $(1+\epsilon)(1+\epsilon) \leq (1+3\epsilon)$ and polynomial number of iterations.

4.5.3 Example of Instances to A-DESIGN

In this section, we give more details deferred from Section 4.3.3, starting with the proof of Theorem 4.3.4.

Theorem 4.3.4. The case $d = 2$ is proven in Lemma 4.3.5, so let $d \geq 3$. Let

$$v_1 = [1; \frac{1}{N^2}; 0; \ldots; 0], v_2 = [1; -\frac{1}{N^2}; 0; \ldots; 0], w_1 = [N^4; N; 0; \ldots; 0],$$

$$w_2 = [N^4; -N; 0; \ldots; 0], U = \left\{ u_i := \frac{1}{N^3} e_i : i = 3 \ldots, d \right\},$$

and let $\{v_1, v_2, w_1, w_2\} \cup U$ be the input vectors to A-DESIGN. Let $p = \lfloor \frac{k-d+2}{2} \rfloor, q = \lceil \frac{k-d+2}{2} \rceil$. Consider a solution S which picks p and q copies of v_1 and v_2, and one copy of u_i for each $i = 3, \ldots, d$. We claim that S is locally optimal.

Consider a swap of elements $S' = S \setminus \{s\} \cup \{s'\}$ where $s' \neq s$. If $s \in U$, then S' does not span full dimension. Hence, $s \in \{v_1, v_2\}$. If $s' = e_i \in U$ for some i, then the increase of eigenvalue of S' in the ith axis reduces the objective by $\Theta(N^3)$. However, by Claim 7, removing a vector s will increase the objective by $\Omega(N^4)$. Finally, if $s' \notin U$, then the swap appears within the first two dimension, so the calculation that a swap increases the objective is identical to the case $d = 2$, proven in Lemma 4.3.5. Therefore, S is locally optimal.

We now observe that the objective given by S is $\Theta(N^4)$, dominated by eigenvalues of eigenvectors spanning the first two dimension. However, consider a solution S^* which picks p and q copies of w_1 and w_2, and one copy of u_i for each $i = 3, \ldots, d$. The objective of S^* contributed by eigenvalues of eigenvectors lying in the first two dimension is $O(N^2)$ (Claim 7), so the total objective of S^* is $\Theta(N^3)$, which is arbitrarily smaller than $\Theta(N^4)$, the objective of S. □

We also remark that the exmple of input vectors to A-DESIGN given in this section also shows that A-DESIGN objective $S \rightarrow \operatorname{tr}\left(\left(\sum_{i \in S} v_i v_i^\top \right)^{-1} \right)$ is not supermodular, making the analysis of algorithms in submodular optimization unapplicable. A set function $g : 2^U \rightarrow \mathbb{R}$ is called submodular if $g(S \cup \{u\}) - g(S) \geq g(S' \cup \{u\}) - g(S')$ for all $S \subseteq S' \subseteq U$ and $u \in U$, and g is supermodular if $-g$ is submodular. In other words, g is supermodular if the marginal loss of g by adding u is decreasing as the set S is increasing by a partial ordering "$\subseteq$". As a set increases, the marginal loss of the A-DESIGN objective not only potentially increase, but also has no upper bound.

Remark 4.5.7. For any $d \geq 2, T > 0$, there exist sets of vectors $S \subsetneq S'$ in $\mathbb{R}^d$ and a vector $w \in \mathbb{R}^d$ such that

$$\frac{\operatorname{tr}\left(\left(\sum_{i\in S'} vv^\top\right)^{-1}\right) - \operatorname{tr}\left(\left(\sum_{i\in S'} vv^\top + ww^\top\right)^{-1}\right)}{\operatorname{tr}\left(\left(\sum_{i\in S} vv^\top\right)^{-1}\right) - \operatorname{tr}\left(\left(\sum_{i\in S} vv^\top + ww^\top\right)^{-1}\right)} > T$$

Proof. We first assume $d = 2$. Use the same definitions of vectors from Lemma 4.3.5 and set $S = \{v_1, v_2\}, S' = \{v_1, v_2, w_1\}$ and $w = w_2$. By Claim 7,

$$\operatorname{tr}\left(\left(\sum_{i\in S} vv^\top\right)^{-1}\right) - \operatorname{tr}\left(\left(\sum_{i\in S} vv^\top + ww^\top\right)^{-1}\right) = O(N)$$

and

$$\begin{aligned}\operatorname{tr}\left(\left(\sum_{i\in S'} vv^\top\right)^{-1}\right) - \operatorname{tr}\left(\left(\sum_{i\in S'} vv^\top + ww^\top\right)^{-1}\right) &\geq \operatorname{tr}\left(\left(\sum_{i\in S'} vv^\top\right)^{-1}\right) \\ &\quad - \operatorname{tr}\left(\left(w_1w_1^\top + w_2w_2^\top\right)^{-1}\right) \\ &= \Theta(N^4),\end{aligned}$$

so the proof is done because N tends to infinity. For the case $d \geq 3$, we may pad zeroes to all vectors in the above example and add a unit vector to S, S' to each of other $d - 2$ dimensions. $\square$

4.6 Approximate Local Search for D-DESIGN

While Theorem 4.2.1 proves a guarantee for every local optimum, it is not clear at all whether the local optimum solution can be obtained efficiently. Here we give a approximate local search algorithm that only makes improvements when they result in substantial reduction in the objective. We show that this algorithm is polynomial time as well results in essentially the same guarantee as Theorem 4.2.1.

Algorithm 4.6 Approximate Local search algorithm for D-DESIGN

```
Input: $V = v_1, \ldots, v_n \in \mathbb{R}^d, d \leq k \in n$, parameter $\delta > 0$.
Let $I$ be any (multi)-subset of $[1, n]$ of size $k$ such that $X = \sum_{i \in I} v_i v_i^\top$ is non-singular matrix.
While$\exists i \in I, j \in [1, n]$ such that $det\left(X - v_i v_i^\top + v_j v_j^\top\right) > (1 + \delta) \cdot \det(X)$:
    $X = X - v_i v_i^\top + v_j v_j^\top$
    $I = I \setminus \{i\} \cup \{j\}$
Return $(I, X)$
```

Recall that ϕ_f^{D} denote the be the common optimum value of (D-REL) and its dual (D-REL-DUAL). $I^\star$ denote the indices of the vector in the optimal solution and $\phi^{\mathrm{D}} = \det\left(\sum_{i \in I^\star} v_i v_i^\top\right)^{\frac{1}{d}}$ be its objective. We have $\phi_f^{\mathrm{D}} \geq \log \phi^{\mathrm{D}}$. We have the following result about Algorithm 4.6.

Theorem 4.6.1. *Let X be the solution returned by Algorithm 4.6. Then,*

$$det(X) \geq e^{-k\delta} \left(\frac{k - d + 1}{k}\right)^d e^{d \cdot \phi_f^{\mathrm{D}}}$$

and therefore,

$$\det(X)^{\frac{1}{d}} \geq e^{-\frac{k\delta}{d}} \frac{k - d + 1}{k} \cdot \phi^{\mathrm{D}}.$$

Moreover, the running time of the algorithm is polynomial in $n, d, k, \frac{1}{\delta}$ and the size of the input.

Proof of the theorem is analogous to the proof of Theorem 4.2.1. Let (I, X) be the returned solution of the algorithm. We also let V_I denote the $d \times |I|$ matrix whose columns are v_i for each $i \in I$. Observe that $X = V_I V_I^\top$ and X is invertible since $\det(X) > 0$ at the beginning of the iteration and it only increases in later iterations. We let $\tau_i = v_i^\top X^{-1} v_i$ for any $1 \le i \le n$. Observe that if $i \in I$, then τ_i is the leverage score of row v_i with respect to the matrix $V_I^\top$. We also $\tau_{ij} = v_i^\top X^{-1} v_j$ for any $1 \le i, j \le n$. As in Theorem 4.2.1, we have some properties regarding τ_i and h_i.

Lemma 4.6.2. *We have the following.*

1. *For any $i \in I$, we have $\tau_i \le 1$. Moreover, for any $i \in I$, $\tau_i = 1$ if and only if $X - v_i v_i^\top$ is singular.*

2. *We have $\sum_{i \in I} \tau_i = d$.*

3. *For any $1 \le j \le n$, we have $\sum_{i \in I} \tau_{ij} \tau_{ji} = \tau_j$.*

4. *For any $1 \le i, j \le n$, we have $\tau_{ij} = \tau_{ji}$ and $\tau_{ij} \le \sqrt{\tau_i \tau_j}$.*

Proof of the lemma is identical to that of Lemma 4.2.2. Next, we show an upper bound on τ_j for the approximate local optimal solution.

Lemma 4.6.3. *For any $j \in [1, n]$,*

$$\tau_j \le \frac{d + \delta k}{k - d + 1}.$$

Before we prove the lemma, we complete the proof of Theorem 4.6.1.

Proof. [Theorem 4.6.1] We construct a feasible solution to the (D-REL-DUAL) of the objective value of at most $\frac{1}{d} \log\det(X) + \log \frac{k}{k-d+1} + \frac{k\delta}{d}$. This would imply that

$$O_f^\star \le \frac{1}{d} \log\det(X) + \log \frac{k}{k-d+1} + \frac{k\delta}{d}$$

which proves the first part of the theorem. The second part follows since $\phi_f^{\mathsf{D}} \ge \log \phi^{\mathsf{D}}$.

Let

$$Y = \alpha X, \qquad \mu = \max_{1 \le j \le n} v_j^\top Y^{-1} v_j = \frac{1}{\alpha} \max_{j \in [1,n]} v_j^\top X^{-1} v_j$$

where $\alpha > 0$ will be fixed later. Then, (Y, μ) is a feasible solution of (D-REL-DUAL). Hence,

$$\begin{aligned}\phi_f^{\mathsf{D}} &\le \frac{1}{d} \log\det(\alpha X) + \frac{k}{d} \cdot \frac{1}{\alpha} \max_{j \in [1,n]} v_j^\top X^{-1} v_j - 1 \\ &\le \log \alpha + \frac{1}{d} \log\det(X) + \frac{k}{d\alpha} \cdot \frac{d + k\delta}{k - d + 1} - 1 \qquad \text{(Lemma 4.6.3)}\end{aligned}$$

Setting $\alpha = \frac{k}{k-d+1}$, we get

$$\phi_f^{\mathsf{D}} \le \log \frac{k}{k - d + 1} + \frac{1}{d} \log\det(X) + 1 + \frac{k\delta}{d} - 1 = \log \frac{k}{k - d + 1} + \frac{1}{d} \log\det(X) + \frac{k\delta}{d}$$

as required. □

Lemma 4.6.3. Since X is a symmetric matrix, X^{-1} is also a symmetric matrix and therefore $\tau_{ij} = \tau_{ji}$ for each i, j. We first show that the approximate local optimality condition implies the following claim:

Claim 19. *For any $i \in I$ and $j \in [n]$, we have*

$$\tau_j - \tau_i \tau_j + \tau_{ij} \tau_{ji} \le \delta + \tau_i. \tag{4.15}$$

Proof. Let $i \in I$, $j \in [n]$ and $X_{-i} = X - v_i v_i^\top$. First, consider the case when X_{-i} is singular. From Lemma 4.2.2, we have that $\tau_i = 1, \tau_{ij} = \tau_{ji} \le \sqrt{\tau_i \tau_j} \le 1$. Hence,

$$\tau_j - \tau_i \tau_j + \tau_{ij} \tau_{ji} \le \tau_j - \tau_j + 1 = \tau_i \le \delta + \tau_i$$

Now consider the case when X_{-i} is non-singular. By local optimality of I, we get that

$$\det\left(X_{-i} + v_j v_j^\top\right) \leq (1+\delta)\det\left(X_{-i} + v_i v_i^\top\right) \tag{4.16}$$

Claim 20. *For any invertible matrix $A \in \mathbb{R}^{d\times d}$ and $v \in \mathbb{R}^d$,*

$$\det(A + vv^\top) = \det(A)(1 + v^\top A^{-1} v)$$

Hence, local optimality of I implies that for any $i \in I, j \in [n]$,

$$\det(X_{-i})(1 + v_j^\top X_{-i}^{-1} v_j) \leq (1+\delta)\det(X_{-i})(1 + v_i^\top X_{-i}^{-1} v_i)$$

Dividing both sides by $\det\left(X_{-i}\right)$, we get for each $i \in I$ and $j \in [n]$, we have $1 + v_j^\top X_{-i}^{-1} v_j \leq (1+\delta)(1 + v_i^\top X_{-i}^{-1} v_i)$ or equivalently,

$$v_j^\top X_{-i}^{-1} v_j \leq \delta + (1+\delta) v_i^\top X_{-i}^{-1} v_i.$$

From the Sherman-Morrison Formula we obtain that for any $i \in I$ and $j \in [n]$, we have

$$v_j^\top \left(X^{-1} + \frac{X^{-1} v_i v_i^\top X^{-1}}{1 - v_i^\top X^{-1} v_i}\right) v_j \leq \delta + (1+\delta) v_i^\top \left(X^{-1} + \frac{X^{-1} v_i v_i^\top X^{-1}}{1 - v_i^\top X^{-1} v_i}\right) v_i.$$

Now using the definition of τ_i, τ_j and τ_{ij}, we obtain that for any $i \in I$ and $1 \leq j \leq n$, we have

$$\tau_j + \frac{\tau_{ji}\tau_{ij}}{1 - \tau_i} \leq \delta + (1+\delta)\left(\tau_i + \frac{\tau_i^2}{1-\tau_i}\right).$$

Multiplying by $1 - \tau_i$, which is positive from Lemma 4.2.2, on both sides we obtain that for any

$i \in I$ and $1 \leq j \leq n$,

$$\tau_j - \tau_i \tau_j + \tau_{ij}\tau_{ji} \leq \delta(1-\tau_i) + (1+\delta)\tau_i = \delta + \tau_i$$

thus finishing the proof of the claim. □

Now summing over the inequality in Claim 19 for all $i \in I$, we get

$$\sum_{i \in I} (\tau_j - \tau_i \tau_j + \tau_{ij}\tau_{ji}) \leq \sum_{i \in I} \delta + \sum_{i \in I} \tau_i.$$

Applying Lemma 4.2.2, we obtain that

$$k\tau_j - d\tau_j + \tau_j \leq \delta k + d.$$

Rearranging, we obtain that

$$\tau_j \leq \frac{d + \delta k}{k - d + 1}$$

□

Runtime Analysis. One may obtain the worst-case runtime for local search for D-design as follows. Let L be the maximum number of the length of binary string that encodes the number in each component across all input vectors v_i. Suppose we start with any solution S with nonzero determinant $\det(V_S V_S^T) = \sum_{R \subseteq S, |R|=d} \det(V_R V_R^T)$ (Cauchy-Binet), which can be done in polynomial time by finding a set of linearly independent vectors. Since $V_S V_S^\top$ is PSD, $\det(V_S V_S^T)$ is nonnegative and hence must be strictly positive, and therefore at least one term $\det(V_R V_R^T)$ is strictly positive. We now use the fact that for a square matrix A, the binary encoding length of $\det(A)$ is at most twice of the encoding length of matrix A (the exact definition of encoding length and the proof are in Theorem 3.2 of [Sch98]). Since the length of $d \times d$ matrix $V_R V_R^T$ is at most $2Ld^2 \log d$ (by bounds of values from direct matrix multiplication), the length of $\det(V_R V_R^T)$ is at most $4Ld^2 \log d$. Hence, the value of the determinant is at least $2^{-4Ld^2 \log d}$.

The optimum solution S^* of D-DESIGN attains the objective value $\sum_{R \subset S^*, |R|=d} \det(V_R V_R^T)$ (Cauchy-Binet). Each term $\det(V_R V_R^T)$ again has length at most $4Ld^2 \log d$, and so is at most $2^{4Ld^2 \log d}$. Therefore, the optimum is at most $\binom{k}{d} \cdot 2^{4Ld^2 \log d} \leq k^d 2^{4Ld^2 \log d}$. Hence, any solution S with nonzero determinant is a $k^d 2^{8Ld^2 \log d}$-approximation. Each swap increases the objective by a multiplicative factor $1 + \delta$, so the algorithm takes at most

$$\log_{1+\delta}(k^d 2^{8Ld^2 \log d}) \leq \frac{2}{\delta} \left(d \log k + 8Ld^2 \log d \right) = O\left(\frac{Ld^2 \log d + d \log k}{\delta} \right)$$

swapping steps, where we use $\frac{1}{\log(1+\delta)} \leq \frac{2}{\delta}$ for $\delta < 2$. We can use matrix determinant lemma (for rank-one update) to compute the new determinant objective rather than recomputing it in the next iteration. The matrix determinant lemma computation takes $O(d^2)$ times, so one swapping steps takes $O(knd^2)$ time by computing all kn potential pairs of swaps. Therefore, the local search in total takes $O\left(\frac{Lknd^3 \log d + knd^2 \log k}{\delta} \right)$ arithmetic operations.

4.7 Approximate Local Search for A-DESIGN

Algorithm 4.7 Approximate Local search algorithm for A-DESIGN

```
Input: $U = \{u_1, \dots, u_n\} \subseteq \mathbb{R}^d, d \leq k \in \mathbb{N}$.
Let $I$ be any (multi)-subset of $[1, n]$ of size $k$ such that $X = \sum_{i \in I} v_i v_i^\top$ is non-singular.
While$\exists i \in I, j \in [1, n]$ such that $\operatorname{tr}\left( (X - u_i u_i^\top + u_j u_j^\top)^{-1} \right) < (1 - \delta) \operatorname{tr}(X^{-1})$:
    $X = X - u_i u_i^\top + u_j u_j^\top$
    $I = I \setminus \{i\} \cup \{j\}$
Return $(I, X)$
```

Recall that for any input vectors $V = \{v_1, \dots, v_n\}$, the primal program is A-REL(V) and the dual program is A-REL-DUAL(V). We index these convex program by input vectors as we aim to analyze their objectives when the input changes by the capping algorithm. $\phi_f^{\mathsf{A}}(V)$ denote the (common) optimal value of objective values of the convex program with input vectors from V. $I^\star$ denote the indices of the vectors in the optimal solution of A-DESIGN with input vector set V and let $\phi^{\mathsf{A}}(V) = \operatorname{tr}\left(\left(\sum_{i \in I^\star} v_i v_i^\top\right)^{-1}\right)$ be its objective. Recall that $\phi_f^{\mathsf{A}}(V) \leq \phi^{\mathsf{A}}(V)$.

Similar to the local search result for A-DESIGN of Theorem 4.3.2, we can prove the following theorem:

Theorem 4.7.1. *Let X be the matrix returned by Algorithm 4.7. If $||u_i||_2^2 \leq \Delta$ for all $i \in [n]$,*

$$\operatorname{tr}(X^{-1}) \leq \phi_f^{\mathsf{A}}(U) \left(\left(1 - \frac{d-2}{k}\right) \frac{1}{1 + (k-d)\delta} - \sqrt{\frac{\Delta \phi_f^{\mathsf{A}}(U)}{k}} \right)^{-1}.$$

To prove Theorem 4.7.1, we can prove the following lemma instead of Lemma 4.5.3.

Lemma 4.7.2. *For any $j \in [n]$,*

$$\frac{h_j}{1 + \tau_j} \leq \frac{\beta(1 + (k-d)\delta)}{k - d + 2}$$

Instead of Theorem 4.3.3, Theorem 4.7.1 now leads to the following theorem:

Theorem 4.7.3. *For input vectors $V = \{v_1, \ldots, v_n\}$ and parameter k, let $U = \{u_1, \ldots, u_n\}$ be the set of vectors returned by the Capping Algorithm 4.2 with vector set V and $\Delta = \frac{d}{\epsilon^2 \phi^{\mathsf{A}}(V)}$. Let (I, X) be the solution returned by Algorithm 4.3 with vector set U and parameter k. If $k \geq \frac{2d}{\epsilon^4}$, $\delta \leq \frac{\epsilon d}{2k}$, and$\epsilon \leq 0.001$ then,*

$$\operatorname{tr}\left(\left(\sum_{i \in I} v_i v_i^\top \right)^{-1} \right) \leq (1 + 2\epsilon)\phi^{\mathsf{A}}(V).$$

Proof of the theorems and lemmas are identical to the corresponding theorems and lemmas proved in Section 4.3. Hence, we avoid the tedious calculations in reproving these theorems.

Runtime Analysis We claim that the running times of both capping and approximate local search for A-DESIGN are polynomial in $n, d, k, \frac{1}{\delta}$ and the size of the input. We first analyze approximate local search algorithm for A-DESIGN. Denote L the biggest bit complexity across any entries of any input vector v_i. We claim that the runtime complexity is identical to that of D-DESIGN. The analysis follows similarly to the runtime analysis of D-DESIGN, and we analyze here in detail for completeness.

We first want to bound the original objective and the optimum in k, d, L. By Cauchy-Binet,

$$E_{d-1}(V_S V_S^\top) = \sum_{R \subseteq S : |R| = d-1} \det(V_R^T V_R) \text{ and } \det(V_S V_S^\top) = \sum_{R \subseteq S : |R| = d} \det(V_R^T V_R)$$

Since the length of $V_R^T V_R$ for $|R| = d - 1$ is at most $2Ld^2 \log d$, we have that if $E_{d-1}(V_S V_S^\top)$ is strictly positive, then $E_{d-1}(V_S V_S^\top) \geq 2^{-4Ld^2 \log d}$ ([Sch98]), for at least one term $\det(V_R^T V_R)$ is strictly positive. We also have $E_{d-1}(V_S V_S^\top) \leq \binom{k}{d-1} \cdot 2^{4Ld^2 \log d}$ for any S of size k. The same is true for $\det$ and R of size k in place of E_{d-1} and R of size $d-1$. Therefore, an optimal set S^* satisfies

$$\frac{E_{d-1}(V_{S^*} V_{S^*}^\top)}{\det(V_{S^*} V_{S^*}^\top)} \geq \frac{2^{-4Ld^2 \log d}}{\binom{k}{d} \cdot 2^{4Ld^2 \log d}} \geq k^{-d} 2^{-8Ld^2 \log d}$$

and any initial set S with finite optimum satisfies

$$\frac{E_{d-1}(V_S V_S^\top)}{\det(V_S V_S^\top)} \leq \frac{\binom{k}{d-1} \cdot 2^{4Ld^2 \log d}}{2^{-4Ld^2 \log d}} \leq k^d 2^{8Ld^2 \log d}$$

Therefore, any initial solution to the local search algorithm with finite optimum is a $k^{2d} 2^{16Ld^2 \log d}$-approximation. Hence, the local search algorithm takes at most

$$\log_{1+\delta}(k^{2d} 2^{16Ld^2 \log d}) \leq \frac{2}{\delta} \left(2d \log k + 16Ld^2 \log d\right) = O\left(\frac{Ld^2 \log d + d \log k}{\delta}\right)$$

swapping steps. Similar to D-DESIGN, each swapping step takes $O(knd^2)$ time by matrix determinant lemma, so the total runtime is $O\left(\frac{Lknd^3\log d + knd^2\log k}{\delta}\right)$ number of arithmetic operations.

We now show that capping algorithm terminates in polynomial time. Again, let L be the maximum number of the length of binary string that encodes the number in each component across all input vectors v_i. Then $||v_i||^2 \leq \sqrt{d}\cdot 2^{2L}$ for all i's. In each iteration, the capping algorithm reduces the length of at least one vector by at least half, and hence by $n\log\frac{\sqrt{d}2^{2L}}{\Delta}$ iteration of capping, all vectors have length at most Δ. We show above that $\phi^{\mathsf{A}}(V) \leq k^d 2^{8Ld^2\log d}$ which, by $\Delta = \frac{d}{\epsilon^2\phi^{\mathsf{A}}(V)}$ of the capping algorithm, gives

$$n\log\frac{\sqrt{d}2^{2L}}{\Delta} \leq n\left(2L + \log d + d\log k + 8Ld^2\log d + \log\frac{\epsilon^2}{d}\right) = O\left(Ld^2\log d + d\log k\right)$$

where we use that ϵ is a small constant. Each step takes $O(nd)$ to compute n norms of d-dimensional vectors, and $O(d^2)$ for computing a $d\times d$ matrix and multiplying it with a vector for scaling operation. Therefore, the runtime of capping algorithm is $O\left(Lnd^3\log d + nd^2\log k\right)$.

Finally, we note that the input to local search algorithm are not the same as original input, which we assume with bit complexity L on each entry. However, by Lemma 4.3.1 which shows that the objective of capped vectors and original vectors are at most constant factors within each other, the gap between the initial objective (which is finite) and optimum changes by at most a constant factor, and hence the complexity of number of swaps remains unchanged. The total runtime of modified local search is therefore $O\left(\frac{Lknd^3\log d + knd^2\log k}{\delta}\right)$, dominated by the local search time complexity.

4.8 Greedy Algorithm for D-DESIGN

To prove Theorem 4.1.2, we again use the convex programming relaxation for the D-DESIGN problem. Recall the relaxation (D-REL) and its dual (D-REL-DUAL) shown in figure 2.2b. ϕ_f^{D} denote the be the common optimum value of (D-REL) and its dual (D-REL-DUAL). $I^\star$ denote the indices of the vector in the optimal solution and let $\phi^{\mathsf{D}} = \det\left(\sum_{i\in I^\star} v_i v_i^\top\right)^{\frac{1}{d}}$ be its objective. Observe that $\phi_f^{\mathsf{D}} \geq \log\phi^{\mathsf{D}}$. Now, Theorem 4.1.2 follows from the following theorem with an

Algorithm 4.8 Greedy algorithm for D-DESIGN

```
Input: $V = v_1, \dots, v_n \in \mathbb{R}^d, d \le k \in \mathbb{N}, S_0 \subset [n]$.
$X_0 = \sum_{j \in S_0} v_j v_j^\top$.
For $i = 1$ to $k - |S_0|$:
    $j_i = \mathrm{argmax}_{j \in [n]} \det(X + v_j v_j^\top)$
    $S_i = S_{i-1} \cup \{j_i\}, X_i = X_{i-1} + v_{j_i} v_{j_i}^\top$
$I = S_{k-|S_0|}, X = X_{k-|S_0|}$
Return $(I, X)$.
```

appropriate initialization of first d vectors which will be specified later.

Theorem 4.8.1. *For any set of vectors $v_1, \dots, v_n \in \mathbb{R}^d$, suppose $S_0 \subset [1, n]$ is a set of size d such that* $\det\left(\sum_{i \in S_0} v_i v_i^\top\right)^{\frac{1}{d}} > \frac{d}{k}\kappa \cdot \phi^{\mathsf{D}}$ *for some $\frac{1}{e} \ge \kappa > 0$ and $k \ge \frac{d}{\epsilon}\left(\log \frac{1}{\epsilon} + \log\log \frac{1}{\kappa}\right)$. Let (I, X) be the solution returned by Algorithm 4.8. Then,*

$$\det(X) \ge (1 - 5\epsilon)\phi^{\mathsf{D}}$$

Before we prove Theorem 4.8.1, we state and prove the following theorem, which better conveys main ideas of the proof.

Theorem 4.8.2. *For any set of vectors $v_1, \dots, v_n \in \mathbb{R}^d$ and $k \ge \frac{d \log \frac{1}{\epsilon}}{\epsilon}$, suppose $S_0 \subset [1, n]$ is a set of size d such that* $\det\left(\sum_{i \in S_0} v_i v_i^\top\right)^{\frac{1}{d}} > \frac{d}{k}\kappa \cdot \phi^{\mathsf{D}}$ *for some $1 > \kappa > 0$. Let $s = \max\{d \log\log \frac{1}{\kappa}, 0\}$ and (I, X) be the solution returned by picking $k - d + s$ vectors greedily. Then,*

$$\det(X) \ge (1 - 4\epsilon)\phi^{\mathsf{D}}$$

Theorem 4.8.2 gives a bi-criteria approximation where we pick small number s of extra vectors than the budget k while obtaining near-optimal solution. These s vectors are required to improve the initial approximation $\frac{d}{k}\kappa$ to a ratio $\frac{d}{k}$ independent of n or κ.

Theorem 4.8.2. To prove this theorem, we show the following two lemmas. First lemma shows the increase in the solution value in each greedy step.

Lemma 4.8.3. *For* $t \in [0, k - |S_0| - 1]$, $\det(X_{t+1}) \geq \det(X_t) \left(1 + \frac{d}{k} \frac{e^{\phi_f^{\mathrm{D}}}}{(\det(X_t))^{1/d}}\right)$

Next lemma shows that this recursion leads to the desired bound in the theorem.

Lemma 4.8.4. *Let* $\ell \geq 0$. *Let* $z_0, \ldots, z_{k-\ell}$ *be such that for* $t \in [0, k-\ell-1]$, $z_{t+1} \geq z_t \left(1 + \frac{d}{kz_t}\right)^{1/d}$. *Then,*

1. *If* $z_0 < \frac{d}{k}$*, then for any* $s \geq d \log\log \frac{dz_0}{k}$*, we have*

$$z_s \geq \frac{d}{ek}$$

2. *If* $z_0 \geq \frac{d}{ek}$*, then we have*

$$z_{k-\ell} \geq \frac{k-d-\ell}{k} - \frac{2d}{k} \log \frac{k}{d}$$

Proof of Theorem 4.8.2 follows from these two lemmas by defining $z_t = \frac{e^{\phi_f^{\mathrm{D}}}}{(\det(X_t))^{1/d}}$ in the bound in Lemma 4.8.3. Lemma 4.8.4 implies that for any initial κ approximation with d initial vectors to the D design problem of k vectors, $s = d \log\log \frac{1}{\kappa}$ vectors is enough to guarantee $\frac{d}{ek}$-approximation. Then, the second bound of Lemma 4.8.4 applies for the rest of the greedy algorithm. We now prove these two lemmas.

Lemma 4.8.3. By definition, $\det(X_{t+1}) = \max_{j\in[n]} \det(X_t + v_j v_j^\top)$. By Lemma 4.4.2, $\det(X_t + v_j v_j^\top) = \det(X_t)(1 + v_j^\top X_t^{-1} v_j)$. Hence,

$$\det(X_{t+1}) = \det(X_t) \left(1 + \max_{j\in[n]} v_j^\top X_t^{-1} v_j\right) \tag{4.17}$$

Next, we lower bound $\max_{j\in[n]} v_j^\top X_t^{-1} v_j$ by constructing a feasible solution to the (D-REL-DUAL). Let

$$Y = \alpha X_t, \qquad \mu = \max_{j\in[n]} v_j^\top Y^{-1} v_j = \frac{1}{\alpha} \max_{j\in[n]} v_j^\top X_t^{-1} v_j$$

where α will be fixed later. Then, (Y, μ) is a feasible solution of (D-REL-DUAL). Hence,

$$\phi_f^{\mathsf{D}} \leq \frac{1}{d} \log\det(\alpha X_t) + \frac{k}{d} \cdot \frac{1}{\alpha} \max_{j \in [n]} v_j^\top X_t^{-1} v_j - 1$$

which implies

$$\frac{d\alpha}{k} \left(\phi_f^{\mathsf{D}} + 1 - \log \alpha - \frac{1}{d} \log\det(X_t) \right) \leq \max_{j \in [n]} v_j^\top X_t^{-1} v_j$$

Setting, $\alpha = \frac{e^{\phi_f^{\mathsf{D}}}}{\det(X_t)^{1/d}}$, we get

$$\max_{j \in [n]} v_j^\top X_t^{-1} v_j \geq \frac{d}{k} \frac{e^{\phi_f^{\mathsf{D}}}}{\det(X_t)^{1/d}} \left(\phi_f^{\mathsf{D}} + 1 - \log \frac{e^{\phi_f^{\mathsf{D}}}}{\det(X_t)^{1/d}} - \frac{1}{d} \log\det(X_t) \right) = \frac{d}{k} \frac{e^{\phi_f^{\mathsf{D}}}}{\det(X_t)^{1/d}}$$

Substituting the bounds in equation (4.17), we get

$$\det(X_{t+1}) \geq \det(X_t) \left(1 + \frac{d}{k} \frac{e^{\phi_f^{\mathsf{D}}}}{(\det(X_t))^{1/d}} \right).$$

This finishes the proof of Lemma 4.8.3. □

Lemma 4.8.4. We first prove the first bound. The recursion implies that $\frac{z_{t+1}}{z_t} \geq \left(\frac{d}{kz_t} \right)^{\frac{1}{d}}$, which is equivalent to

$$\log z_{t+1} \geq \frac{1}{d} \log \frac{d}{k} + \frac{d-1}{d} \log z_t \tag{4.18}$$

Define $a_t := \log \frac{d}{k} - \log z_t$. If $a_u \leq 0$ for any $u \leq s$, then we are done because $z_s \geq z_u \geq \frac{d}{k}$. Else, we can rearrange terms to obtain

$$a_{t+1} \leq \left(1 - \frac{1}{d} \right) a_t \tag{4.19}$$

Hence, we have

$$\begin{aligned} a_s &\leq \left(1 - \frac{1}{d}\right)^s a_0 \\ &\leq e^{-\frac{s}{d}} a_0 \leq e^{-\frac{s}{d}} \log \frac{dz_0}{k} \\ &\leq 1 \end{aligned}$$

where the last inequality follows from $s \geq \log\log \frac{dz_0}{k}$. Therefore, $\log \frac{d}{k} - \log z_s = a_s \leq 1$, giving the desired bound.

To prove the second bound, the recursion is equivalent to

$$\log \frac{z_{t+1}}{z_t} \geq \frac{1}{d} \log \left(1 + \frac{d}{kz_t}\right) \tag{4.20}$$

It is clear that z_t is an increasing sequence in t, hence $\frac{d}{kz_t} \leq \frac{d}{kz_0} = e$. We use $\log(1+x) \geq \frac{x}{e}$ for $0 \leq x \leq e$ (by concavity of $\log\ x$) to lower bound the right-hand-side of (4.20) above inequality to obtain

$$\log \frac{z_{t+1}}{z_t} \geq \frac{1}{d} \cdot \frac{d}{ekz_t} = \frac{1}{ekz_t}$$

Thus, by using $e^x \geq 1 + x$, we have $\frac{z_{t+1}}{z_t} \geq e^{\frac{1}{ekz_t}} \geq 1 + \frac{1}{ekz_t}$, which implies

$$z_{t+1} \geq z_t + \frac{1}{ek}$$

Therefore, we obtain $z_t \geq \frac{t}{ek}$ for all $t \geq 0$.

Next, we apply the bound $\log(1+x) \geq x - \frac{x^2}{2} = x\left(1 - \frac{x}{2}\right)$ whenever $0 \leq x$ on the right-hand-side of (4.20) to obtain

$$\log \frac{z_{t+1}}{z_t} \geq \frac{1}{d} \frac{d}{kz_t} \left(1 - \frac{d}{2kz_t}\right) \geq \frac{1}{kz_t} \cdot \left(1 - \frac{2d}{t}\right)$$

where the last inequality comes from $z_t \geq \frac{t}{ek}$. Thus, applying $e^x \geq 1 + x$, we have $\frac{z_{t+1}}{z_t} \geq 1 + \frac{1}{kz_t} \cdot \left(1 - \frac{2d}{t}\right)$, which implies

$$z_{t+1} \geq z_t + \frac{1}{k} - \frac{2d}{tk} \tag{4.21}$$

Summing (4.21) from $t = d$ to $t = k - \ell - 1$ gives

$$\begin{aligned} z_{k-\ell} &\geq z_d + \frac{k-d-\ell-1}{k} - \frac{2d}{k}\left(\frac{1}{d} + \frac{1}{d+1} + \ldots + \frac{1}{k-\ell-1}\right) \\ &\geq \frac{k-d-\ell}{k} - \frac{2d}{k}\log\frac{k}{d} \end{aligned}$$

as desired. □

Now we prove Theorem 4.8.2. We first pick s vectors greedily to guarantee that $z_s \geq \frac{d}{ek}$. (If $z_0 > \frac{d}{ek}$, then $s = 0$.) Substituting $\ell = d$ and $k \geq \frac{d \log \frac{1}{\epsilon}}{\epsilon}$ in Lemma 4.8.4 gives

$$\begin{aligned} z_{k-\ell} &\geq 1 - \frac{d}{k}\left(2 + 2\log\frac{k}{d}\right) \\ &\geq 1 - \frac{2\epsilon}{\log\frac{1}{\epsilon}}\left(1 + \log\frac{1}{\epsilon} + \log\log\frac{1}{\epsilon}\right) \geq 1 - 4\epsilon \end{aligned}$$

where the second inequality follows from $\frac{1}{x}(1 + \log x)$ being decreasing function on $x \geq 1$, and the last inequality is by $1 + x \leq e^x$ with $x = \log\frac{1}{\epsilon}$. □

We are now ready to prove the main theorem.

Theorem 4.8.1. The proof is identical to the proof of Theorem 4.8.2 except that, after using $s = \log\log\frac{1}{\kappa}$ vectors to obtain $\frac{d}{ek}$-approximation, we only take $k - d - s$ greedy steps instead of $k - d$ greedy steps. Hence, we set $\ell = d + s$ to the second bound of Lemma 4.8.4 to obtain

$$z_{k-\ell} \geq \frac{k-2d-s}{k} - \frac{2d}{k}\log\frac{k}{d} = 1 - \frac{d}{k}\left(2 + 2\log\frac{k}{d}\right) - \frac{s}{k}$$

We have $1-\frac{d}{k}\left(2+2\log\frac{k}{d}\right)\geq 1-4\epsilon$ identical to the proof of Theorem 4.8.2. By $k\geq\frac{d}{\epsilon}\log\log\frac{1}{\kappa}=\frac{s}{\epsilon}$, we have $\frac{s}{k}\leq\epsilon$, completing the proof. □

We finally note on combinatorial algorithms for setting initial solution of size d. One may use volume sampling algorithms to achieve $\frac{n}{k}$-approximation to optimal objective in for picking d vectors [AB13]. Alternatively, we can perform local search on initial d vectors to obtain $d(1+\delta)$-approximation in time polynomial in $\frac{1}{\delta}$, as shown in Section 4.6. Since we know that the relaxation gaps of A- and D- optimal design are at most $\frac{k}{k-d+1}$, we can bound the optimum values of design problems between picking d and k vectors to be at most k multiplicative factor apart [AB13, NST19]. The approximation ratios of two algorithms are hence n and $dk(1+\delta)$, respectively. We formalize this argument and the result with locally optimal initial set as the following statement, which proves Theorem 4.1.2.

Corollary 4.8.5. *Greedy algorithm initialized by a local optimal set of size d returns a $(1+5\epsilon)$-approximation whenever $k\geq\frac{d}{\epsilon}(\log\frac{1}{\epsilon}+\log\log d+1)$.*

We first argue the ratio of optimum D-DESIGN values when the size of the set is d and k. Denote $\phi^{\mathsf{D}}(d),\phi^{\mathsf{D}}(k)=\phi^{\mathsf{D}}$ the optimum D-DESIGN objective $\det\left(\sum_{i\in S}v_iv_i^\top\right)^{\frac{1}{d}}$ on size d,k, respectively. Denote $\phi_f^{\mathsf{D}}(d),\phi_f^{\mathsf{D}}(k)=\phi_f^{\mathsf{D}}$ the common optimum value of (D-REL) and its dual (D-REL-DUAL) for size constraints of d,k respectively.

Claim 21. *We have*

$$\phi^{\mathsf{D}}(k)\leq k\phi^{\mathsf{D}}(d)$$

Proof. Because (D-REL) is a relaxation of D-DESIGN (up to $\log$ scale), we have

$$\exp\phi_f^{\mathsf{D}}(k)\geq\phi^{\mathsf{D}}(k),\qquad \exp\phi_f^{\mathsf{D}}(d)\geq\phi^{\mathsf{D}}(d)$$

We may scale any optimal solution of (D-REL) with size k to size d by applying $x_i:=\frac{d}{k}x_i$

coordinate-wise. Therefore, we have

$$\phi_f^{\mathsf{D}}(d) \geq \phi_f^{\mathsf{D}}(k) + \log \frac{d}{k}$$

Finally, we know that the integrality gap of (D-REL) is at most $\frac{k}{k-d+1}$. This follows from the approximation result of local search algorithm which compares the objective value of returned set to the objective to the convex relaxation. (This exact bound of the gap also follows from previous work on proportional volume sampling [NST19].) We apply this gap for size budget d to obtain

$$\exp \phi_f^{\mathsf{D}}(d) \leq d\phi^{\mathsf{D}}(d)$$

Therefore, we have

$$\phi^{\mathsf{D}}(k) \leq \exp \phi_f^{\mathsf{D}}(k) \leq \frac{k}{d} \exp \phi_f^{\mathsf{D}}(d) \leq k\phi^{\mathsf{D}}(d) \tag{4.22}$$

as desired. □

Corollary 4.8.5. Theorem 4.1.1 implies that a local search solution satisfies d-approximation when budget size is d. Hence, by Claim 21, a local solution is dk-approximation compared to D-DESIGN with a size budget of k.

We now apply Theorem 4.8.1: it is sufficient to show that

$$k \geq \frac{d}{\epsilon} \left(\log \frac{1}{\epsilon} + \log \log \frac{1}{\kappa} \right) \tag{4.23}$$

for $\kappa = \frac{1}{d^2}$, so the result follows. □

4.9 Greedy Algorithm for A-DESIGN

In this section, we prove Theorem 4.1.4. As remarked in the case of local search algorithm, we need to modify the instance to cap the length of the vectors in the case of greedy algorithm as well.

This is done by Algorithm 4.2. As shown in Lemma 4.3.1, the value of any feasible solution only increases after capping and the value of the convex programming relaxation increases by a small factor if k is large.

We now show that the greedy algorithm run on these vectors returns a near optimal solution. For any input vectors $V = \{v_1, \ldots, v_n\}$, the primal program is A-REL(V) and the dual program is A-REL-DUAL(V). $\phi_f^{\mathsf{A}}(V)$ denotes the (common) optimal value of objective values of the convex program with input vectors from V. $I^\star$ denotes the indices of the vectors in the optimal solution of A-DESIGN with input vector set V and $\phi^{\mathsf{A}}(V) = \operatorname{tr}\left(\left(\sum_{i\in I^\star} v_i v_i^\top\right)^{-1}\right)$ be its objective. We show the following theorem about Algorithm 4.9 in terms of capping length Δ.

Algorithm 4.9 Greedy algorithm for A-DESIGN

```
Input: $U = u_1, \ldots, u_n \in \mathbb{R}^d, d \leq k \in \mathbb{N}, S_0 \subset [n]$.
$X_0 = \sum_{j \in S_0} u_j u_j^\top$.
For $i = 1$ to $k - |S_0|$:
    $j_i = \operatorname{argmin}_{j\in[n]} \operatorname{tr}\left(\left(X + u_j u_j^\top\right)^{-1}\right)$
    $S_i = S_{i-1} \cup \{j_i\}, X_i = X_{i-1} + u_{j_i} u_{j_i}^\top$
$I = S_{k-|S_0|}, X = X_{k-|S_0|}$.
Return $(I, X)$.
```

Theorem 4.9.1. *Let $||u_i||_2^2 \leq \Delta$, $S_0 \subseteq [n]$ of size $r \geq d$ such that $\operatorname{tr}\left(\left(\sum_{i\in S_0} u_i u_i^\top\right)^{-1}\right) \leq \kappa \cdot \phi^{\mathsf{A}}(U)$ for some $\kappa \geq 1$, and $\Lambda = \sqrt{\frac{\Delta \phi_f^{\mathsf{A}}(U)}{k}}$. Let (I, X) be the solution returned by Algorithm 4.9. Then we have*

$$\operatorname{tr}(X^{-1}) \leq \left(1 - \frac{d+r}{k} - 2\Lambda \log \frac{k \max\{\Lambda\kappa, 1\}}{d}\right)^{-1} \phi^{\mathsf{A}}(U)$$

Similar to the analysis of local search for A-DESIGN, capping vector length is necessary to obtain theoretical guarantee. We will optimize over the length Δ later in Theorem 4.9.4.

Theorem 4.9.1. To prove the theorem, we show the following two lemmas:

Lemma 4.9.2. *For any $t \in [0, k-|S_0|]$, let $z_t = \operatorname{tr}(X_t^{-1})/\phi_f^{\mathrm{A}}(U)$. Then, for any $t \in [0, k-|S_0|-1]$,*

$$z_{t+1} \leq z_t \left(1 - \frac{z_t}{k\left(1 + z_t \sqrt{\frac{\Delta \phi_f^{\mathrm{A}}(U)}{k}}\right)} \right)$$

Lemma 4.9.3. *Let $\Lambda \geq 0$ and $\ell \geq 0$. Suppose $z_{t+1} \leq z_t \left(1 - \frac{z_t}{k(1+z_t\Lambda)}\right)$ for all $t \geq 0$, then*

1. *If $z_0 > \frac{1}{\Lambda}$, then for any $s \geq 2\Lambda k \log(\Lambda z_0)$, we have*

$$z_s \leq \frac{1}{\Lambda}$$

2. *If $z_0 \leq \frac{1}{\Lambda}$, we have*

$$z_{k-\ell} \leq \left(1 - \frac{d+\ell}{k} - 2\Lambda \log \frac{k}{d}\right)^{-1}$$

Lemma 4.9.2. By definition,

$$\operatorname{tr}(X_{t+1}^{-1}) = \min_{j\in[n]} \operatorname{tr}\left(\left(X_t + u_j u_j^\top\right)^{-1}\right).$$

By Sherman-Morrison formula,

$$\operatorname{tr}(X_{t+1}^{-1}) = \operatorname{tr}(X_t^{-1}) - \max_{j\in[n]} \frac{u_j^\top X_t^{-2} u_j}{1 + u_j^\top X_t^{-1} u_j}$$

Note that $u_j^\top X_t^{-1} u_j = \langle u_j, X_t^{-1} u_j \rangle$. By Cauchy-Schwarz inequality, $u_j^\top X_t^{-1} u_j$ is at most $||u_j||_2 ||X_t^{-1} u_j||_2 = ||u_j||_2 \sqrt{u_j^\top X_t^{-2} u_j}$. Since, $||u_j||_2^2 \leq \Delta$, we get $u_j^\top X_t^{-1} u_j \leq \sqrt{\Delta \cdot u_j^\top X_t^{-2} u_j}$. Hence,

$$\operatorname{tr}(X_{t+1}^{-1}) \leq \operatorname{tr}(X_t^{-1}) - \max_{j\in[n]} \frac{u_j^\top X_t^{-2} u_j}{1 + \sqrt{\Delta \cdot u_j^\top X_t^{-2} u_j}} \tag{4.24}$$

Next, we lower bound $\max_{j\in[n]} u_j^\top X_t^{-2} u_j$ by finding a feasible solution to A-REL-DUAL. Let,

$$Y = \gamma X_t^{-2}, \qquad \lambda = \max_{j\in[n]} u_j^\top Y u_j = \gamma \max_{j\in[n]} u_j^\top X_t^{-2} u_j$$

where $\gamma > 0$ will be fixed later. Then, (Y, λ) is a feasible solution to A-REL-DUAL(U). Hence,

$$\begin{aligned} \phi_f^{\mathsf{A}}(U) &\geq 2\operatorname{tr}\left(\left(\gamma X_t^{-2}\right)^{1/2}\right) - k\gamma \max_{j\in[n]} u_j^\top X_t^{-2} u_j \\ \max_{j\in[n]} u_j^\top X_t^{-2} u_j &\geq \frac{1}{k\gamma}\left(2\sqrt{\gamma}\operatorname{tr}(X_t^{-1}) - \phi_f^{\mathsf{A}}(U)\right) \end{aligned}$$

Substituting $\gamma = \left(\frac{\phi_f^{\mathsf{A}}(U)}{\operatorname{tr}(X_t^{-1})}\right)^2$, we get

$$\max_{j\in[n]} u_j^\top X_t^{-2} u_j \geq \frac{\operatorname{tr}(X_t^{-1})^2}{k\phi_f^{\mathsf{A}}(U)}.$$

As proved in Claim 8, $\frac{x}{1+c\sqrt{x}}$ is a monotonically increasing function for $x \geq 0$ if $c \geq 0$. Hence,

$$\max_{j\in[n]} \frac{u_j^\top X_t^{-2} u_j}{1 + \sqrt{\Delta \cdot u_j^\top X_t^{-2} u_j}} \geq \frac{\frac{\operatorname{tr}(X_t^{-1})^2}{k\phi_f^{\mathsf{A}}(U)}}{1 + \sqrt{\Delta \frac{\operatorname{tr}(X_t^{-1})^2}{k\phi_f^{\mathsf{A}}(U)}}}$$

Substituting $z_t = \frac{\operatorname{tr}(X_t^{-1})}{\phi_f^{\mathsf{A}}(U)}$, we get

$$\max_{j\in[n]} \frac{u_j^\top X_t^{-2} u_j}{1 + \sqrt{\Delta \cdot u_j^\top X_t^{-2} u_j}} \geq \frac{\operatorname{tr}(X_t^{-1})}{k} \frac{z_t}{1 + z_t\sqrt{\frac{\Delta\phi_f^{\mathsf{A}}(U)}{k}}}.$$

Substituting this inequality in Equation (4.24), we get

$$\operatorname{tr}(X_{t+1}^{-1}) \leq \operatorname{tr}(X_t^{-1}) \left(1 - \frac{z_t}{k\left(1 + z_t\sqrt{\frac{\Delta\phi_f^{\mathsf{A}}(U)}{k}}\right)}\right).$$

Substituting $z_t = \mathrm{tr}(X_t^{-1})/\phi_f^{\mathsf{A}}(U)$ and $z_{t+1} = \mathrm{tr}(X_{t+1}^{-1})/\phi_f^{\mathsf{A}}(U)$, we get

$$z_{t+1} \le z_t \left(1 - \frac{z_t}{k\left(1 + z_t\sqrt{\frac{\Delta\phi_f^{\mathsf{A}}(U)}{k}}\right)} \right).$$

This finishes the proof of Lemma 4.9.2. □

Lemma 4.9.3. We first prove the first bound. If $z_t \le \frac{1}{\Lambda}$ for any $t < s$, then we are done, so assume $z_t\Lambda \ge 1$. The recursion then implies

$$z_{t+1} \le z_t \left(1 - \frac{z_t}{k(2z_t\Lambda)}\right) = z_t\left(1 - \frac{1}{2k\Lambda}\right)$$

Therefore,

$$\begin{aligned} z_s &\le z_0\left(1 - \frac{1}{2k\Lambda}\right)^s \\ &\le z_0 e^{-\frac{1}{2\Lambda}s} \le z_0 e^{-\log\Lambda z_0} = \frac{1}{\Lambda} \end{aligned}$$

as desired.

We now prove the second bound. Let $a_t = \frac{1}{z_t}$. Then the recursion $z_{t+1} \le z_t\left(1 - \frac{z_t}{k(1+z_t\Lambda)}\right)$ can be rewritten as

$$\frac{a_{t+1}}{a_t} \ge \left(1 - \frac{1}{k\left(\Lambda + a_t\right)}\right)^{-1} \tag{4.25}$$

Applying $\left(1 - \frac{1}{k(\Lambda+a_t)}\right)^{-1} \ge 1 + \frac{1}{k(\Lambda+a_t)}$ and rearranging terms, we obtain

$$a_{t+1} \ge a_t + \frac{a_t}{k(\Lambda + a_t)} = a_t + \frac{1}{k} - \frac{\Lambda}{k(\Lambda + a_t)} \tag{4.26}$$

It is obvious from (4.25) that a_t is an increasing sequence, and hence $a_t \ge a_0 \ge \Lambda$ for all $t \ge 0$.

So (4.26) implies

$$a_{t+1} \geq a_t + \frac{1}{k} - \frac{\Lambda}{k(2\Lambda)} = a_t + \frac{1}{2k} \tag{4.27}$$

Therefore, we have $a_t \geq \frac{t}{2k}$ for all $t \geq 0$.

Using this bound $a_t \geq \frac{t}{2k}$, the recursion (4.26) also implies

$$a_{t+1} \geq a_t + \frac{1}{k} - \frac{\Lambda}{k(\frac{t}{2k})} = a_t + \frac{1}{k} - \frac{2\Lambda}{t} \tag{4.28}$$

Summing 4.28 from $t = d$ to $t = k - \ell - 1$ gives

$$\begin{aligned} a_{k-\ell} &\geq a_d + \frac{k-d-\ell}{k} - 2\Lambda \sum_{t=d}^{k-\ell-1} \frac{1}{t} \\ &= \frac{k-d-\ell}{k} - 2\Lambda \log \frac{k}{d} \end{aligned}$$

proving the desired bound. □

We now prove Theorem 4.9.1. The first bound of Lemma 4.9.3 shows that with initial approximation κ, we require $s = \max\{0, 2\Lambda k \log(\Lambda\kappa)\}$ steps to ensure $\frac{1}{\Lambda}$ approximation ratio. After that, we can pick $k - r - s$ vectors. Hence, we apply the second bound of Lemma 4.9.3 with $\ell = r + s$ to get the approximation ratio of X as

$$\begin{aligned} z_{k-\ell} &\leq \left(1 - \frac{d+r+s}{k} - 2\Lambda \log \frac{k}{d}\right)^{-1} \\ &= \left(1 - \frac{d+r}{k} - 2\Lambda \left(\log \frac{k}{d} + \max\left\{\log \Lambda\kappa, 0\right\}\right)\right)^{-1} \\ &= \left(1 - \frac{d+r}{k} - 2\Lambda \log \frac{k \max\{\Lambda\kappa, 1\}}{d}\right)^{-1} \end{aligned}$$

proving the desired bound. □

Next, we tune Δ in Theorem 4.9.1 and use Lemma 4.3.1 to obtain the final bound, from which

Theorem 4.1.4 will follow.

Theorem 4.9.4. *For input vectors $V = \{v_1, \dots, v_n\}$ and parameter $k \in \mathbb{N}$, let $U = \{u_1, \dots, u_n\}$ be the set of vectors returned by the Capping Algorithm 4.2 with input vector set V and $\Delta = \frac{d}{\epsilon \phi^{\mathsf{A}}(V)}$. Let $S_0 \subseteq [n]$ be an initial set of size $r \geq d$ where $\operatorname{tr}\left(\left(\sum_{i \in S_0} u_i u_i^\top\right)^{-1}\right) \leq \kappa \cdot \phi^{\mathsf{A}}(U)$ for some $\kappa \geq 1$. Let (I, X) be the solution returned by Algorithm 4.9 with vector set U and parameter k. If $k \geq \frac{r}{\epsilon} + \frac{d\left(\log^2 \kappa + \log^2 \frac{1}{\epsilon}\right)}{\epsilon^3}$ and $\epsilon \leq 0.0001$, then*

$$\operatorname{tr}\left(\left(\sum_{i \in I} v_i v_i^\top\right)^{-1}\right) \leq (1 + 6000\epsilon)\phi^{\mathsf{A}}(V)$$

Proof. By Lemma 4.3.1, substituting Δ, we have

$$\begin{aligned} \phi_f^{\mathsf{A}}(U) &\leq \left(1 + \frac{5000d}{k}\right)\left(\phi_f^{\mathsf{A}}(V) + 150\epsilon\phi^{\mathsf{A}}(V)\right) \\ &\leq (1 + 5500\epsilon)\phi^{\mathsf{A}}(V) \end{aligned} \tag{4.29}$$

where the last inequality follows from $\phi^{\mathsf{A}}(V) \geq \phi_f^{\mathsf{A}}(V)$, $k \geq \frac{d}{\epsilon}$. and $\epsilon \leq 0.0001$. Thus, we have

$$\Lambda = \sqrt{\frac{\Delta \phi_f^{\mathsf{A}}(U)}{k}} = \sqrt{\frac{d\phi_f^{\mathsf{A}}(U)}{\epsilon k \phi^{\mathsf{A}}(V)}} \leq \sqrt{\frac{d(1 + 5500\epsilon)}{\epsilon k}} \leq 2\sqrt{\frac{d}{\epsilon k}}$$

Next, Theorem 4.9.1 implies that

$$\operatorname{tr}(X^{-1}) \leq \left(1 - \frac{d + r}{k} - 2\Lambda \log \frac{k \max\{\Lambda\kappa, 1\}}{d}\right)^{-1} \phi^{\mathsf{A}}(U) \tag{4.30}$$

Note that

$$\begin{aligned} 2\Lambda \log \frac{k \max\{\Lambda\kappa, 1\}}{d} &\le 2\Lambda \log \frac{k\kappa}{d} \\ &\le 4\sqrt{\frac{d}{\epsilon k}} \log \frac{k}{d} + 4\sqrt{\frac{d}{\epsilon k}} \log \kappa \end{aligned}$$

Since $\frac{1}{\sqrt{x}} \log x$ is a decreasing function on $x \ge 8$, applying $k \ge \frac{d \log^2 \frac{1}{\epsilon}}{\epsilon^3}$, we have

$$\sqrt{\frac{d}{\epsilon k}} \log \frac{k}{d} \le \frac{\epsilon}{\log \frac{1}{\epsilon}} \left(3 \log \frac{1}{\epsilon} + \log \log \frac{1}{\epsilon} + \log 2 \right) \le 4\epsilon$$

where the last inequality follows from $\epsilon \le 0.0001$. Also, applying $k \ge \frac{d \log^2 \frac{1}{\kappa}}{\epsilon^3}$, $k \ge \frac{d \log^2 \frac{1}{\epsilon}}{\epsilon^3} \ge \frac{d}{\epsilon}$, and $k \ge \frac{r}{\epsilon}$, we have

$$\sqrt{\frac{d}{\epsilon k}} \log \kappa \le \epsilon, \qquad \frac{d}{k} \le \epsilon, \qquad \frac{r}{k} \le \epsilon$$

Hence, (4.30) implies that

$$\operatorname{tr}(X^{-1}) \le (1 - 22\epsilon)^{-1} \phi^{\mathrm{A}}(U) \tag{4.31}$$

Combining (4.31) with Lemma 4.3.1 and (4.29) gives

$$\begin{aligned} \operatorname{tr}\left(\left(\sum_{i \in I} v_i v_i^\top \right)^{-1} \right) &\le \operatorname{tr}(X^{-1}) \le (1 - 22\epsilon)^{-1} (1 + 5500\epsilon) \phi^{\mathrm{A}}(V) \\ &\le (1 + 6000\epsilon) \phi^{\mathrm{A}}(V) \end{aligned}$$

where the last inequality follows from $\epsilon \le 0.0001$. □

We note an efficient combinatorial algorithm of volume sampling [AB13, DW17a] that gives $\frac{n}{k}$-approximation to the A-DESIGN problem of selecting d vectors (note that these randomized algorithms can be derandomized, e.g. by rejection sampling). Alternatively, from our result on approximate local search algorithm for A-DESIGN in Section 4.7, we can also initialize with $c \cdot d$ vectors for an absolute constant c and perform local search algorithm to obtain $1 + 0.0001 + \delta$ approximation in time polynomial in $\frac{1}{\delta}$ for some small δ. Similar to Claim 21, we can relate the optimum of A-DESIGN of size budget $d \leq r \leq k$ and k to be at most factor $\frac{k}{r-d+1}$ apart [AB13, NST19]. Hence, the volume sampling on initial set of size d and local search on initial set of size cd give approximation ratio of n and $\frac{k}{cd-d+1}(1 + 0.0001 + \delta) \leq \frac{k}{d}$, respectively; that is, κ can be set to n or $\frac{k}{d}$ in Theorem 4.9.4 and we adjust r accordingly. Using the local search on initial cd vectors to set the value of κ and r, we prove Theorem 4.1.4.

Proof of Theorem 4.1.4. Suppose $k \geq C \cdot \frac{d}{\epsilon^3} \log^2 \frac{1}{\epsilon}$ for some absolute constant $C > 0$ to be specified later and $\epsilon \leq 0.0001$. By Theorem 4.9.4, it is sufficient to have $k \geq \frac{r}{\epsilon} + \frac{d\left(\log^2 \kappa + \log^2 \frac{1}{\epsilon}\right)}{\epsilon^3}$, where $\kappa = \frac{k}{d}$ and $r = cd$ by initializing the greedy algorithm with an output from an approximate local search algorithm of size cd for an absolute constant c. By checking the derivative of $f(k) := k - \frac{cd}{\epsilon} - \frac{d\left(\log^2 \frac{k}{d} + \log^2 \frac{1}{\epsilon}\right)}{\epsilon^3}$, $f(k)$ is increasing when $2d \log \frac{k}{d} \leq k\epsilon^3$, which is true for a large enough C. Hence, we only need to show $f(k) \geq 0$ for $k = C \cdot \frac{d}{\epsilon^3} \log^2 \frac{1}{\epsilon}$. The condition $f(k) \geq 0$ is equivalent to

$$C \log^2 \frac{1}{\epsilon} \geq \log^2 \frac{C \log^2 \frac{1}{\epsilon}}{\epsilon^3} + \log^2 \frac{1}{\epsilon} + c\epsilon^2 \qquad (4.32)$$

It is clear that $\log^2 \frac{1}{\epsilon} + c\epsilon^2 \leq \frac{C}{2} \log^2 \frac{1}{\epsilon}$ for $C \geq 3 + c$. We also have

$$\begin{aligned}
\log^2 \frac{C \log^2 \frac{1}{\epsilon}}{\epsilon^3} &= \left(\log C + 3 \log \frac{1}{\epsilon} + 2 \log \log \frac{1}{\epsilon} \right)^2 \\
&\leq \left(\log C + 5 \log \frac{1}{\epsilon} \right)^2 \\
&\leq \left(\sqrt{\frac{C}{2}} - 5 + 5 \log \frac{1}{\epsilon} \right)^2 \\
&\leq \left(\sqrt{\frac{C}{2}} \log \frac{1}{\epsilon} \right)^2
\end{aligned}$$

where we use $x \leq e^x$ for $x = \log \frac{1}{\epsilon}$, $\log C \leq \sqrt{C} - 5$ for a sufficiently large C, and $\log \frac{1}{\epsilon} \geq 1$ for the three inequalities above, respectively. Hence, we finished the proof of (4.32). □

CHAPTER 5
MULTI-CRITERIA DIMENSIONALITY REDUCTION WITH APPLICATIONS TO FAIRNESS

5.1 Introduction

Dimensionality reduction is the process of choosing a low-dimensional representation of a large, high-dimensional data set. It is a core primitive for modern machine learning and is being used in image processing, biomedical research, time series analysis, etc. Dimensionality reduction can be used during the preprocessing of the data to reduce the computational burden as well as at the final stages of data analysis to facilitate data summarization and data visualization [RSA99, IP91]. Among the most ubiquitous and effective of dimensionality reduction techniques in practice are Principal Component Analysis (PCA) [Pea01, Jol86, Hot33], multidimensional scaling [Kru64], Isomap [TDSL00], locally linear embedding [RS00], and t-SNE [MH08].

One of the major obstacles to dimensionality reduction tasks in practice is complex high-dimensional data structures that lie on multiple different low-dimensional subspaces. For example, Maaten and Hinton [MH08] address this issue for low-dimensional visualization of images of objects from diverse classes seen from various viewpoints, or Samadi et al. [Sam+18] study PCA on human data when different groups in the data (e.g., high-educated vs low-educated or men vs women) have an inherently different structure. Although these two contexts might seem unrelated, our work presents a general framework that addresses both issues. In both setting, a single criteria for the dimensionality reduction might not be sufficient to capture different structures in the data. This motivates our study of multi-criteria dimensionality reduction.

As an illustration, consider applying PCA on a high dimensional data to do a visualization analysis in low dimensions. Standard PCA aims to minimize the single criteria of average reconstruction error over the whole data. But the reconstruction error on different parts of data can be widely different. In particular, [Sam+18] show that on real world data sets, PCA has more reconstruction error on images of women vs images of men. A similar phenomenon is also noticed on other data sets when groups are formed based on education. Unbalanced average reconstruction error or equivalently unbalanced variance could have implications of representational harms [Cra17] in early stages of data analysis.

Multi-criteria dimensionality reduction. Multi-criteria dimensionality reduction could be used as an umbrella term with specifications changing based on the applications and the metrics that the machine learning researcher has in mind. Aiming for an output with a balanced error over different subgroups seems to be a natural choice as reflected by minimizing the maximum of average reconstruction errors studied by [Sam+18] and maximizing geometric mean of the variances of the groups, which is the well-studied Nash social welfare (NSW) objective [KN79, NJ50]. Motivated by these settings, the more general question that we would like to study is as following.

Question 1. *How might one redefine dimensionality reduction to produce projections which optimize different groups' representation in a balanced way?*

For simplicity of explanation, we first describe our framework for PCA, but the approach is general and applies to a much wider class of dimensionality reduction techniques. Consider the data points as rows of an $m \times n$ matrix A. For PCA, the objective is to find an $n \times d$ projection matrix P that maximizes the Frobenius norm, $\|AP\|_F^2$ (this is equivalent to minimizing the reconstruction error). Suppose that the rows of A belong to different *groups*, based on demographics or some other semantically meaningful clustering. The definition of these groups need not be a partition; each group could be defined as a different weighting of the data set (rather than a subset, which is a 0/1 weighting). Multi-criteria dimensionality reduction can then be viewed as simultaneously considering objectives on the different weightings of A, i.e., A_i. One way to balance multiple objectives is to find a projection P that maximizes the minimum objective value over each of the groups (weightings):

$$\max_{P:P^TP=I_d} \min_{1\le i\le k} \|A_iP\|_F^2 = \langle A_i^TA_i, PP^T\rangle. \qquad \text{(FAIR-PCA)}$$

(We note that our FAIR-PCA is different from one in [Sam+18], but equivalent by additive and multiplicative scalings.) More generally, let $\mathcal{P}_d$ denote the set of all $n \times d$ projection matrices P, i.e., matrices with d orthonormal columns. For each group A_i, we associate a function $f_i : \mathcal{P}_d \to \mathbb{R}$ that denotes the group's objective value for a particular projection. For any $g : \mathbb{R}^k \to \mathbb{R}$, we define the (f, g)-multi-criteria dimensionality reduction problem as finding a d-dimensional projection P which optimizes

$$\max_{P\in\mathcal{P}_d} g(f_1(P), f_2(P), \ldots, f_k(P)). \qquad \text{(MULTI-CRITERIA-DIMENSION-REDUCTION)}$$

In the above example of max-min Fair-PCA, g is simply the $\min$ function and $f_i(P) = \|A_iP\|^2$ is the total squared norm of the projection of vectors in A_i. Other examples include: defining each f_i as the average squared norm of the projections rather than the total, or the marginal variance — the difference in total squared norm when using P rather than the best possible projection for that group. One could also choose the product function $g(y_1, \ldots, y_k) = \prod_i y_i$ for the accumulating function g. This is also a natural choice, famously introduced in Nash's solution to the bargaining problem [NJ50, KN79]. This framework can also describe the pth power mean of the projections, e.g. $f_i(P) = \|A_iP\|^2$ and $g(y_1, \ldots, y_k) = \left(\sum_{i\in[k]} y_i^{p/2}\right)^{1/p}$.

The appropriate weighting of k objectives often depends on the context and application. The central motivating questions of this paper are the following:

◇ *What is the complexity of* FAIR-PCA *?*

◇ *More generally, what is the complexity of* MULTI-CRITERIA-DIMENSION-REDUCTION *?*

Framed another way, we ask whether these multi-criteria optimization problems force us to incur substantial computational cost compared to optimizing g over A alone. Samadi et al. [Sam+18] introduced the problem of Fair-PCA and showed how to use the natural semi-definite relaxation to find a rank-($d + k - 1$) approximation whose cost is at most that of the optimal rank-d approximation. For $k = 2$ groups, this is an increase of 1 in the dimension (as opposed to the naïve bound of $2d$, by taking the span of the optimal d-dimensional subspaces for the two groups). The computational complexity of finding the exact optimal solution to Fair-PCA was left as an open question.

5.1.1 Results and Techniques

Let us first focus on Fair-PCA for ease of exposition. The problem can be reformulated as the following mathematical program where we denote PP^T by X. A natural approach to solving this problem is to consider the SDP relaxation obtained by relaxing the rank constraint to a bound on the trace.

Exact Fair-PCA	**SDP Relaxation of Fair-PCA**
$\max \ z$	$\max \ z$
$\langle A_i^T A_i, X \rangle \geq z \quad i \in \{1, \ldots, k\}$	$\langle A_i^T A_i, X \rangle \geq z \quad i \in \{1, \ldots, k\}$
$\text{rank}(X) \leq d$	$\mathrm{tr}(X) \leq d$
$0 \preceq X \preceq I$	$0 \preceq X \preceq I$

Our first main result is that the SDP relaxation is exact when there are *two* groups. Thus finding an extreme point of this SDP gives an exact algorithm for Fair-PCA for two groups. Previously, only approximation algorithms were known for this problem. This result also resolves the open problem posed by Samadi et al. [Sam+18].

Theorem 5.1.1. *Any optimal extreme point solution to the SDP relaxation for* FAIR-PCA *with two groups has rank at most d. Therefore, 2-group* FAIR-PCA *can be solved in polynomial time.*

Given m datapoints partitioned into $k \leq n$ groups in n dimensions, the algorithm runs in $O(nm + n^{6.5})$ time. $O(mnk)$ is from computing $A_i^T A_i$ and $O(n^{6.5})$ is from solving an SDP over $n \times n$ PSD matrices [BTN01]. Our results also hold for the MULTI-CRITERIA-DIMENSION-REDUCTION when g is monotone nondecreasing in any one coordinate and concave, and each f_i is an affine function of PP^T (and thus a special case of a quadratic function in P).

Theorem 5.1.2. *There is a polynomial time algorithm for 2-group* MULTI-CRITERIA-DIMENSION-REDUCTION *problem when g is concave and monotone nondecreasing for at least one of its two arguments, and each f_i is linear in PP^T, i.e., $f_i(P) = \langle B_i, PP^T \rangle$ for some matrix $B_i(A)$.*

As indicated in the theorem, the core idea is that extreme-point solutions of the SDP in fact have rank d, not just trace equal to d.

For $k > 2$, the SDP need not recover a rank d solution. In fact, the SDP may be inexact even for $k = 3$ (see Section 5.7). Nonetheless, we show that we can bound the rank of a solution to the SDP and obtain the following result. We state it for FAIR-PCA, though the same bound holds for MULTI-CRITERIA-DIMENSION-REDUCTION under the same assumptions as in Theorem 5.1.1. Note that this result generalizes Theorem 5.1.1.

Theorem 5.1.3. *For any concave g that is monotone nondecreasing in at least one of its arguments, there exists a polynomial time algorithm for* FAIR-PCA *with k groups that returns a $d + \lfloor\sqrt{2k + \frac{1}{4}} - \frac{3}{2}\rfloor$-dimensional embedding whose objective value is at least that of the optimal d-dimensional embedding. If g is only concave, then the solution lies in at most $d + 1$ dimensions.*

This strictly improves and generalizes the bound of $d+k-1$ for FAIR-PCA problem. Moreover, if the dimensionality of the solution is a hard constraint, instead of tolerating $s = O(\sqrt{k})$ extra dimension in the solution, one may solve FAIR-PCA for target dimension $d - s$ to guarantee a solution of rank at most d. Thus, we obtain an approximation algorithm for FAIR-PCA of factor $1-\frac{O(\sqrt{k})}{d}$.

Theorem 5.1.4. *Let $A_1,\ldots,A_k$ be data sets of k groups and suppose $s := \lfloor\sqrt{2k+\frac{1}{4}}-\frac{3}{2}\rfloor < d$. Then there exists a polynomial-time approximation algorithm of factor $1-\frac{s}{d} = 1-\frac{O(\sqrt{k})}{d}$ to* FAIR-PCA *problem.*

That is, the algorithm returns a project $P \in \mathcal{P}_d$ of *exact* rank d with objective at least $1-\frac{s}{d}$ of the optimal objective. More details on the approximation result are in Section 5.3. The runtime of Theorems 5.1.2 and 5.1.3 depends on access to first order oracle to g and standard application of the ellipsoid algorithm would take $\tilde{O}(n^2)$ oracle calls.

We now focus our attention to the marginal loss function. This measures the maximum over the groups of the difference between the variance of a common solution for the k groups and an optimal solution for an individual group (“the marginal cost of sharing a common subspace"). For this problem, the above scaling method could substantially harm the objective value, since the target function is nonlinear. MULTI-CRITERIA-DIMENSION-REDUCTION captures the marginal loss functions by setting the utility $f_i(P) = \|A_iP\|_F^2 - \max_{Q\in\mathcal{P}_d}\|A_iQ\|_F^2$ for each group i and $g(f_1,f_2,\ldots,f_k) := \min\{f_1,f_2,\ldots,f_k\}$, giving an optimization problem

$$\min_{P\in\mathcal{P}_d}\max_{i\in[k]}\left(\max_{Q\in\mathcal{P}_d}\|A_iQ\|_F^2 - \|A_iP\|_F^2\right) \tag{5.1}$$

and the marginal loss objective is indeed the objective of the problem.

In Section 5.4, we develop a general rounding framework for SDPs with eigenvalue upper bounds and k other linear constraints. This algorithm gives a solution of desired rank that violates each constraint by a bounded amount. The precise statement is Theorem 5.1.8. It implies that for

FAIR-PCA with marginal loss as the objective the additive error is

$$\Delta(\mathcal{A}) := \max_{S\subseteq[m]} \sum_{i=1}^{\lfloor\sqrt{2|S|}+1\rfloor} \sigma_i(A_S)$$

where $A_S = \frac{1}{|S|}\sum_{i\in S} A_i$.

It is natural to ask whether FAIR-PCA is NP-hard to solve exactly. The following result implies that it is, even for target dimension $d = 1$.

Theorem 5.1.5. *The max-min* FAIR-PCA *problem for target dimension $d = 1$ is NP-hard when the number of groups k is part of the input.*

This raises the question of the complexity for constant $k \geq 3$ groups. For k groups, we would have k constraints, one for each group, plus the eigenvalue constraint and the trace constraint; now the tractability of the problem is far from clear. In fact, as we show in Section 5.7, the SDP has an integrality gap even for $k = 3, d = 1$. We therefore consider an approach beyond SDPs, to one that involves solving non-convex problems. Thanks to the powerful algorithmic theory of quadratic maps, developed by Grigoriev and Pasechnik [GP05], it is polynomial-time solvable to check feasibility of a set of quadratic constraints for any fixed k. As we discuss next, their algorithm can check for zeros of a function of a set of k quadratic functions, and can be used to optimize the function. Using this result, we show that for $d = k = O(1)$, there is a polynomial-time algorithm for rather general functions g of the values of individual groups.

Theorem 5.1.6. *Let the fairness objective be $g : \mathbb{R}^k \to \mathbb{R}$ where g is a degree ℓ polynomial in some computable subring of $\mathbb{R}^k$ and each f_i is quadratic for $1 \leq i \leq k$. Then there is an algorithm to solve the fair dimensionality reduction problem in time $(\ell dn)^{O(k+d^2)}$.*

By choosing g to be the product polynomial over the usual $(\times, +)$ ring or the $\min$ function which is degree k in the $(\min, +)$ ring, this applies to the variants of FAIR-PCA discussed above and various other problems.

SDP extreme points. For $k = 2$, the underlying structural property we show is that extreme point solutions of the SDP have rank exactly d. First, for $k = d = 1$, this is the largest eigenvalue problem, since the maximum obtained by a matrix of trace equal to 1 can also be obtained by one of the extreme points in the convex decomposition of this matrix. This extends to trace equal to any d, i.e., the optimal solution must be given by the top k eigenvectors of $A^T A$. Second, without the eigenvalue bound, for any SDP with k constraints, there is an upper bound on the rank of any extreme point, of $O(\sqrt{k})$, a seminal result of Pataki [Pat98] (see also Barvinok [Bar95]). However, we cannot apply this directly as we have the eigenvalue upper bound constraint. The complication here is that we have to take into account the constraint $X \preceq I$ without increasing the rank.

Theorem 5.1.7. *Let C and $A_1, \ldots, A_m$ be $n \times n$ real matrices, $d \leq n$, and $b_1, \ldots b_m \in \mathbb{R}$. Suppose the semi-definite program $\mathbb{SDP}(\mathbb{I})$:*

$$\min\langle C, X\rangle \textit{ subject to} \tag{5.2}$$

$$\langle A_i, X\rangle \quad \lhd_i \quad b_i \;\; \forall\, 1 \leq i \leq m \tag{5.3}$$

$$\mathrm{tr}(X) \quad \leq \quad d \tag{5.4}$$

$$0 \preceq X \quad \preceq \quad I_n \tag{5.5}$$

where $\lhd_i \in \{\leq, \geq, =\}$, has a nonempty feasible set. Then, all extreme optimal solutions X^* to $\mathbb{SDP}(\mathbb{I})$ have rank at most $r^* := d + \lfloor\sqrt{2m + \frac{9}{4}} - \frac{3}{2}\rfloor$. Moreover, given a feasible optimal solution, an extreme optimal solution can be found in polynomial time.

To prove the theorem, we extend Pataki [Pat98]'s characterization of rank of SDP extreme points with minimal loss in the rank. We show that the constraints $0 \preceq X \preceq I$ can be interpreted as a generalization of restricting variables to lie between 0 and 1 in the case of linear programming relaxations. From a technical perspective, our results give new insights into structural properties of extreme points of semi-definite programs and more general convex programs. Since the result of

[Pat98] has been studied from perspective of fast algorithms [BVB16, BM03, BM05] and applied in community detection and phase synchronization Bandeira, Boumal, and Voroninski [BBV16], we expect our extension of the result to have further applications in many of these areas.

SDP iterative rounding. Using Theorem 5.1.7, we extend the iterative rounding framework for linear programs (see [LRS11] and references therein) to semi-definite programs, where the $0, 1$ constraints are generalized to eigenvalue bounds. The algorithm has a remarkably similar flavor. In each iteration, we fix the subspaces spanned by eigenvectors with 0 and 1 eigenvalues, and argue that one of the constraints can be dropped while bounding the total violation in the constraint over the course of the algorithm. While this applies directly to the FAIR-PCA problem, in fact is a general statement for SDPs, which we give below.

Let $\mathcal{A} = \{A_1, \dots, A_m\}$ be a collection of $n \times n$ matrices. For any set $S \subseteq \{1, \dots, m\}$, let $\sigma_i(S)$ the i^{th} largest singular of the average of matrices $\frac{1}{|S|} \sum_{i \in S} A_i$. We let

$$\Delta(\mathcal{A}) := \max_{S \subseteq [m]} \sum_{i=1}^{\lfloor \sqrt{2|S|+1} \rfloor} \sigma_i(S).$$

Theorem 5.1.8. *Let C be a $n \times n$ matrix and $\mathcal{A} = \{A_1, \dots, A_m\}$ be a collection of $n \times n$ real matrices, $d \leq n$, and $b_1, \dots b_m \in \mathbb{R}$. Suppose the semi-definite program $\mathbb{SDP}$:*

$$\begin{aligned} &\min \langle C, X \rangle \textit{ subject to} \\ &\qquad \langle A_i, X \rangle \geq b_i \;\; \forall\, 1 \leq i \leq m \\ &\qquad \operatorname{tr}(X) \leq d \\ &\qquad 0 \preceq X \preceq I_n \end{aligned}$$

has a nonempty feasible set and let X^* denote an optimal solution. The Algorithm ITERATIVE-

SDP *(see Figure 5.1) returns a matrix* $\tilde{X}$ *such that*

1. *rank of* $\tilde{X}$ *is at most* d*,*
2. $\langle C, \tilde{X}\rangle \leq \langle C, X^*\rangle$*, and*
3. $\langle A_i, \tilde{X}\rangle \geq b_i - \Delta(\mathcal{A})$ *for each* $1 \leq i \leq m$*.*

The time complexity of Theorems 5.1.7 and 5.1.8 is analyzed in Sections 5.2 and 5.4. Both algorithms introduce the rounding procedures that do not contribute significant computational cost; rather, solving the SDPis the bottleneck for running time both in theory and practice.

5.1.2 Related Work

As mentioned earlier, Pataki [Pat98] (see also Barvinok [Bar95]) showed low rank solutions to semi-definite programs with small number of affine constraints can be obtained efficiently. Restricting a feasible region of certain SDPs relaxations with low-rank constraints has been shown to avoid spurious local optima [BBV16] and reduce the runtime due to known heuristics and analysis [BM03, BM05, BVB16]. We also remark that methods based on Johnson-Lindenstrauss lemma can also be applied to obtain bi-criteria results for FAIR-PCA problem. For example, So, Ye, and Zhang [SYZ08] give algorithms that give low rank solutions for SDPs with affine constraints without the upper bound on eigenvalues. Here we have focused on single criteria setting, with violation either in the number of dimensions or the objective but not both. We also remark that extreme point solutions to linear programming have played an important role in design of approximation algorithms [LRS11] and our result add to the comparatively small, but growing, number of applications for utilizing extreme points of semi-definite programs.

A closely related area, especially to MULTI-CRITERIA-DIMENSION-REDUCTION problem, is multi-objective optimization which has a vast literature. We refer the reader to Deb [Deb14] and references therein. We also remark that properties of extreme point solutions of linear programs [RG96, GRSZ14] have also been utilized to obtain approximation algorithms to multi-objective problems. For semi-definite programming based methods, the closest works are on simultaneous max-cut [BKS15, Bha+18] that utilize sum of squares hierarchy to obtain improved approximation algorithms.

The applications of multi-criteria dimensionality reduction in fairness are closely related to studies on representational bias in machine learning [Cra17, Nob18, Bol+16] and fair resource allocation in game theory [WVZX10, FB04]. There have been various mathematical formulations studied for representational bias in ML [CKLV17, Cel+18, Sam+18, KAM19, KSAM19] among which our model covers unbalanced reconstruction error in PCA suggested by Samadi et al. [Sam+18]. From the game theory literature, our model covers Nash social welfare objective [KN79, NJ50] and others [KS+75, Kal77].

5.2 Low-Rank Solutions of MULTI-CRITERIA-DIMENSION-REDUCTION

In this section, we show that all extreme solutions of SDP relaxation of MULTI-CRITERIA-DIMENSION-REDUCTION have low rank, proving Theorem 5.1.1-5.1.3. Before we state the results, we make following assumptions. In this section, we let $g : \mathbb{R}^k \to \mathbb{R}$ be a concave function which is monotonic in at least one coordinate, and mildly assume that g can be accessed with a polynomial-time subgradient oracle and is polynomially bounded by its input. We are explicitly given functions $f_1, f_2, \ldots, f_k$ which are affine in PP^T, i.e. we are given real $n \times n$ matrices $B_1, \ldots, B_k$ and constants $\alpha_1, \alpha_2, \ldots, \alpha_k \in \mathbb{R}$ and $f_i(P) = \langle B_i, PP^T \rangle + \alpha_i$.

We assume g to be G-Lipschitz. For functions $f_1, \ldots, f_k, g$ that are $L_1, \ldots, L_k, G$-Lipschitz, we define an ϵ-optimal solution to (f, g)-MULTI-CRITERIA-DIMENSION-REDUCTION problem as a projection matrix $X \in \mathbb{R}^{n\times n}, 0 \preceq X \preceq I_n$ of rank d whose objective value is at most $G\epsilon \left(\sum_{i=1}^k L_i^2\right)^{1/2}$ from the optimum. In the context where an optimization problem has affine constraints $F_i(X) \leq b_i$ where F_i is L_i Lipschitz, we also define ϵ-solution as a projection matrix $X \in \mathbb{R}^{n\times n}, 0 \preceq X \preceq I_n$ of rank d that violates ith affine constraints by at most ϵL_i. Note that the feasible region of the problem is implicitly bounded by the constraint $X \preceq I_n$.

In this section, the algorithm may involve solving an optimization under a matrix linear inequality, which may not give an answer representable in finite bits of computation. However, we give algorithms that return an ϵ-close solution whose running time depends polynomially on $\log \frac{1}{\epsilon}$ for any $\epsilon > 0$. This is standard for computational tractability in convex optimization (see, for example, in [BTN01]). Therefore, for ease of exposition, we omit the computational error dependent on this ϵ to obtain an ϵ-feasible and ϵ-optimal solution, and define polynomial running time as polynomial in n, k and $\log \frac{1}{\epsilon}$.

We first prove Theorem 5.1.7 below. To prove Theorem 5.1.1-5.1.3, we first show that extreme point solutions in semi-definite cone under affine constraints and $X \preceq I$ have low rank. The statement builds on a result of [Pat98]. We then apply our result to MULTI-CRITERIA-DIMENSION-REDUCTION problem, which contains the FAIR-PCA problem. Finally, we show that existence of low-rank solution leads to an approximation algorithm to FAIR-PCA problem.

Proof of Theorem 5.1.7: Let X^* be an extreme point optimal solution to $\mathbb{SDP}(\mathbb{I})$. Suppose rank of X^*, say r, is more than r^*. Then we show a contradiction to the fact that X^* is extreme. Let $0 \leq l \leq r$ of the eigenvalues of X^* be equal to one. If $l \geq d$, then we have $l = r = d$ since $\operatorname{tr}(X) \leq d$ and we are done. Thus we assume that $l \leq d - 1$. In that case, there exist matrices $Q_1 \in \mathbb{R}^{n \times r-l}$, $Q_2 \in \mathbb{R}^{n \times l}$ and a symmetric matrix $\Lambda \in \mathbb{R}^{(r-l)\times(r-l)}$ such that

$$X^* = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} \Lambda & 0 \\ 0 & I_l \end{pmatrix} \begin{pmatrix} Q_1 & Q_2 \end{pmatrix}^\top = Q_1 \Lambda Q_1^\top + Q_2 Q_2^T$$

where $0 \prec \Lambda \prec I_{r-l}$, $Q_1^T Q_1 = I_{r-l}$, $Q_2^T Q_2 = I_l$, and that the columns of Q_1 and Q_2 are orthogonal, i.e. $Q = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix}$ has orthonormal columns. Now, we have

$$\langle A_i, X^* \rangle = \langle A_i, Q_1 \Lambda Q_1^\top + Q_2 Q_2^\top \rangle = \langle Q_1^\top A_i Q_1, \Lambda \rangle + \langle A_i, Q_2 Q_2^\top \rangle$$

and $\operatorname{tr}(X^*) = \langle Q_1^\top Q_1, \Lambda \rangle + \operatorname{tr}(Q_2 Q_2^\top)$ so that $\langle A_i, X^* \rangle$ and $\operatorname{tr}(X^*)$ are linear in Λ.

Observe the set of $s \times s$ symmetric matrices forms a vector space of dimension $\frac{s(s+1)}{2}$ with the above inner product where we consider the matrices as long vectors. If $m+1 < \frac{(r-l)(r-l+1)}{2}$ then there exists a $(r-l) \times (r-l)$-symmetric matrix $\Delta \neq 0$ such that $\langle Q_1^\top A_i Q_1, \Delta \rangle = 0$ for each $1 \leq i \leq m$ and $\langle Q_1^\top Q_1, \Delta \rangle = 0$.

But then we claim that $Q_1(\Lambda \pm \delta\Delta)Q_1^\top + Q_2 Q_2^T$ is feasible for small $\delta > 0$, which implies a contradiction to X^* being extreme. Indeed, it satisfies all the linear constraints by construction of Δ. Thus it remains to check the eigenvalues of the newly constructed matrix. Observe that

$$Q_1(\Lambda \pm \delta\Delta)Q_1^\top + Q_2 Q_2^T = Q \begin{pmatrix} \Lambda \pm \delta\Delta & 0 \\ 0 & I_l \end{pmatrix} Q^\top$$

with orthonormal Q. Thus it is enough to consider the eigenvalues of $\begin{pmatrix} \Lambda \pm \delta\Delta & 0 \\ 0 & I_l \end{pmatrix}$.

Observe that eigenvalues of the above matrix are exactly l ones and eigenvalues of $\Lambda \pm \delta\Delta$. Since eigenvalues of Λ are bounded away from 0 and 1, one can find small δ such that the eigenvalue of $\Lambda \pm \delta\Delta$ are bounded away from 0 and 1 as well, so we are done. Therefore, we must have $m+1 \geq \frac{(r-l)(r-l+1)}{2}$ which implies $r - l \leq -\frac{1}{2} + \sqrt{2m + \frac{9}{4}}$. By $l \leq d-1$, we have $r \leq r^*$.

For the algorithmic version, given feasible $\bar{X}$, we iteratively reduce $r-l$ by at least one until $m+1 \geq \frac{(r-l)(r-l+1)}{2}$. While $m+1 < \frac{(r-l)(r-l+1)}{2}$, we obtain Δ by using Gaussian elimination. Now we want to find the correct value of $\pm\delta$ so that $\Lambda' = \Lambda \pm \delta\Delta$ takes one of the eigenvalues to zero or one. First, determine the sign of $\langle C, \Delta \rangle$ to find the correct sign to move Λ that keeps the objective non-increasing, say it is in the positive direction. Since the set of feasible X is convex and bounded, the ray $f(t) = Q_1(\Lambda + t\Delta)Q_1^\top + Q_2 Q_2^\top, t \geq 0$ intersects the boundary of feasible region at a unique $t' > 0$. Perform binary search for the correct value of t' and set $\delta = t'$ up to the desired accuracy. Since $\langle Q_1^\top A_i Q_1, \Delta \rangle = 0$ for each $1 \leq i \leq m$ and $\langle Q_1^\top Q_1, \Delta \rangle = 0$, the additional tight constraint from moving $\Lambda' \leftarrow \Lambda + \delta\Delta$ to the boundary of feasible region must be an eigenvalue constraint $0 \preceq X \preceq I_n$, i.e., at least one additional eigenvalue is now at 0 or 1, as desired. We apply eigenvalue decomposition to Λ' and update Q_1 accordingly, and repeat.

The algorithm involves at most n rounds of reducing $r-l$, each of which involves Gaussian elimination and several iterations (from binary search) of $0 \preceq X \preceq I_n$ which can be done by eigenvalue value decomposition. Gaussian elimination and eigenvalue decomposition can be done in $O(n^3)$ time, and therefore the total runtime of SDP rounding is $\tilde{O}(n^4)$ which is polynomial. □

In practice, one may initially reduce the rank of given feasible $\bar{X}$ using an LP rounding (in $O(n^{3.5})$ time) introduced in [Sam+18] so that the number of rounds of reducing $r-l$ is further bounded by $k-1$. The runtime complexity is then $O(n^{3.5})+\tilde{O}(kn^3)$.

The next corollary is obtained from the bound $r-l \leq -\frac{1}{2}+\sqrt{2m+\frac{9}{4}}$ in the proof of Theorem 5.1.7.

Corollary 5.2.1. *The number of fractional eigenvalues in any extreme point solution X to $\mathbb{SDP}(\mathbb{I})$ is bounded by $\sqrt{2m+\frac{9}{4}}-\frac{1}{2} \leq \lfloor\sqrt{2m}+1\rfloor$.*

We are now ready to state the main result of this section that we can find a low-rank solution for MULTI-CRITERIA-DIMENSION-REDUCTION . Recall that $\mathcal{P}_d$ is the set of all $n \times d$ projection matrices P, i.e., matrices with d orthonormal columns and the (f,g)-MULTI-CRITERIA-DIMENSION-REDUCTION problem is to solve

$$\max_{P \in \mathcal{P}_d} g(f_1(P), f_2(P), \ldots, f_k(P)) \tag{5.6}$$

Theorem 5.2.2. *There exists a polynomial-time algorithm to solve (f,g)-*MULTI-CRITERIA-DIMENSION-REDUCTION *that returns a solution $\hat{X}$ of rank at most $r^* := d + \lfloor\sqrt{2k+\frac{1}{4}}-\frac{3}{2}\rfloor$ whose objective value is at least that of the optimal d-dimensional embedding.*

If the assumption that g is monotonic in at least one coordinate is dropped, Theorem 5.2.2 will hold with r^* by indexing constraints (5.11) in $\mathbb{SDP}(\mathbb{II})$ for all groups instead of $k-1$ groups.

Proof of Theorem 5.2.2: First, we write a relaxation of (5.6):

$$\max_{X\in\mathbb{R}^{n\times n}} g(\langle B_1, X\rangle + \alpha_1, \ldots, \langle B_k, X\rangle + \alpha_k) \text{ subject to} \tag{5.7}$$

$$\mathrm{tr}(X) \leq d \tag{5.8}$$

$$0 \preceq X \preceq I_n \tag{5.9}$$

Since $g(x)$ is concave in $x \in \mathbb{R}^k$ and $\langle B_i, X\rangle + \alpha_i$ is affine in $X \in \mathbb{R}^{n\times n}$, we have that g as a function of X is also concave in X. By assumptions on g, and the fact that the feasible set is convex and bounded, we can solve the convex program in polynomial time, e.g. by ellipsoid method, to obtain a (possibly high-rank) optimal solution $\bar{X} \in \mathbb{R}^{n\times n}$. (In the case that f_i is linear, the relaxation is also an SDP and may be solved faster in theory and practice). By assumptions on g, without loss of generality, we let g be nondecreasing in the first coordinate. To reduce the rank of $\bar{X}$, we consider an $\mathbb{SDP}(\mathbb{II})$:

$$\max_{X\in\mathbb{R}^{n\times n}} \quad \langle B_1, X\rangle \text{ subject to} \tag{5.10}$$

$$\langle B_i, X\rangle = \left\langle B_i, \bar{X}\right\rangle \qquad \forall\, 2 \leq i \leq k \tag{5.11}$$

$$\mathrm{tr}(X) \leq d \tag{5.12}$$

$$0 \preceq X \preceq I_n \tag{5.13}$$

$\mathbb{SDP}(\mathbb{II})$ has a feasible solution $\bar{X}$ of objective $\langle B_1, X\rangle$ and note that there are $k-1$ constraints in (5.11). Hence, we can apply the algorithm in Theorem 5.1.7 with $m = k-1$ to find an extreme solution X^* of $\mathbb{SDP}(\mathbb{II})$ of rank at most r^*. Since g is nondecreasing in $\langle B_1, X\rangle$, optimal solutions to $\mathbb{SDP}(\mathbb{II})$ gives objective value g at least the optimum of the relaxation and hence at least the optimum of the original MULTI-CRITERIA-DIMENSION-REDUCTION problem. □

Another way to state Theorem 5.2.2 is that the number of groups must reach $\frac{(s+1)(s+2)}{2}$ before additional s dimensions in the solution matrix P is required to achieve the optimal objective value. For $k=2$, no additional dimension in the solution is necessary to attain the optimum. We state this fact as follows. In particular, it applies to FAIR-PCA with two groups, proving Theorem 5.1.1.

Corollary 5.2.3. *The* (f,g)-MULTI-CRITERIA-DIMENSION-REDUCTION *problem on two groups can be solved in polynomial time.*

5.3 Approximation Algorithm for FAIR-PCA

Recall that we require $s := \lfloor \sqrt{2k+\frac{1}{4}} - \frac{3}{2} \rfloor$ additional dimensions for the projection to achieve the optimal objective. One way to ensure that the algorithm outputs d-dimensional projection is to solve the problem in lower target dimension $d-s$, then apply the rounding described in Section 5.2. The relationship of objectives between problems with target dimension $d-s$ and d is at most $\frac{d-s}{d}$ factor apart for FAIR-PCA problem because the objective scales linearly with P, giving an approximation guarantee of $1-\frac{s}{d}$. Recall that given $A_1,\ldots,A_k$, FAIR-PCA problem is to solve

$$\max_{P:P^TP=I_d} \min_{1\le i\le k} \|A_iP\|_F^2 = \langle A_i^TA_i, PP^T\rangle$$

We state the approximation guarantee and the algorithm formally as follows.

Corollary 5.3.1. *Let* $A_1,\ldots,A_k$ *be data sets of* k *groups and suppose* $s := \lfloor \sqrt{2k+\frac{1}{4}} - \frac{3}{2} \rfloor < d$. *Then there exists a polynomial-time approximation algorithm of factor* $1-\frac{s}{d} = 1-\frac{O(\sqrt{k})}{d}$ *to* FAIR-PCA *problem.*

Proof. We find an extreme solution X^* of the FAIR-PCA problem of finding a projection from n to $d-s$ target dimensions. By Theorem 5.2.2, the rank of X^* is at most d.

Denote OPT_d, X_d^* the optimal value and an optimal solution to FAIR-PCA with target dimension d. Note that $\frac{d-s}{d}X_d^*$ is a feasible solution to FAIR-PCA relaxation on target dimension $d-s$

which is at least $\frac{d-s}{d}\mathrm{OPT}_d$ because the objective scales linearly with X. Therefore, the optimal FAIR-PCA relaxation of target dimension $d-s$ attains optimum at least $\frac{d-s}{d}\mathrm{OPT}_d$, giving $(1-\frac{s}{d})$-approximation ratio. □

5.4 Iterative Rounding Framework with Applications to FAIR-PCA

In this section, we first prove Theorem 5.1.8.

We give an iterative rounding algorithm. The algorithm maintains three subspaces that are mutually orthogonal. Let F_0, F_1, F denote matrices whose columns form an orthonormal basis of these subspaces. We will also abuse notation and denote these matrices by sets of vectors in their columns. We let the rank of F_0, F_1 and F be r_0, r_1 and r, respectively. We will ensure that $r_0 + r_1 + r = n$, i.e., vectors in F_0, F_1 and F span $\mathbb{R}^n$.

We initialize $F_0 = F_1 = \emptyset$ and $F = I_n$. Over iterations, we increase the subspaces spanned by columns of F_0 and F_1 and decrease F while maintaining pairwise orthogonality. The vectors in columns of F_1 will be eigenvectors of our final solution with eigenvalue 1. In each iteration, we project the constraint matrices A_i orthogonal to F_1 and F_0. We will then formulate a residual SDP using columns of F as a basis and thus the new constructed matrices will have size $r \times r$. To readers familiar with the iterative rounding framework in linear programming, this generalizes the method of fixing certain variables to 0 or 1 and then formulating the residual problem. We also maintain a subset of constraints indexed by S where S is initialized to $\{1, \ldots, m\}$.

The algorithm is specified in Figure 5.1. In each iteration, we formulate the following $\mathbb{SDP}(r)$ with variables $X(r)$ which will be a $r \times r$ symmetric matrix. Recall r is the number of columns in

F.

$$\begin{aligned} \max\ & \langle F^T C F, X(r)\rangle \\ \langle F^T A_i F, X(r)\rangle & \geq b_i - F_1^T A_i F_1 \quad i \in S \\ \mathrm{tr}(X) & \leq d - \text{rank}(F_1) \\ 0 \preceq\ & X(r) \preceq I_r \end{aligned}$$

1. Initialize F_0, F_1 to be empty matrices and $F = I_n$, $S \leftarrow \{1, \ldots, m\}$.
2. If the $\mathbb{SDP}$ is infeasible, declare infeasibility. Else,
3. While F is not the empty matrix.
   - (a) Solve $\mathbb{SDP}(r)$ to obtain extreme point $X^*(r) = \sum_{j=1}^{r} \lambda_j v_j v_j^T$ where λ_j are the eigenvalues and $v_j \in \mathbb{R}^r$ are the corresponding eigenvectors.
   - (b) For any eigenvector v of $X^*(r)$ with eigenvalue 0, let $F_0 \leftarrow F_0 \cup \{Fv\}$.
   - (c) For any eigenvector v of $X^*(r)$ with eigenvalue 1, let $F_1 \leftarrow F_1 \cup \{Fv\}$.
   - (d) Let $X_f = \sum_{j:0<\lambda_j<1} \lambda_j v_j v_j^T$. If there exists a constraint $i \in S$ such that $\langle F^T A_i F, X_f \rangle < \Delta(\mathcal{A})$, then $S \leftarrow S \setminus \{i\}$.
   - (e) For every eigenvector v of $X^*(r)$ with eigenvalue not equal to 0 or 1, consider the vectors Fv and form a matrix with these columns and use it as the new F.
4. Return $\tilde{X} = F_1 F_1^T$.

Figure 5.1: Iterative Rounding Algorithm ITERATIVE-SDP.

It is easy to see that the semi-definite program remains feasible over all iterations if $\mathbb{SDP}$ is declared feasible in the first iteration. Indeed the solution X_f defined at the end of any iteration is a feasible solution to the next iteration. We also need the following standard claim.

Claim 22. *Let Y be a positive semi-definite matrix such that $Y \preceq I$ with $\mathrm{tr}(Y) \leq l$. Let B be real* *matrix of the same size as Y and let $\lambda_i(B)$ denote the i^{th} largest singular value of B. Then*

$$\langle B, Y\rangle \leq \sum_{i=1}^{l} \lambda_i(B).$$

The following result follows from Corollary 5.2.1 and Claim 22. Recall that

$$\Delta(\mathcal{A}) := \max_{S\subseteq[m]} \sum_{i=1}^{\lfloor\sqrt{2|S|}+1\rfloor} \sigma_i(S).$$

where $\sigma_i(S)$ is the i'th largest singular value of $\frac{1}{|S|}\sum_{i\in S} A_i$.

We let Δ denote $\Delta(\mathcal{A})$ for the rest of the section.

Lemma 5.4.1. *Consider any extreme point solution $X(r)$ of $\mathbb{SDP}(r)$ such that* $\text{rank}(X(r)) > \text{tr}(X(r))$*. Let $X(r) = \sum_{j=1}^{r} \lambda_j v_j v_j^T$ be its eigenvalue decomposition and $X_f = \sum_{0<\lambda_j<1} \lambda_j v_j v_j^T$. Then there exists a constraint i such that $\langle F^T A_i F, X_f\rangle < \Delta$.*

Proof. Let $l = |S|$. From Corollary 5.2.1, it follows that number of fractional eigenvalues of $X(r)$ is at most $-\frac{1}{2} + \sqrt{2l + \frac{9}{4}} \leq \sqrt{2l} + 1$. Observe that $l > 0$ since $\text{rank}(X(r)) > \text{tr}(X(r))$. Thus $\text{rank}(X_f) \leq \sqrt{2l} + 1$. Moreover, $0 \preceq X_f \preceq I$, thus from Claim 22, we obtain that

$$\left\langle \sum_{j\in S} F^T A_j F, X_f \right\rangle \leq \sum_{i=1}^{\lfloor\sqrt{2l}+1\rfloor} \sigma_i\left(\sum_{j\in S} F^T A_j F\right) \leq \sum_{i=1}^{\lfloor\sqrt{2l}+1\rfloor} \sigma_i\left(\sum_{j\in S} A_j\right) \leq l\cdot\Delta$$

where the first inequality follows from Claim 22 and second inequality follows since the sum of top l singular values reduces after projection. But then we obtain, by averaging, that there exists $j \in S$ such that

$$\langle F^T A_j F, X_f\rangle < \frac{1}{l}\cdot l\Delta = \Delta$$

as claimed. □

Now we complete the proof of Theorem 5.1.8. Observe that the algorithm always maintains that end of each iteration, trace of X_f plus the rank of F_1 is at most d. Thus at the end of the algorithm, the returned solution has rank at most d. Next, consider the solution $X = F_1 F_1^T + F X_f F^T$ over the course of the algorithm. Again, it is easy to see that the objective value is non-increasing over the iterations. This follows since X_f defined at the end of an iteration is a feasible solution to the next iteration.

Now we argue the violation in any constraint i. While the constraint i remains in the SDP, the solution $X = F_1 F_1^T + F X_f F^T$ satisfies

$$\begin{aligned}&\langle A_i, X\rangle = \langle A_i, F_1 F_1^T\rangle + \langle A_i, F X_f F^T\rangle \\ =&\langle A_i, F_1 F_1^T\rangle + \langle F^T A_i F, X_f\rangle \leq \langle A_i, F_1 F_1^T\rangle + b_i - \langle A_i, F_1 F_1^T\rangle = b_i.\end{aligned}$$

where the inequality again follows since X_f is feasible with the updated constraints.

When constraint i is removed it might be violated by a later solution. At this iteration, $\langle F^T A_i F, X_f\rangle \leq \Delta$. Thus, $\langle A_i, F_1 F_1^T\rangle \geq b_i - \Delta$. In the final solution this bound can only go up as F_1 might only become larger. This completes the proof of theorem.

We now analyze the runtime of the algorithm which contains at most k iterations. Each iteration requires solving an SDP and eigenvector decompositions over $r \times r$ matrices, and recomputing F. The SDP has runtime $O(r^{6.5})$ which exceeds eigenvector decomposition and computing X_f, F takes $O(n^2)$. However, the result in Section 5.2 shows that $r \leq \sqrt{2k}$, and hence the total runtime of iterative rounding is $O(k^{4.25} + kn^2)$.

Application to FAIR-PCA . For the FAIR-PCA problem, iterative rounding recovers a rank-d solution whose variance goes down from the SDP solution by at most $\Delta(\{A_1^T A_1, \ldots, A_k^T A_k\})$. While this is no better than what we get by scaling (Corollary 5.3.1) for the max variance objective function, when we consider the marginal loss, i.e., the difference between the variance of the common d-dimensional solution and the best d-dimensional solution for each group, then iterative rounding can be much better. The scaling solution guarantee relies on the max-variance being a concave function and for the marginal loss, the loss for each group could go up proportional to the *largest* max variance (largest sum of top k singular values over the groups). With iterative rounding applied to the SDP solution, the loss Δ is the sum of only $O(\sqrt{k})$ singular values of the average of some subset of data matrices, so it can be better by as much as a factor of $\sqrt{k}$.

5.5 Polynomial Time Algorithm for Fixed Number of Groups

Functions of quadratic maps. We briefly summarize the approach of [GP05]. Let $f_1, \ldots, f_k : \mathbb{R}^n \to \mathbb{R}$ be real-valued quadratic functions in n variables. Let $p : \mathbb{R}^k \to \mathbb{R}$ be a polynomial of degree ℓ over some subring of $\mathbb{R}^k$ (e.g., the usual $(\times, +)$ or $(+, \min)$) The problem is to find all roots of the polynomial $p(f_1(x), f_2(x), \ldots, f_k(x))$, i.e., the set

$$Z = \{x \,:\, p(f_1(x), f_2(x), \ldots, f_k(x)) = 0\}.$$

First note that the set of solutions above is in general not finite and is some manifold and highly non-convex. The key idea of Grigoriev and Paleshnik (see also Barvinok [Bar93] for a similar idea applied to a special case) is to show that this set of solutions can be partitioned into a relatively small number of connected components such that there is an into map from these components to roots of a univariate polynomial of degree $(\ell n)^{O(k)}$; this therefore bounds the total number of components. The proof of this mapping is based on an explicit decomposition of space with the property that if a piece of the decomposition has a solution, it must be the solution of a linear system. The number of possible such linear systems is bounded as $n^{O(k)}$, and these systems can be enumerated efficiently.

The core idea of the decomposition starts with the following simple observation that relies crucially on the maps being quadratic (and not of higher degree).

Proposition 5.5.1. *The partial derivatives of any degree d polynomial p of quadratic forms $f_i(x)$, where $f_i : \mathbb{R}^n \to \mathbb{R}$, is linear in x for any fixed value of $\{f_1(x), \ldots, f_k(x)\}$.*

To see this, suppose $Y_j = f_j(x)$ and write

$$\frac{\partial p}{\partial x_i} = \sum_{j=1}^{k} \frac{\partial p(Y_1, \ldots, Y_k)}{\partial Y_j} \frac{\partial Y_j}{\partial x_i} = \sum_{j=1}^{k} \frac{\partial p(Y_1, \ldots, Y_k)}{\partial Y_j} \frac{\partial f_j(x)}{\partial x_i}.$$

Now the derivatives of f_j are linear in x_i as f_j is quadratic, and so for any fixed values of $Y_1, \ldots, Y_k$, the expression is linear in x.

The next step is a nontrivial fact about connected components of analytic manifolds that holds in much greater generality. Instead of all points that correspond to zeros of p, we look at all “critical" points of p defined as the set of points x for which the partial derivatives in all but the first coordinate, i.e.,

$$Z_c = \{x \, : \, \frac{\partial p}{\partial x_i} = 0, \quad \forall 2 \leq i \leq n\}.$$

The theorem says that Z_c will intersect every connected component of Z [GVJ88].

Now the above two ideas can be combined as follows. We will cover all connected components of Z_c. To do this we consider, for each fixed value of $Y_1, \ldots, Y_k$, the possible solutions to the linear system obtained, alongside minimizing x_1. The rank of this system is in general at least $n - k$ after a small perturbation (while [GP05] uses a deterministic perturbation that takes some care, we could also use a small random perturbation). So the number of possible solutions grows only as exponential in $O(k)$ (and not n), and can be effectively enumerated in time $(\ell d)^{O(k)}$. This last step is highly nontrivial, and needs the argument that over the reals, zeros from distinct components need only to be computed up to finite polynomial precision (as rationals) to keep them distinct. Thus, the perturbed version still covers all components of the original version. In this enumeration, we check for true solutions. The method actually works for any level set of p, $\{x \, : \, p(x) = t\}$ and not just its zeros. With this, we can optimize over p as well. We conclude this section by paraphrasing the main theorem from [GP05].

Theorem 5.5.2. *[GP05] Given k quadratic maps $q_1, \ldots, q_k : \mathbb{R}^k \to \mathbb{R}$ and a polynomial $p : \mathbb{R}^k \to \mathbb{R}$ over some computable subring of $\mathbb{R}$ of degree at most ℓ, there is an algorithm to compute a set of points satisfying $p(q_1(x), \ldots, q_k(x)) = 0$ that meets each connected component of the set of zeros of p using at most $(\ell n)^{O(k)}$ operations with all intermediate representations bounded by $(\ell n)^{O(k)}$ times the bit sizes of the coefficients of $p, q_1, \ldots, q_k$. The minimizer, maximizer or infimum of any polynomial $r(q_1(x), \ldots, q_k(x))$ of degree at most ℓ over the zeros of p can also be computed in the same complexity.*

5.5.1 Proof of Theorem 5.1.6

We apply Theorem 5.5.2 and the corresponding algorithm as follows. Our variables will be the entries of an $n \times d$ matrix P. The quadratic maps will be $f_i(P)$ plus additional maps for $q_{ii}(P) = \|P_i\|^2 - 1$ and $q_{ij}(P) = P_i^T P_j$ for columns P_i, P_j of P. The final polynomial is

$$p(f_1, \ldots, f_k, q_{11}, \ldots, q_{dd}) = \sum_{i \le j} q_{ij}(P)^2.$$

We will find the maximum of the polynomial $r(f_1, \ldots f_k) = g(f_1, \ldots, f_k)$ over the set of zeros of p using the algorithm of Theorem 5.5.2. Since the total number of variables is dn and the number of quadratic maps is $k + d(d+1)/2$, we get the claimed complexity of $O(\ell dn)^{O(k+d^2)}$ operations and this times the input bit sizes as the bit complexity of the algorithm.

5.6 Hardness

Theorem 5.6.1. *The* FAIR-PCA *problem:*

$$\max_{z\in\mathbb{R},P\in\mathbb{R}^{n\times d}} z \qquad \text{subject to} \tag{5.14}$$

$$\left\langle B_i, PP^T \right\rangle \geq z \qquad ,\forall i \in [k] \tag{5.15}$$

$$P^T P = I_d \tag{5.16}$$

for arbitrary $n \times n$ symmetric real PSD matrices $B_1, \ldots, B_k$ is NP-hard for $d = 1$ and $k = O(n)$.

Proof of Theorem 5.6.1: We reduce another NP-hard problem of MAX-CUT to the stated fair PCA problem. In MAX-CUT, given a simple graph $G = (V, E)$, we optimize

$$\max_{S\subseteq V} e(S, V \setminus S) \tag{5.17}$$

over all subset S of vertices. Here, $e(S, V \setminus S) = |\{e_{ij} \in E : i \in S, j \in V \setminus S\}|$ is the size of the cut S in G. As common NP-hard problems, the decision version of MAX-CUT:

$$\exists ? S \subseteq V : e(S, V \setminus S) \geq b \tag{5.18}$$

for an arbitrary $b > 0$ is also NP-hard. We may write MAX-CUT as an integer program as follows:

$$\exists ? v \in \{-1, 1\}^V : \frac{1}{2} \sum_{ij\in E} (1 - v_i v_j) \geq b \tag{5.19}$$

Here v_i represents whether a vertex i is in the set S or not:

$$v_i = \begin{cases} 1 & i \in S \\ -1 & i \notin S \end{cases} \tag{5.20}$$

and it can be easily verified that the objective represents the desired cut function.

We now show that this MAX-CUT integer feasibility problem can be formulated as an instance of the fair PCA problem (5.14)-(5.16). In fact, it will be formulated as a feasibility version of the fair PCA by checking if the optimal z of an instance is at least b. We choose $d = 1$ and $n = |V|$ for this instance, and we write $P = [u_1; \ldots ; u_n] \in \mathbb{R}^n$. The rest of the proof is to show that it is possible to construct constraints in the fair PCA form (5.15)-(5.16) to 1) enforce a discrete condition on u_i to take only two values, behaving similarly as v_i; and 2) check an objective value of MAX-CUT.

The reason u_i as written cannot behave exactly as v_i is that constraint (5.16) requires $\sum_{i=1}^n {u_i}^2 = 1$ but $\sum_{i=1}^n {v_i}^2 = n$. Hence, we scale the variables in MAX-CUT problem by writing $v_i = \sqrt{n}u_i$ and rearrange terms in (5.19) to obtain an equivalent formulation of MAX-CUT:

$$\exists ? u \in \left\{ -\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}} \right\}^n : n \sum_{ij \in E} -u_i u_j \geq 2b - |E| \tag{5.21}$$

We are now ready to give an explicit construction of $\{B_i\}_{i=1}^k$ to solve MAX-CUT formulation (5.21). Let $k = 2n + 1$. For each $j = 1, \ldots , n$, define

$$B_{2j-1} = bn \cdot \text{diag}(\mathbf{e_j}), B_{2j} = \frac{bn}{n-1} \cdot \text{diag}(\mathbf{1} - \mathbf{e_j})$$

where $\mathbf{e_j}$ and $\mathbf{1}$ denote vectors of length n with all zeroes except one at the jth coordinate, and with all ones, respectively. It is clear that B_{2j-1}, B_{2j} are PSD. Then for each $j = 1 \ldots , n$, the constraints $\left\langle B_{2j-1}, PP^T \right\rangle \geq b$ and $\left\langle B_{2j}, PP^T \right\rangle \geq b$ are equivalent to

$$u_j^2 \geq \frac{1}{n}, \text{ and } \sum_{i \neq j} u_j^2 \geq \frac{n-1}{n}$$

respectively. Combining these two inequalities with $\sum_{i=1}^n u_i^2 = 1$ forces both inequalities to be equalities, implying that $u_j \in \left\{-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}\right\}$ for all $j \in [n]$, as we aim.

Next, we set

$$B_{2n+1} = \frac{bn}{2b - |E| + n^2} \cdot (nI_n - A_G)$$

where $A_G = (\mathbb{I}[ij \in E])_{i,j\in[n]}$ is the adjacency matrix of the graph G. Since the matrix $nI_n - A_G$ is diagonally dominant and real symmetric, B_{2n+1} is PSD. We have that $\left\langle B_{2n+1}, PP^T \right\rangle \geq b$ is equivalent to

$$\frac{bn}{2b - |E| + n^2} \left(n \sum_{i=1}^{n} u_i^2 - \sum_{ij\in E} u_i u_j \right) \geq b$$

which, by $\sum_{i=1}^{n} u_i^2 = 1$, is further equivalent to

$$n \sum_{ij\in E} -u_i u_j \geq 2b - |E|$$

To summarize, we constructed $B_1, \ldots, B_{2n+1}$ so that checking whether an objective of fair PCA is at least b is equivalent to checking whether a graph G has a cut of size at least b, which is NP-hard. □

5.7 Integrality Gap

We showed that FAIR-PCA for $k = 2$ groups can be solved up to optimality in polynomial time using an SDP. For $k > 2$, we used a different, non-convex approach to get a polytime algorithm for any fixed k, d. Here we show that the SDP relaxation of FAIR-PCA has a gap even for $k = 3$ and $d = 1$.

Lemma 5.7.1. *The* FAIR-PCA *SDP relaxation:*

$$\begin{aligned} &\max\ z \\ \langle B_i, X\rangle &\geq z \quad i \in \{1,\ldots,k\} \\ \operatorname{tr}(X) &\leq d \\ 0 \preceq X &\preceq I \end{aligned}$$

for $k = 3$, $d = 1$, and arbitrary PSD $\{B_i\}_{i=1}^k$ contains a gap, i.e. the optimum value of the SDP relaxation is different from one of exact FAIR-PCA *problem.*

Proof of Lemma 5.7.1: Let $B_1 = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}, B_2 = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix}, B_3 = \begin{bmatrix} 2 & -1 \\ -1 & 2 \end{bmatrix}$. It can be checked that B_i are PSD. The optimum of the relaxation is $7/4$ (given by the optimal solution $X = \begin{bmatrix} 1/2 & 1/8 \\ 1/8 & 1/2 \end{bmatrix}$). However, an optimal exact FAIR-PCA solution is $\hat{X} = \begin{bmatrix} 16/17 & 4/17 \\ 4/17 & 1/17 \end{bmatrix}$ which gives an optimum $26/17$ (one way to solve for optimum rank-1 solution $\hat{X}$ is by parameterizing $\hat{X} = v(\theta)v(\theta)^T$ for $v(\theta) = [\cos\theta; \sin\theta]$, $\theta \in [0, 2\pi)$). □

5.8 Experiments

First, we note that experiments for FAIR-PCA with marginal loss objective for two groups were done in Samadi et al. [Sam+18]. Their algorithm outputs optimal solutions with exact rank, despite their weaker guarantee that the rank may be violated by at most 1. Hence, our result of Theorem 5.1.1 is the missing mathematical explanation of their empirical finding. We extend their experiments by solving MULTI-CRITERIA-DIMENSION-REDUCTION for more number of groups and objectives as follows.

We perform experiments using the algorithm as outlined in Section 5.2 on the Default Credit data set [YL09] for different target dimensions d. The data is partitioned into $k = 4, 6$ groups by education and gender, and preprocessed to have mean zero and same variance over features. Our algorithms are specified by two objectives for MULTI-CRITERIA-DIMENSION-REDUCTION problem introduced earlier: the marginal loss function and Nash social welfare. The code is publicly available at `https://github.com/SDPforAll/multiCriteriaDimReduction`. Figure 5.2 shows the marginal loss by our algorithms compared to a standard PCA on the entire dataset. Our algorithms significantly reduce "unfairness" in marginal loss of PCA that the standard PCA subtly introduces.

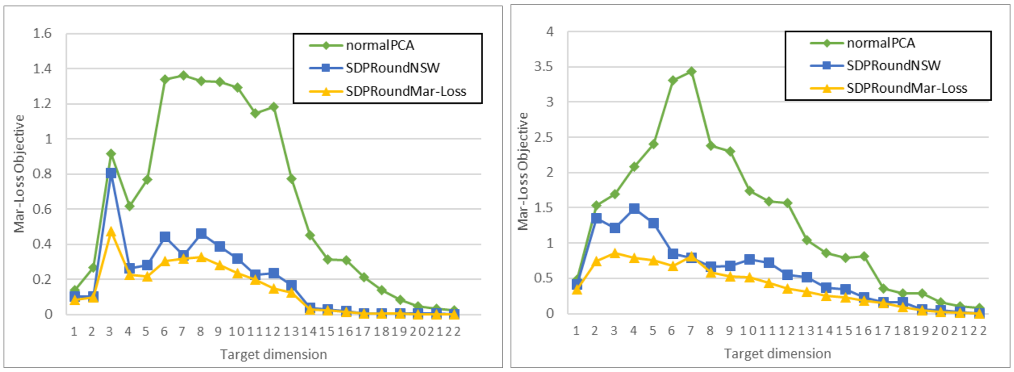

Figure 5.2: Marginal loss function of standard PCA compared to our SDP-based algorithms on Default Credit data. SDPRoundNSW and SDPRoundMar-Loss are two runs of the SDP-based algorithms maximizing NSW and minimizing marginal loss. Left: $k = 4$ groups. Right: $k = 6$.

In the experiments, extreme point solutions from SDPs enjoy lower rank violation than our worst-case guarantee. Indeed, while the guarantee is that the numbers of additional rank are at most $s = 1, 2$ for $k = 4, 6$, almost all SDP solutions have *exact* rank, and in rare cases when the solutions are not exact, the rank violation is only one. While our rank violation guarantee provably cannot be improved in general (due to the integrality gap in Section 5.7), this opens a question whether the guarantee is better for instances that arise in practice.

We also assess the performance of PCA with NSW objective, summarized in Figure 5.3. With respect to NSW, standard PCA performs marginally worse (about 10%) compared to our algorithms. It is worth noting from Figures 5.2 and 5.3 that our algorithms which try to optimize either

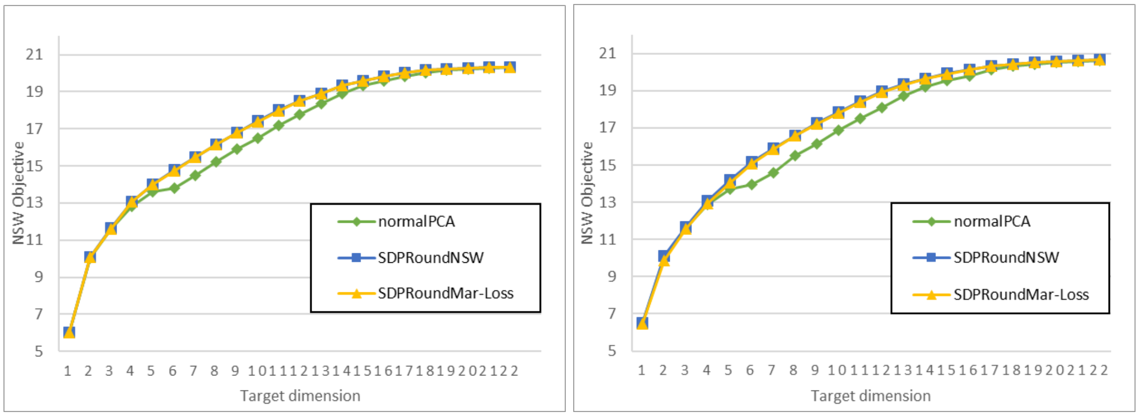

Figure 5.3: NSW objective of standard PCA compared to our SDP-based algorithms on Default Credit data. SDPRoundNSW and SDPRoundMar-Loss are two runs of the SDP algorithms maximizing NSW objective and minimizing maximum marginal loss. Left: $k = 4$ groups. Right: $k = 6$.

marginal loss function or NSW also perform well on the other fairness objective, making these PCAs promising candidates for fairness application.

Same experiments were done on the Adult Income data [UC]. Some categorial features are preprocessed into integers vectors and some features and rows with missing values are discarded. The final preprocessed data contains $m = 32560$ datapoints in $n = 59$ dimensions and is partitioned into $k = 5$ groups based on race. Figure 5.4 shows the performance of our SDP-based algorithms compared to standard PCA on marginal loss and NSW objectives. Similar to the Credit Data, optimizing for either marginal loss or NSW gives a PCA solution that also performs well in another criterion and performs better than the standard PCA in both objectives. Almost all SDP solutions are exact without any rank violation.

5.9 Scalability of the Algorithms

We found that the running time of solving SDP, which depends on n, is the bottleneck in all experiments. Each run (for one value of d) of the experiments is fast (< 0.5 seconds) on Default Credit data ($n = 23$), whereas a run on Adult Income data ($n = 59$) takes between 10 and 15 seconds on a single CPU. However, the runtime is not noticeably impacted by the numbers of data points and

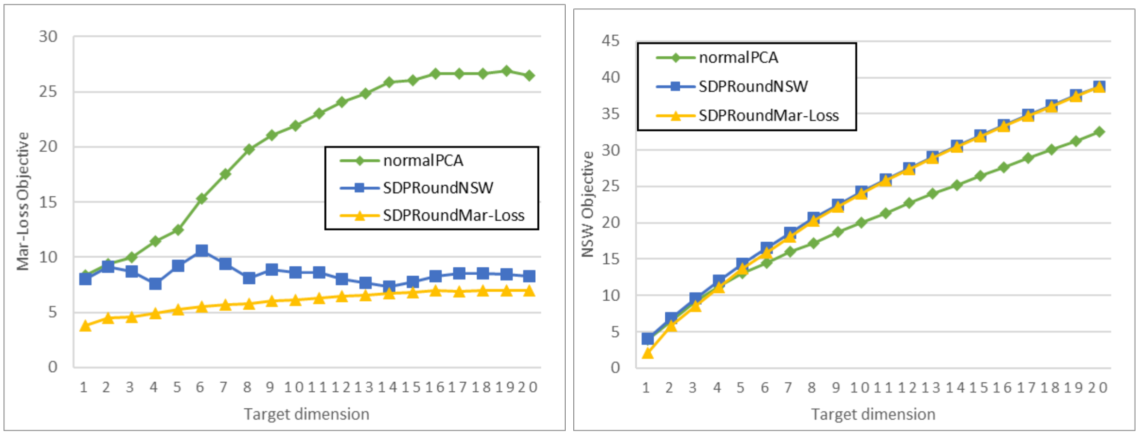

Figure 5.4: Marginal loss and NSW objective of standard PCA compared to our SDP-based algorithms on Adult Income data. SDPRoundNSW and SDPRoundMar-Loss are two runs of the SDP algorithms maximizing NSW objective and minimizing maximum marginal loss.

groups: larger m only increases the data preprocessing time to obtain $n \times n$ matrices and larger k increases the number of constraints. SDP solver and rounding algorithms can handle moderate number of affine constraints efficiently. This observation is as expected from the theoretical analysis.

In this section, we show two heuristics for solving the SDP relaxation that runs significantly faster in practice for large datasets: multiplicative weight update (MW) and Frank-Wolfe (FW). We also discuss several findings and considerations for implementing our algorithms in this thesis in practice. Both heuristics are publicly available at the following site: `https://github.com/SDPforAll/multiCriteriaDimReduction`.

For the rest of this section, we assume that the utility of each group is simply variance, $u_i(X) = \langle B_i, X \rangle$ where $B_i = A_i^T A_i$, and that $g(z_1, \ldots, z_k$ is a concave function of $z_1, \ldots, z_k$. When u_i is other linear function, we can model such different utility function by modifying g without changing the concavitiy of g. The SDP relaxation of MULTI-CRITERIA-DIMENSION-REDUCTION can be framed as SDP (5.22)-(5.25).

$$\max_{X\in\mathbb{R}^{n\times n}} g(z_1, z_2, \ldots, z_k) \text{ subject to} \tag{5.22}$$

$$z_i = \langle B_i, X\rangle \qquad \forall i = 1, 2, \ldots, k \tag{5.23}$$

$$\operatorname{tr}(X) \leq d \tag{5.24}$$

$$0 \preceq X \preceq I_n \tag{5.25}$$

5.9.1 Multiplicative Weight Update

One alternative method to solving (5.22)-(5.25) is multiplicative weight (MW) update [AHK12], suggested by [Sam+18] for solving FAIR-PCA problem for two groups in order to improve runtime. Though this prior works [AHK12, Sam+18] have theoretical guarantee, in practice the learning rate is tuned mre aggressively and the algorithm becomes a heuristic without any certificate of optimality. We show the primal-dual derivation of Multiplicative Weight, which provides the primal-dual gap to certify optimality.

We take the Lagragian dual on (5.23) to obtain that the optimum of the SDP equals to

$$\max_{\substack{X\in\mathbb{R}^{n\times n}\\ z\in\mathbb{R}^n\\ \operatorname{tr}(X)=d\\ 0\preceq X\preceq I}} \inf_{w\in\mathbb{R}^k} g(z) + \sum_{i=1}^{k} w_i \left(\langle B_i, X\rangle - z_i\right)$$

By strong duality, we may swap $\max$ and $\inf$. After rearranging, the optimum of the SDP equals

$$\inf_{w\in\mathbb{R}^k}\left(\max_{\substack{X\in\mathbb{R}^{n\times n}\\ \operatorname{tr}(X)=d, 0\preceq X\preceq I}} \sum_{i=1}^{k} w_i \langle B_i, X\rangle - \min_{z\in\mathbb{R}^n}\left(w^T z - g(z)\right)\right) \tag{5.26}$$

The optimization

$$\max_{\substack{X\in\mathbb{R}^{n\times n}\\ \operatorname{tr}(X)=d, 0\preceq X\preceq I}} \sum_{i=1}^{k} w_i \langle B_i, X\rangle \tag{5.27}$$

in (5.26) can easily be computed by standard PCA on weighted data $\sum_{i=1}^{k} w_i \cdot B_i$ projecting from n to d dimensions. The term (5.27) is also convex in w, as it is a maximum of linear functions. The term $\min_{z\in\mathbb{R}^n} \left(w^T z - g(z)\right)$ is also known as concave conjugate of g, which we will denote by $g_*(w)$. It is also known that $g_*(w)$ is a concave function (as it is a minimum of linear functions). Hence, (5.26) is a convex optimization problem.

Solving the dual problem (5.26) depends of the form of $g_*(w)$. For each fairness criteria outlined in this paper, we summarize the form of $g_*(w)$ below.

Max-Min Variance (FAIR-PCA or MM-Var) : fairness objective $g(z) = \min_{i\in[k]} z_i$ gives

$$g_*(w) = \begin{cases} 0 & \text{if } w \geq 0, \sum_{i=1}^{k} w_i = 1 \\ -\infty & \text{otherwise} \end{cases}$$

Min-Max Loss (MM-Loss) : fairness objective (recall (5.1)) $g(z) = \min_{i\in[k]} z_i - \beta_i$, where $\beta_i = \max_{Q\in\mathcal{P}_d} \|A_i Q\|_F^2$ is the best possible variance the group i can have, gives

$$g_*(w) = \begin{cases} \sum_{i=1}^{k} w_i\beta_i & \text{if } w \geq 0, \sum_{i=1}^{k} w_i = 1 \\ -\infty & \text{otherwise} \end{cases}$$

More generally, the above form of $g_*(w)$ holds for any constants β_i's. For example, this calculation also captures Min-Max reconstruction error: $g(X) = \min_{i\in[k]} -\|A_i - A_i P\|_F^2 = \min_{i\in[k]} z_i - \operatorname{tr}(B_i)$ (recall that $X = PP^T$, $B_i = A_i^T A_i$, and $z_i = \langle B_i, X\rangle$).

Nash Social Welfare (NSW) : fairness objective $g(z) = \sum_{i=1}^k \log(z_i)$ gives

$$g_*(w) = \begin{cases} \sum_{i=1}^k (1 + \log w_i) & \text{if } w > 0 \\ -\infty & \text{otherwise} \end{cases}$$

For fairness criteria in the "max-min" type, including MM-Var and MM-Loss, the dual reduces to solving an optimization over a simplex with standard PCA as the function evaluation oracle. Solving an optimization over a simplex can be done using mirror descent [NY83] with entropy potential function $R(w) = \sum_{i=1}^k w_i \log w_i$. Such optimization is algorithmically identical to multiplicative weight update by [AHK12]; however, with primal-dual formulation, the dual solution w_i obtained in each step of mirror descent can be used to calculate the dual objective in (5.26), and the optimum X in (5.27) is used to calculate the primal objective. The algorithm runs iteratively until the duality gap satisfies a set threshold of choice.

5.9.2 Frank-Wolfe

It is worth noting that while the original optimization (5.22)-(5.25), which is in the form

$$\max_{\substack{X \in \mathbb{R}^{n \times n} \\ \mathrm{tr}(X) = d, 0 \preceq X \preceq I}} \quad g(z(X))$$

where the utility z is a function of projection matrix X is a nontrivial convex optimization, its linear counterpart

$$\max_{\substack{X \in \mathbb{R}^{n \times n} \\ \mathrm{tr}(X) = d, 0 \preceq X \preceq I}} \quad \langle C, X \rangle$$

is solvable by standard PCA for any given matrix C. This motivates Frank-Wolfe (FW) algorithm [FW56] which requires a linear oracle (solving the problem with a linear objective) in each step. The instantiation of FW to MULTI-CRITERIA-DIMENSION-REDUCTION is summarized in Algorithm 5.1. We note the simpler linear oracle step in FW.

Algorithm 5.1 Frank-Wolfe Algorithm for Multi-Criteria Dimensionality Reduction

```
Input: $B_1, \ldots, B_k \in \mathbb{R}^{n\times n}$, $d \le n$, concave $g : \mathbb{R}^k \to \mathbb{R}$, learning rate $\eta_t$, duality gap target
Output: A matrix $X \in \mathbb{R}^{n\times n}$ that maximizes $g(\langle B_1, X\rangle, \ldots, \langle B_k, X\rangle)$ subject to $\mathrm{tr}(X) = d, 0 \preceq X \preceq I$
Initialize a feasible $X_0$ (we use $X_0 = \frac{d}{n} I_n$), $t = 0$
while duality gap exceeds the target do
    $G_t \leftarrow \nabla_X g(X_t)$
    $S_t \leftarrow VV^T$ where $V$ is $n$-by-$d$ matrix of top $d$ eigenvectors of $G_t$    ▷ Linear oracle of FW
    $X_{t+1} \leftarrow (1-\eta_t)x_t + \eta_t S_t$
    $g_t \leftarrow (S_t - X_t) \cdot G_t$    ▷ Duality gap
    $t \leftarrow t+1$
Output $X_t$
```

One additional concern for implementing FW is obtaining gradient $\nabla_X g(X_t)$. For some objectives such as NSW, this gradient can be calculated directly (some small error may need to be added to stabalize the algorithm from exploding gradient when the variance is close to zero). Other objectives, such as MM-Var and MM-Loss, on the other hand, is not differentiable. Though one may try to still use FW, there is no theoretical guarantee in the literature for the convergence of maximizing concave non-differentiable function, even when the feasible set is compact as in our SDP relaxation.

5.9.3 Parameter Tuning

Multiplicative Weight Update. In practice for MM-Var and MM-Loss objectives, we tune the learning rate of mirror descent much higher than in theory. For NSW, the dual is still a convex optimization, so standard technique such as gradient descent can be used. We found that in practice, however, the unboundedness of the feasible set and the exploding gradient when w_i's are close to zero pose a challenge to tune the algorithm to converge quickly.

MW for Two Groups. For for MM-Var and MM-Loss objectives in two groups, the simplex is a segment $[0,1]$. The dual problem (5.26) reduces to

$$\inf_{w\in[0,1]}\left(\max_{\substack{X\in\mathbb{R}^{n\times n}\\ \operatorname{tr}(X)=d,0\preceq X\preceq I}}\langle wB_1+(1-w)B_2,X\rangle\right) \tag{5.28}$$

The function

$$h(w):=\max_{\substack{X\in\mathbb{R}^{n\times n}\\ \operatorname{tr}(X)=d,0\preceq X\preceq I}}\langle wB_1+(1-w)B_2,X\rangle$$

is a maximum of piecewise linear functions $\langle wB_1+(1-w)B_2,X\rangle$ in w, and hence is convex on w. Instead of mirror descent, one can apply ternary search, a technique applicable to maximizing convex function in one dimension in general, to solve (5.28). However, we claim that binary search is also a valid choice.

First, because $h(w)$ is convex, we may assume that h achieves minimum at $w=w^*$ and that all subgradients $\partial h(w)\subseteq(-\infty,0]$ for all $w<w^*$ and $\partial h(w)\subseteq[0,\infty)$ for all $w>w^*$. In the binary search algorithm with current iterate $w=w_t$, let

$$X_t\in\operatorname*{argmax}_{\substack{X\in\mathbb{R}^{n\times n}\\ \operatorname{tr}(X)=d,0\preceq X\preceq I}}\langle w_tB_1+(1-w_t)B_2,X\rangle$$

be any solution of the optimization (which can be implemented easily by the standard PCA). Because a linear function $\langle wB_1+(1-w)B_2,X_t\rangle=\langle B_2,X_t\rangle+w\langle B_1-B_2,X_t\rangle$ is a lower bound of $h(w)$ and h is convex, we have $\langle B_1-B_2,X_t\rangle\in\partial h(w_t)$. Therefore, the binary search algorithm can check the sign of $\langle B_1-B_2,X_t\rangle$ for a correct recursion. If $\langle B_1-B_2,X_t\rangle<0$, then $w^*>w_t$; if $\langle B_1-B_2,X_t\rangle>0$, then $w^*<w_t$; and the algorithm recurses in the left half or right half of the current segment accordingly. If $\langle B_1-B_2,X_t\rangle=0$, then w_t is an optimum dual solution.

Frank-Wolfe. In practice, we experiment with more aggressive learning rate schedule and line search algorithm. We found that FW converges quickly for NSW objective. However, FW does not converge for MM-Var and MM-Loss for any learning rate schedule, including the standard $\eta_t = \frac{1}{t+2}$, and line search. There is modification of FW which has convergence guarantee for maximizing concave non-differentiable functions. It is still an open question on this thesis whether some of those, if any, can speed up the SDP relaxation of MULTI-CRITERIA-DIMENSION-REDUCTION problem.

5.9.4 Practical Considerations and Findings

Extreme Property of SDP Relaxation Solution. We note that a solution for SDP relaxation (5.22)-(5.25) obtained by any of the algorithm (MW, FW, or SDP solver) are already extreme in practice. This is because with probability 1 over random datasets, SDP is not degenerate, and hence have a unique optimal solution. Since any linear optimization over a compact, convex set must have an extreme optimal solution, this optimal solution is necessarily extreme. Therefore, in practice, it is not necessary to apply the SDP rounding algorithm to the solution of SDP relaxation.

Rank Violation of Extreme SDP Relaxation Solution. While the rank violation bound of $\lfloor\sqrt{2k+\frac{1}{4}}-\frac{3}{2}\rfloor$ stated in Theorem 5.1.3 is tight (tight examples in [Pat98] can be applied in our settings), the rank violation in our experiments up to 16 groups are mostly zero, i.e. we obtain an exact solution. In rare cases where the solution is not exact, the rank violation is one. As a result, in all experiments we begin by solving the SDP relaxation targeting dimension d. If the solution is exact, then we are done. Else, we target dimension $d-1$ and check if the solution is of rank at most d. If not, we continue to target dimension $d-2, d-3, \ldots$ until the solution of SDP relaxation has rank at most d.

5.9.5 Runtime Results

We next perform MW and FW heuristics on a larger 1940 Colorado Census dataset [AA]. The census data is preprocessed by one-hot encoding all discrete columns, ignoring columns with N/A,

Table 5.1: Runtime of MW and FW for solving MULTI-CRITERIA-DIMENSION-REDUCTION on different fairness objectives and numbers of dimensions in original data. Times reported are in second(s).

Original Dimensions	MM-Var (by MW)	MM-Loss (by MW)	NSW (by FW)
$n = 1000$	77	65	15
$n = 2000$	585	589	69

and normalizing the data to mean zero and variance one. The preprocessed dataset contain 661k datapoints and 7284 columns. Data are partitioned into 16 groups based on 2 genders and 8 education levels. We solve the SDP relaxation of MULTI-CRITERIA-DIMENSION-REDUCTION with MM-Var, MM-Loss, and NSW objectives until obtain a certificate of duality gap of no more than 0.1% (in the case of NSW, the product of variances, rather than the sum of logarithmic of variances, are used to calculate this gap). The runtime results, in seconds, are in shown in Table 5.1. When n increases, the bottleneck of the experiment became the standard PCA itself. Since speeding up the standard PCA is out of the scope of this work, we capped the original dimension of data by selecting the first n dimensions out of 7284, so that the standard PCA can still be performed in a reasonable amount of time.

Empirical Performance of MW. We found that MM-Var and MM-Loss objectives are solved by efficiently by MW, whereas MW with gradient descent on the dual of NSW does not converge quickly. For the Census Dataset, after parameter tuning, MW runs 100-200 iterations on both objectives. MW for both Credit and Income datasets ($n = 23, 59$) on 4-6 groups with both objectives runs 10-20 iterations, giving a total runtime of is less than few seconds. Therefore, the price of fairness in PCA for MW-Var and MM-Loss objectives is 100-200x runtime for large datasets, and 10-20x runtime for medium datasets, as compared to the standard PCA without fairness constraint.

Empirical Performance of FW. FW converges quickly for NSW objective, and does not converge on MM-Var or MM-Loss objectives. FW terminates in 10-20 iterations for Census Data, where the standard PCA oracle is the bottleneck in each iteration. Therefore, the price of fairness in PCA for NSW objective is 10-20x runtime compared to the standard PCA without fairness constraint.

It is still an open question in this work to explore other heuristics to speed up solving MULTI-CRITERIA-DIMENSION-REDUCTION in practice. It is still open if some (if any) modification of FW may work well for non-differentiable objectives, or if a modification of MW will improve the runtime further for any of the three objectives.

CHAPTER 6
CONCLUSION

This thesis presents novel applications and extensions of convex relaxations for different contexts, namely in diverse subset selection and multi-criteria dimensionality reduction which is motivated from fairness in Principle Component Analysis (PCA). Convex relaxations we use include SDPs, convex programs over polytopes, and convex programs with linear and nonlinear objective over PSD feasible sets. Convex relaxations are intermediate steps of the problem that can efficiently give fractional solutions. We present novel rounding scheme to obtain the original feasible solutions from fractional ones, which includes novel sampling distribution, and show their efficiency. In an application of SDPs, we show that extreme solutions of relaxations *themselves* already have desired properties and no rounding is needed.

Moreover, analyzing convex relaxations and their dual problems gives lower bound on approximability of the problems. Integrality gaps of convex relaxations shows that better approximation ratio from any rounding scheme does not exist, such as our tightness result for E-optimal design. Dual problems of the relaxations can be used to approximate the value of optimum, allowing us to prove the approximation guarantee, even without solving the dual problems. This technique, called *dual-fitting*, gives the best approximation results known for D-optimal design by a simple, widely-used combinatorial algorithm.

Finally, solving the problems as convex programs allows us to consider wider range of tools from convex optimization. We are able to scale algorithms for multi-criteria dimensionality reduction in practice using convex optimization methods, giving our work both theoretical performance and empirical success.

REFERENCES

[ALSW17b] Z. Allen-Zhu, Y. Li, A. Singh, and Y. Wang. “Near-Optimal Design of Experiments via Regret Minimization”. In: *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*. Ed. by D. Precup and Y. W. Teh. Vol. 70. Proceedings of Machine Learning Research. PMLR, 2017, pp. 126–135.

[ALSW17a] Z. Allen-Zhu, Y. Li, A. Singh, and Y. Wang. “Near-Optimal Discrete Optimization for Experimental Design: A Regret Minimization Approach”. In: *arXiv preprint arXiv:1711.05174* (2017).

[Alo86] N. Alon. “Eigenvalues and expanders”. In: *Combinatorica* 6.2 (1986). Theory of computing (Singer Island, Fla., 1984), pp. 83–96.

[AG17] N. Anari and S. O. Gharan. “A generalization of permanent inequalities and applications in counting and optimization”. In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*. Ed. by H. Hatami, P. McKenzie, and V. King. ACM, 2017, pp. 384–396.

[ALMK18] J. Angwin, J. Larson, S. Mattu, and L. Kirchner. *Machine Bias - ProPublica*. `https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing`. 2018.

[AA] T. U. N. Archives and R. Administration. https://1940census.archives.gov/index.asp. Accessed on March 27th, 2020.

[AHK12] S. Arora, E. Hazan, and S. Kale. “The Multiplicative Weights Update Method: a Meta-Algorithm and Applications.” In: *Theory of Computing* 8.1 (2012), pp. 121–164.

[ADT07] A. Atkinson, A. Donev, and R. Tobias. *Optimum experimental designs, with SAS*. Vol. 34. Oxford University Press, 2007.

[AB13] H. Avron and C. Boutsidis. “Faster subset selection for matrices and applications”. In: *SIAM Journal on Matrix Analysis and Applications* 34.4 (2013), pp. 1464–1499.

[BBV16] A. S. Bandeira, N. Boumal, and V. Voroninski. “On the low-rank approach for semidefinite programs arising in synchronization and community detection”. In: *Conference on learning theory*. 2016, pp. 361–382.

[BS06] N. Bansal and M. Sviridenko. “The santa claus problem”. In: *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*. ACM. 2006, pp. 31–40.

[Bar93] A. I. Barvinok. “Feasibility testing for systems of real quadratic equations”. In: *Discrete & Computational Geometry* 10.1 (1993), pp. 1–13.

[Bar95] A. I. Barvinok. “Problems of distance geometry and convex properties of quadratic maps”. In: *Discrete & Computational Geometry* 13.2 (1995), pp. 189–202.

[BSS12a] J. Batson, D. A. Spielman, and N. Srivastava. “Twice-ramanujan sparsifiers”. In: *SIAM Journal on Computing* 41.6 (2012), pp. 1704–1721.

[BSS12b] J. D. Batson, D. A. Spielman, and N. Srivastava. “Twice-Ramanujan Sparsifiers”. In: *SIAM J. Comput.* 41.6 (2012), pp. 1704–1721.

[BTN01] A. Ben-Tal and A. Nemirovski. *Lectures on modern convex optimization: analysis, algorithms, and engineering applications*. Vol. 2. Siam, 2001.

[Ber05] D. S. Bernstein. *Matrix mathematics: Theory, facts, and formulas with application to linear systems theory*. Vol. 41. Princeton university press Princeton, 2005.

[BKS15] A. Bhangale, S. Kopparty, and S. Sachdeva. “Simultaneous approximation of constraint satisfaction problems”. In: *International Colloquium on Automata, Languages, and Programming*. Springer. 2015, pp. 193–205.

[Bha+18] A. Bhangale, S. Khot, S. Kopparty, S. Sachdeva, and D. Thimvenkatachari. “Near-optimal approximation algorithm for simultaneous M ax-C ut”. In: *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics. 2018, pp. 1407–1425.

[Bol+16] T. Bolukbasi, K.-W. Chang, J. Y. Zou, V. Saligrama, and A. T. Kalai. “Man is to computer programmer as woman is to homemaker? debiasing word embeddings”. In: *Advances in neural information processing systems*. 2016, pp. 4349–4357.

[BVB16] N. Boumal, V. Voroninski, and A. Bandeira. “The non-convex Burer-Monteiro approach works on smooth semidefinite programs”. In: *Advances in Neural Information Processing Systems*. 2016, pp. 2757–2765.

[BT87] J. Bourgain and L. Tzafriri. “Invertibility of Large Submatrices with Applications to the Geometry of Banach Spaces and Harmonic Analysis”. In: *Israel journal of mathematics* 57.2 (1987), pp. 137–224.

[BDM11] C. Boutsidis, P. Drineas, and M. Magdon-Ismail. “Sparse Features for PCA-Like Linear Regression”. In: *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain.* Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. C. N. Pereira, and K. Q. Weinberger. 2011, pp. 2285–2293.

[BMI13] C. Boutsidis and M. Magdon-Ismail. “Deterministic feature selection for k-means clustering”. In: *IEEE Transactions on Information Theory* 59.9 (2013), pp. 6099–6110.

[BV04] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[BM61] P. Bullen and M. Marcus. “Symmetric Means and Matrix Inequalities”. In: *Proceedings of the American Mathematical Society* 12.2 (1961), pp. 285–290.

[BG18] J. Buolamwini and T. Gebru. “Gender shades: Intersectional accuracy disparities in commercial gender classification”. In: *Conference on Fairness, Accountability and Transparency*. 2018, pp. 77–91.

[BM03] S. Burer and R. D. Monteiro. “A nonlinear programming algorithm for solving semidefinite programs via low-rank factorization”. In: *Mathematical Programming* 95.2 (2003), pp. 329–357.

[BM05] S. Burer and R. D. Monteiro. “Local minima and convergence in low-rank semidefinite programming”. In: *Mathematical Programming* 103.3 (2005), pp. 427–444.

[CDKV16] L. E. Celis, A. Deshpande, T. Kathuria, and N. K. Vishnoi. “How to be Fair and Diverse?” In: *arXiv preprint arXiv:1610.07183* (2016).

[Cel+18] L. E. Celis, V. Keswani, D. Straszak, A. Deshpande, T. Kathuria, and N. K. Vishnoi. “Fair and diverse DPP-based data summarization”. In: *arXiv preprint arXiv:1802.04023* (2018).

[Che52] H. Chernoff. “A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations”. In: *The Annals of Mathematical Statistics* (1952), pp. 493–507.

[CKLV17] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii. “Fair clustering through fairlets”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 5029–5037.

[CR18] A. Chouldechova and A. Roth. “The frontiers of fairness in machine learning”. In: *arXiv preprint arXiv:1810.08810* (2018).

[ÇMI09] A. Çivril and M. Magdon-Ismail. “On selecting a maximum volume sub-matrix of a matrix and related problems”. In: *Theoretical Computer Science* 410.47-49 (2009), pp. 4801–4811.

[CN80] R. D. Cook and C. J. Nachtrheim. “A comparison of algorithms for constructing exact D-optimal designs”. In: *Technometrics* 22.3 (1980), pp. 315–324.

[CLRS09] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2009.

[Cra17] K. Crawford. *The Trouble with Bias*. Invited Talk by Kate Crawford at NIPS 2017, Long Beach, CA. 2017.

[Dav57] C. Davis. “All convex invariant functions of hermitian matrices”. In: *Arch. Math.* 8 (1957), pp. 276–278.

[Deb14] K. Deb. “Multi-objective optimization”. In: *Search methodologies*. Springer, 2014, pp. 403–449.

[DW17a] M. Dereziński and M. K. Warmuth. “Subsampling for Ridge Regression via Regularized Volume Sampling”. In: *arXiv preprint arXiv:1710.05110* (2017).

[DW17b] M. Dereziński and M. K. Warmuth. “Unbiased estimates for linear regression via volume sampling”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 3087–3096.

[DR10] A. Deshpande and L. Rademacher. “Efficient Volume Sampling for Row/Column Subset Selection”. In: *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2010, October 23-26, 2010, Las Vegas, Nevada, USA*. IEEE Computer Society, 2010, pp. 329–338.

[DRVW06] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. “Matrix approximation and projective clustering via volume sampling”. In: *Theory of Computing* 2.1 (2006), pp. 225–247.

[DV06] A. Deshpande and S. Vempala. “Adaptive sampling and fast low-rank matrix approximation”. In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2006, pp. 292–303.

[DMIMW12] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff. “Fast approximation of matrix coherence and statistical leverage”. In: *Journal of Machine Learning Research* 13.Dec (2012), pp. 3475–3506.

[FB04] Z. Fang and B. Bensaou. “Fair bandwidth sharing algorithms based on game theory frameworks for wireless ad hoc networks”. In: *IEEE infocom*. Vol. 2. Citeseer. 2004, pp. 1284–1295.

[Fed+55] W. T. Federer et al. Macmillan Co., New York and London, 1955.

[Fed72] V. V. Fedorov. *Theory of optimal experiments*. Elsevier, 1972.

[FW56] M. Frank and P. Wolfe. “An algorithm for quadratic programming”. In: *Naval research logistics quarterly* 3.1-2 (1956), pp. 95–110.

[GJ79] M. R. Garey and D. S. Johnson. *Computers and intractability*. A guide to the theory of NP-completeness, A Series of Books in the Mathematical Sciences. W. H. Freeman and Co., San Francisco, Calif., 1979, pp. x+338. ISBN: 0-7167-1045-5.

[GSS11] S. O. Gharan, A. Saberi, and M. Singh. “A randomized rounding approach to the traveling salesman problem”. In: *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*. IEEE. 2011, pp. 550–559.

[GBS08] A. Ghosh, S. Boyd, and A. Saberi. “Minimizing effective resistance of a graph”. In: *SIAM review* 50.1 (2008), pp. 37–66.

[GRSZ14] F. Grandoni, R Ravi, M. Singh, and R. Zenklusen. “New approaches to multi-objective optimization”. In: *Mathematical Programming* 146.1-2 (2014), pp. 525–554.

[GVJ88] D. Y. Grigor’ev and N. Vorobjov Jr. “Solving systems of polynomial inequalities in subexponential time”. In: *Journal of Symbolic Computation* 5.1-2 (1988), pp. 37–64.

[GP05] D. Grigoriev and D. V. Pasechnik. “Polynomial-time computing over quadratic maps i: sampling in real algebraic sets”. In: *Computational complexity* 14.1 (2005), pp. 20–52.

[GS12] V. Guruswami and A. K. Sinop. “Optimal column-based low-rank matrix reconstruction”. In: *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms*. SIAM. 2012, pp. 1207–1214.

[Har97] D. A. Harville. *Matrix algebra from a statistician’s perspective*. Vol. 1. Springer, 1997.

[HTF09] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.

[HK70] A. E. Hoerl and R. W. Kennard. “Ridge regression: Biased estimation for nonorthogonal problems”. In: *Technometrics* 12.1 (1970), pp. 55–67.

[HM07] F. de Hoog and R. Mattheij. “Subset selection for matrices”. In: *Linear Algebra and its Applications* 422.2-3 (2007), pp. 349–359.

[HM11] F. de Hoog and R. Mattheij. “A note on subset selection for matrices”. In: *Linear Algebra and its Applications* 434.8 (2011), pp. 1845–1850.

[Hot33] H. Hotelling. “Analysis of a complex of statistical variables into principal components.” In: *Journal of educational psychology* 24.6 (1933), p. 417.

[IP91] A. F. Iezzoni and M. P. Pritts. “Applications of principal component analysis to horticultural research”. In: *HortScience* 26.4 (1991), pp. 334–338.

[Jol86] I. T. Jolliffe. “Principal component analysis and factor analysis”. In: *Principal component analysis*. Springer, 1986, pp. 115–128.

[JB09] S. Joshi and S. Boyd. “Sensor selection via convex optimization”. In: *IEEE Transactions on Signal Processing* 57.2 (2009), pp. 451–462.

[Kah96] J. Kahn. “Asymptotics of the chromatic index for multigraphs”. In: *journal of combinatorial theory, Series B* 68.2 (1996), pp. 233–254.

[Kah00] J. Kahn. “Asymptotics of the list-chromatic index for multigraphs”. In: *Random Structures & Algorithms* 17.2 (2000), pp. 117–156.

[Kal77] E. Kalai. “Proportional solutions to bargaining situations: interpersonal utility comparisons”. In: *Econometrica: Journal of the Econometric Society* (1977), pp. 1623–1630.

[KS+75] E. Kalai, M. Smorodinsky, et al. “Other solutions to Nash bargaining problem”. In: *Econometrica* 43.3 (1975), pp. 513–518.

[KN79] M. Kaneko and K. Nakamura. “The Nash social welfare function”. In: *Econometrica: Journal of the Econometric Society* (1979), pp. 423–435.

[KMM15] M. Kay, C. Matuszek, and S. A. Munson. “Unequal representation and gender stereotypes in image search results for occupations”. In: *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM. 2015, pp. 3819–3828.

[KAM19] M. Kleindessner, P. Awasthi, and J. Morgenstern. “Fair k-Center Clustering for Data Summarization”. In: *Proceedings of the 36th International Conference on Machine Learning*. 2019.

[KSAM19] M. Kleindessner, S. Samadi, P. Awasthi, and J. Morgenstern. “Guarantees for Spectral Clustering with Fairness Constraints”. In: *Proceedings of the 36th International Conference on Machine Learning*. 2019.

[Kru64] J. B. Kruskal. “Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis”. In: *Psychometrika* 29.1 (1964), pp. 1–27.

[KT+12] A. Kulesza, B. Taskar, et al. “Determinantal point processes for machine learning”. In: *Foundations and Trends® in Machine Learning* 5.2–3 (2012), pp. 123–286.

[LRS11] L. C. Lau, R. Ravi, and M. Singh. *Iterative methods in combinatorial optimization*. Vol. 46. Cambridge University Press, 2011.

[LJS17] C. Li, S. Jegelka, and S. Sra. “Column Subset Selection via Polynomial Time Dual Volume Sampling”. In: *arXiv preprint arXiv:1703.02674* (2017).

[LPYZ19] H. Li, S. Patterson, Y. Yi, and Z. Zhang. “Maximizing the number of spanning trees in a connected graph”. In: *IEEE Transactions on Information Theory* (2019).

[LT93] M. Lin and N. S. Trudinger. *On Some Inequalities for Elementary Symmetric Function*. Australian National University, School of Mathematical Sciences, Centre for Mathematics and its Applications, 1993.

[MH08] L. v. d. Maaten and G. Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.

[MSTX19] V. Madan, M. Singh, U. Tantipongpipat, and W. Xie. “Combinatorial Algorithms for Optimal Design”. In: *Conference on Learning Theory*. 2019, pp. 2210–2258.

[ML57] M. Marcus and L. Lopes. “Inequalities for symmetric functions and Hermitian matrices”. In: *Canad. J. Math.* 9 (1957), pp. 305–312.

[MS17] Z. E. Mariet and S. Sra. “Elementary Symmetric Polynomials for Optimal Experimental Design”. In: *Advances in Neural Information Processing Systems*. 2017, pp. 2136–2145.

[MS75] D. W. Marquardt and R. D. Snee. “Ridge regression in practice”. In: *The American Statistician* 29.1 (1975), pp. 3–20.

[Mey73] C. D. Meyer Jr. “Generalized inversion of modified matrices”. In: *SIAM Journal on Applied Mathematics* 24.3 (1973), pp. 315–323.

[MN94] A. J. Miller and N.-K. Nguyen. “Algorithm AS 295: A Fedorov exchange algorithm for D-optimal design”. In: *Journal of the royal statistical society. series c (applied statistics)* 43.4 (1994), pp. 669–677.

[MMJ70] T. J. Mitchell and F. Miller Jr. “Use of design repair to construct designs for special linear models”. In: *Math. Div. Ann. Progr. Rept.(ORNL-4661)* 13 (1970).

[NY17] A. Naor and P. Youssef. “Restricted Invertibility Revisited”. In: *A Journey Through Discrete Mathematics: A Tribute to Jiří Matoušek*. Cham: Springer International Publishing, 2017, pp. 657–691. ISBN: 978-3-319-44479-6.

[NJ50] J. F. Nash Jr. “The bargaining problem”. In: *Econometrica: Journal of the Econometric Society* (1950), pp. 155–162.

[NY83] A. S. Nemirovsky and D. B. Yudin. “Problem complexity and method efficiency in optimization.” In: (1983).

[NM92] N.-K. Nguyen and A. J. Miller. “A review of some exchange algorithms for constructing discrete D-optimal designs”. In: *Computational Statistics & Data Analysis* 14.4 (1992), pp. 489–498.

[Nik15] A. Nikolov. “Randomized rounding for the largest simplex problem”. In: *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*. ACM. 2015, pp. 861–870.

[NS16] A. Nikolov and M. Singh. “Maximizing determinants under partition constraints”. In: *STOC*. 2016, pp. 192–201.

[NST19] A. Nikolov, M. Singh, and U. T. Tantipongpipat. “Proportional volume sampling and approximation algorithms for A-optimal design”. In: *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*. SIAM. 2019, pp. 1369–1386.

[NT15] A. Nikolov and K. Talwar. “Approximating Hereditary Discrepancy via Small Width Ellipsoids”. In: *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*. Ed. by P. Indyk. SIAM, 2015, pp. 324–336.

[NTZ16] A. Nikolov, K. Talwar, and L. Zhang. “The Geometry of Differential Privacy: The Small Database and Approximate Cases”. In: *SIAM J. Comput.* 45.2 (2016), pp. 575–616.

[Nil91] A. Nilli. “On the second eigenvalue of a graph”. In: *Discrete Math.* 91.2 (1991), pp. 207–210.

[Nob18] S. U. Noble. *Algorithms of oppression: How search engines reinforce racism*. nyu Press, 2018.

[Pan94] V. Y. Pan. “Simple multivariate polynomial multiplication”. In: *Journal of Symbolic Computation* 18.3 (1994), pp. 183–186.

[Pat98] G. Pataki. “On the rank of extreme matrices in semi-definite programs and the multiplicity of optimal eigenvalues”. In: *Mathematics of operations research* 23.2 (1998), pp. 339–358.

[Pea01] K. Pearson. “On lines and planes of closest fit to systems of points in space”. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11 (1901), pp. 559–572.

[Pet94] E. Petrank. “The hardness of approximation: gap location”. In: *Comput. Complexity* 4.2 (1994), pp. 133–157.

[Puk06] F. Pukelsheim. *Optimal design of experiments*. SIAM, 2006.

[RG96] R. Ravi and M. X. Goemans. “The constrained minimum spanning tree problem”. In: *Scandinavian Workshop on Algorithm Theory*. Springer. 1996, pp. 66–75.

[RSA99] S. Raychaudhuri, J. M. Stuart, and R. B. Altman. “Principal components analysis to summarize microarray experiments: application to sporulation time series”. In: *Biocomputing 2000*. World Scientific, 1999, pp. 455–466.

[RS00] S. T. Roweis and L. K. Saul. “Nonlinear dimensionality reduction by locally linear embedding”. In: *science* 290.5500 (2000), pp. 2323–2326.

[Sam+18] S. Samadi, U. Tantipongpipat, J. H. Morgenstern, M. Singh, and S. Vempala. “The Price of Fair PCA: One Extra Dimension”. In: *Advances in Neural Information Processing Systems*. 2018, pp. 10976–10987.

[Sch98] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

[SX18] M. Singh and W. Xie. “Approximate Positive Correlated Distributions and Approximation Algorithms for D-optimal Design”. In: *To appear in SODA* (2018).

[SYZ08] A. M.-C. So, Y. Ye, and J. Zhang. “A unified theorem on SDP rank reduction”. In: *Mathematics of Operations Research* 33.4 (2008), pp. 910–920.

[SS10] D. Spielman and N. Srivastava. “An Elementary Proof of the Restricted Invertibility Theorem”. In: *Israel Journal of Mathematics* (2010), pp. 1–9.

[SS11] D. A. Spielman and N. Srivastava. “Graph sparsification by effective resistances”. In: *SIAM Journal on Computing* 40.6 (2011), pp. 1913–1926.

[ST17] N. Srivastava and L. Trevisan. “An Alon-Boppana Type Bound for Weighted Graphs and Lowerbounds for Spectral Sparsification”. In: *CoRR* abs/1707.06364 (2017). arXiv: `1707.06364`.

[SV17] D. Straszak and N. K. Vishnoi. “Real stable polynomials and matroids: optimization and counting”. In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*. Ed. by H. Hatami, P. McKenzie, and V. King. ACM, 2017, pp. 370–383.

[SY12] J. Stufken and M. Yang. “Optimal designs for generalized linear models”. In: *Design and Analysis of Experiments* 3 (2012), pp. 137–164.

[SEFM15] M. D. Summa, F. Eisenbrand, Y. Faenza, and C. Moldenhauer. “On largest volume simplices and sub-determinants”. In: *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*. Society for Industrial and Applied Mathematics. 2015, pp. 315–323.

[Swe13] L. Sweeney. “Discrimination in online ad delivery”. In: *Communications of the ACM* 56.5 (2013), pp. 44–54.

[Tan+19] U. Tantipongpipat, S. Samadi, M. Singh, J. H. Morgenstern, and S. Vempala. “Multi-Criteria Dimensionality Reduction with Applications to Fairness”. In: *Advances in Neural Information Processing Systems 32*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett. Curran Associates, Inc., 2019, pp. 15161–15171.

[TDSL00] J. B. Tenenbaum, V. De Silva, and J. C. Langford. “A global geometric framework for nonlinear dimensionality reduction”. In: *science* 290.5500 (2000), pp. 2319–2323.

[UC] UC Irvine Repository. *Adult Data Set*. Machine Learning Repository, Center for Machine Learning and Intelligent Systems. https://archive.ics.uci.edu/ml/datasets/adult. Accessed May 2019.

[Vaz13] V. V. Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

[Ver01] R. Vershynin. “John’s Decompositions: Selecting a Large Part”. In: *Israel Journal of Mathematics* 122.1 (2001), pp. 253–277.

[WUSK18] Y. Wang, E. Ulu, A. Singh, and L. B. Kara. “Detc 2018-85310 Efficient Load Sampling for Worst-case Structural Analysis under Force Location Uncertainty”. In: 2018.

[WYS16] Y. Wang, A. W. Yu, and A. Singh. “On Computationally Tractable Selection of Experiments in Regression Models”. In: *arXiv preprint arXiv:1601.02068* (2016).

[WYS17] Y. Wang, A. W. Yu, and A. Singh. “On computationally tractable selection of experiments in measurement-constrained regression models”. In: *The Journal of Machine Learning Research* 18.1 (2017), pp. 5238–5278.

[WVZX10] G. Wei, A. V. Vasilakos, Y. Zheng, and N. Xiong. “A game-theoretic method of fair resource allocation for cloud computing services”. In: *The journal of supercomputing* 54.2 (2010), pp. 252–269.

[YL09] I.-C. Yeh and C.-h. Lien. “The comparisons of data mining techniques for the predictive accuracy of probability of default of credit card clients”. In: *Expert Systems with Applications* 36.2 (2009), pp. 2473–2480.

[You14] P. Youssef. “Restricted invertibility and the Banach-Mazur distance to the cube”. In: *Mathematika* 60.1 (2014), pp. 201–218.