COLUMN SUBSET SELECTION FOR APPROXIMATING DATA MATRICES

By

Ali Çivril

A Thesis Submitted to the Graduate

Faculty of Rensselaer Polytechnic Institute

in Partial Fulfillment of the

Requirements for the Degree of

DOCTOR OF PHILOSOPHY

Major Subject: COMPUTER SCIENCE

Approved by the
Examining Committee:

_____
Malik Magdon-Ismail, Thesis Adviser

_____
Petros Drineas, Member

_____
Mark Goldberg, Member

_____
John E. Mitchell, Member

Rensselaer Polytechnic Institute
Troy, New York

December 2009
(For Graduation December 2009)

Annem, babam ve kardeşime ithaf olunur.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

ACKNOWLEDGMENT

I am truly grateful to my advisor Malik Magdon-Ismail who played the most important role in my acceptance to graduate school, who has always supported and guided me. He was not only academically sharp and diverse, but also quite lenient and understanding on personal issues. I also learned a lot from him about the academic environment and different research paradigms.

I would like to thank my committee members Petros Drineas, Mark Goldberg and John Mitchell. Special thanks go to Petros Drineas and his student Christos Boutsidis for several encouraging discussions and help on technical issues.

I would like to thank Bülent Yener for being interested in my progress and for his several candid suggestions.

A PhD is impossible without friends. My roommate Çağrı Özçağlar has the greatest credit whose close support for my ill health was invaluable. We also have a common background to a great extent which resulted in unforgettable late-night conversations. My special thanks go to Hilmi Yıldırım and Erol Akmercan for also supporting me during my difficult times despite their busy schedule. Others include Süleyman Vural and Asil Ali Özdoğru. I would like to thank Jon Purnell and Eli Bocek-Rivele for introducing a warm American perspective into my life, and my other lab-mate Mykola Hayvanovich. The non-Western gang deserves a big thanks, too: Asif Javed, Saeed Salem, Vineet Chaoji and Mohammad Hasan. I would especially like to thank Saeed who is my previous roommate. Thanks to Cemal Çağatay Bilgin and Buğra Çaşkurlu who have always been around my office for leisurely talk. I hope the ones who were unintentionally forgotten due to lack of time and space would except my apologies.

Terry Hayden and Chris Coonrad helped me on several administrative issues which significantly reduced my load and stress.

Finally, I want to express my deepest gratitude to my family, my mother Nimet Çivril, my father Nihat Çivril and my sister Hanife Çivril for their unconditional, unceasing support and prayers. This work is dedicated to them.

ABSTRACT

In this thesis, we study the problem of selecting a subset of columns of a matrix so that they capture the important information contained in the matrix. We present complexity results and algorithms. The problem, in a very broad sense, asks for a "good" subset of columns of a given real matrix that provides a performance guarantee in terms of an objective function related to the spectrum of the matrix.

We first present a linear-time spectral graph drawing algorithm as a motivation which is a vast improvement over the standard quadratic-time method Classical Multidimensional Scaling (CMDS). To guarantee a fast implementation of the algorithm, it is desirable to quickly select a subset of columns of a distance matrix associated with the graph. Intuitively, in order to obtain a well conditioned sub-matrix, one has to choose a subset of column vectors -in a geometrical sense- such that they are as "far away" from each other as possible. We consider formalizations of this notion by studying the problem of selecting a subset of columns of size k such that it satisfies some certain orthogonality conditions. We establish the NP-hardness of a few such problems and further show that two of them do not admit PTAS. For the problem of choosing the maximum volume sub-matrix, which we call MAX-VOL, we analyze a greedy algorithm and show that it provides a $2^{-O(k \log k)}$ approximation. Our analysis of the greedy heuristic is tight to within a logarithmic factor in the exponent, which we show by explicitly constructing an instance for which the greedy heuristic is $2^{-\Omega(k)}$ from optimal. Further, we show that no efficient algorithm can appreciably improve upon the greedy algorithm by proving that MAX-VOL is NP-hard to approximate within 2^{-ck} for some constant c. Our proof is via a reduction from the Label-Cover problem.

Our last result is a constructive solution to the low-rank matrix approximation problem which asks for a subset of columns of a matrix that captures "most" of its spectrum. Our main result is a simple greedy deterministic algorithm with guarantees on the performance while choosing a small number of columns. Specifically, our

greedy algorithm chooses c columns from A with $c = \tilde{O}\left(\frac{k^2 \log k}{\epsilon^2}\mu^2(A)\right)$ such that

$$\|A - CC^+A\|_F \leq (1+\epsilon)\|A - A_k\|_F,$$

where C is the matrix composed of the c columns, C^+ is the pseudo-inverse of C (CC^+A is the best reconstruction of A from C), and $\mu(A)$ is a measure of the *coherence* in the normalized columns of A. To the best of our knowledge, this is the first deterministic algorithm with performance guarantees on the number of columns and a $(1+\epsilon)$ approximation ratio in Frobenius norm. Numerical results suggest that the performance of the algorithm might be far better than the theoretical bounds suggest.

CHAPTER 1
Introduction

1.1 Overview

Most data can be represented as an $m \times n$ matrix where the columns are objects and the rows are the features associated with them. Among the important examples of such representation in modern statistical analysis are document-term data, DNA microarray data and user-movie data where the analysts often need to define a feature vector for a specific object. Hence, given a matrix $A \in \mathbb{R}^{m \times n}$, it is of practical importance to obtain the "significant information" contained in A. It becomes especially important to have a compact representation of A when A is large and has low numerical rank, as is typical of modern data. Thus, in a broad sense, we are interested in concise representations of matrices. Besides the tremendous practical impact of linear algebraic algorithms, they also come up in different theoretical forms and paradigms. Specifically, the formalization of "significant information" can be done in several ways and to a great extent, it depends on how a matrix is interpreted.

From a conceptual point of view, rather than interpreting a matrix as a block of numbers, we constrain ourselves to view it as a set of vectors (specifically, column vectors) which are indivisible entities. Thus, the formalization of "significant information" is essentially related to finding a subset of columns of the matrix which satisfies some certain spectral conditions or orthogonality requirements. From a purely combinatorial perspective, treating vectors as elements of a set, one can also view subset selection in matrices as a generalization of the usual subset selection problem where the elements contain little or no information. To give a specific example, the well known Set Cover problem asks for a smallest cardinality subset of a set system which covers a universal set. Likewise, the problem we are interested in essentially asks for a small number of column vectors to "cover" the whole matrix. In what follows, we intuitively state two measures of quality for this problem, which will be the subject matter of this thesis.

Several problems in matrix analysis require to construct a more concise version of a matrix generally performed by a re-ordering of the columns [33], such that the new smaller matrix is as good a representative of the original as possible. One of the criteria that defines the quality of a subset of columns of a matrix is how well-conditioned the sub-matrix that they define is. To motivate the discussion, consider the set of three vectors

$$\left\{ e_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, e_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, u = \begin{pmatrix} \sqrt{1-\epsilon^2} \\ \epsilon \end{pmatrix} \right\},$$

which are clearly dependent, and any two of which are a basis. Thus any pair can serve to reconstruct all vectors. Suppose we choose e_1, u as the basis, then $e_2 = (1/\epsilon)u - (\sqrt{1-\epsilon^2}/\epsilon)e_1$, and we have a numerical instability in this representation as $\epsilon \to 0$. Such problems get more severe as the dimensionality of the space gets large (curse of dimensionality), and it is natural to ask the representatives to be "as far away from each other as possible". From the simplest example, it is clear to see that two vectors orthogonal to each other will capture more information about a superset of columns than the two that have an acute angle between each other. Hence, in its generality, this vaguely stated problem can be stated as finding a subset of columns with the maximum volume possible or equivalently with the maximum determinant. Indeed, in one of the early works studying Rank Revealing QR (RRQR) factorizations [39], while discussing different options on how to choose a good sub-matrix, it was noted that it turns out that "the selection of the sub-matrix with the maximum smallest singular value suggested in [32] can be replaced by the selection of a sub-matrix with maximum determinant".

Although selecting a subset of columns is insightful to obtain a compact representation, this approach in general does not yield an optimal subspace as a solution to our problem with respect to certain matrix norms. The usual spectral approach to this problem is to take the best rank-k ($k < \min\{m, n\}$) approximation revealed by the Singular Value Decomposition (SVD), which minimizes the error with respect to any unitarily invariant norm. Geometrically viewed, SVD reveals the best k dimensional subspace in the sense that the column vectors of A are not so "far

away" from it. The immediate question is, whether it is possible to quickly select a subset of columns so that the subspace that they define is as close as possible to the optimal one, hoping that it will give similar results, i.e. that the column vectors of A will be close to it. There are numerous practical impacts of choosing actual columns of a matrix rather than using a new set of coordinates revealed by the singular values. For example, in statistical data analysis, the singular vector representation might not be suitable to make inferences about the actual underlying data since they are generally combinations of all the columns of the raw information in A: in the microarray data, the combinations of the column vectors have no sensible interpretation [45].

In brief, there are two simple paradigms for attacking the problem we consider which can be stated as:

- choose columns so that they are "far apart" from each other
- choose columns so that they are "close enough" to the optimal subspace that reconstructs the whole matrix

This thesis presents algorithms and complexity theoretic results about the problems related to these two main approaches.

1.2 Our Contributions and Outline

In Chapter 2, we first present preliminaries together with the related work. As a motivation to our theoretical work, we then present a fast spectral graph drawing algorithm in Chapter 3 which is an improvement over the well known method called Multidimensional Scaling (MDS). The algorithm uses a matrix decomposition called CUR decomposition [50] (or CGR by [34]) and chooses a subset of columns by making one pass over the matrix. The method of choosing pivot nodes corresponding to the columns of the associated matrix gives valuable insights about the theoretical justification of the success of the algorithm in practice.

In Chapter 4, inspired from the heuristic approach developed for the graph drawing algorithm, we then turn our attention to formalizing the notions of a "good"

subset of columns. We define four related problems in which we try to find a submatrix $C \in \mathbb{R}^{m\times k}$ of a given matrix $A \in \mathbb{R}^{m\times n}$ such that (i) $\sigma_{max}(C)$ (the largest singular value of C) is minimum, (ii) $\sigma_{min}(C)$ (the smallest singular value of C) is maximum, (iii) $\kappa(C) = \sigma_{max}(C)/\sigma_{min}(C)$ (the condition number of C) is minimum, and (iv) the volume of the parallelepiped defined by the column vectors of C is maximum. We establish the NP-hardness of these problems and further show that (i) and (iv) do not admit PTAS. We then study a natural greedy heuristic for the maximum volume problem and show that it has approximation ratio $2^{-O(k\log k)}$. Our analysis of the greedy heuristic is tight to within a logarithmic factor in the exponent, which we show by explicitly constructing an instance for which the greedy heuristic is $2^{-\Omega(k)}$ from optimal.

Chapter 4 is dedicated to proving an exponential inapproximability result for MAX-VOL, thereby reducing the gap between the hardness of proved in the previous chapter and the upper bound provided by the greedy algorithm. Specifically, we prove that it is not approximable to within 2^{-ck} for some absolute constant $c > 0$ unless $P = NP$. The conclusion is that the greedy algorithm is about all we can hope for when it comes to worst case performance.

Chapter 5 presents a simple greedy *deterministic* algorithm for the problem of low-rank approximation to a matrix, with guarantees on the performance and the number of columns chosen. Our greedy algorithm chooses c columns from A with $c = \tilde{O}\left(\frac{k^2\log k}{\epsilon^2}\mu^2(A)\right)$ such that

$$\|A - CC^+A\|_F \leq (1+\epsilon)\,\|A - A_k\|_F,$$

where C is the matrix composed of the c columns, C^+ is the pseudo-inverse of C, CC^+A is the best reconstruction of A from C, and $\mu(A)$ is a measure related to the *coherence* in the normalized columns of A. To the best of our knowledge, this is the first deterministic algorithm with performance guarantees on the number of columns and a $(1+\epsilon)$ approximation ratio in Frobenius norm. The algorithm is quite simple and intuitive and is obtained by combining a generalization of the well known *sparse approximation problem* from information theory with an *existence* result on the possibility of sparse approximation. We provide empirical results on various spe-

cially constructed matrices comparing our algorithm with the previous deterministic approaches based on QR factorizations and a recently proposed randomized algorithm. The results indicate that in practice, the performance of our algorithm can be significantly better than the bounds suggest. In particular, our algorithm generally outperforms the other algorithms, even the randomized algorithms which have better asymptotic theoretical guarantees (with high probability).

Finally, Chapter 6 discusses the results in a general framework and proposes several open problems in different venues of research we have opened along with possible ways of attack.

Chapter 3 contains mostly work which was published in [14, 15]. The material in Chapter 4 appeared in [13]. The material in Chapter 5 is based on a manuscript to be submitted. Chapter 6 is an extension of the paper which appeared in [12].

CHAPTER 2
Preliminaries and Related Work

This section summarizes the basic facts about linear algebra and matrices together with the definitions related to our main work. The non-standard notation that we adopt for a specific chapter will be given later at the beginning of the chapter. The related work about spectral graph drawing will be given in the next chapter where we introduce our own algorithm.

2.1 Overview of Matrices and Linear Algebra

We deal with matrices and vectors with real entries in Euclidean space. We denote matrices and sets with capital letters. Small letters are generally used for vectors and elements. Specifically, $A \in \mathbb{R}^{m \times n}$ denotes a matrix with m rows and n columns, which can be thought of as n column vectors in m dimensional Euclidean space. The dot product of two m dimensional vectors x and y is given by

$$x \cdot y = \sum_{i=1}^{m} x_i y_i,$$

where x_i denotes the i^{th} element of x. The standard norm of an m dimensional vector x in this space is denoted by

$$\|x\|_2 = x \cdot x = \sqrt{\sum_{i=1}^{m} |x_i|^2}.$$

$A_{(i)}$ denotes the i^{th} row of A for $1 \leq i \leq m$, and $A^{(j)}$, the j^{th} column of A for $1 \leq j \leq n$. A_{ij} is the element at i^{th} row and the j^{th} column. A^T denotes the *transpose* of A, and $diag(a_1, a_2 \ldots, a_n)$ denotes a matrix with entries $a_1, a_2 \ldots, a_n$ on its diagonal and zeros elsewhere, so

$$diag(a_1, a_2, \ldots, a_n) = \begin{pmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_n \end{pmatrix}$$

I denotes the identity matrix with diagonal entries 1, i.e. $I = diag(1, 1, \ldots, 1)$. The spectral norm of a matrix A is given by

$$\|A\|_2 = \max_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2}.$$

The Frobenius norm is given by

$$\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} A_{ij}^2}.$$

$span(A)$ denotes the subspace spanned by the column vectors of A. The range of a matrix $A \in \mathbb{R}^{m \times n}$ is

$$range(A) = \{y \in \mathbb{R}^m | y = Ax \text{ for some } x \in \mathbb{R}^n\} = span(A^{(1)} \ldots A^{(n)}).$$

The rank of A, $rank(A)$, is the dimension of $range(A)$ and is equal to the number of linearly independent columns of A.

A matrix is called *orthonormal* if the pair-wise dot products of its columns are all 0 and the columns have unit norm. An orthonormal transformation does not change the norm of a vector; that is $\|Ux\|_2 = \|x\|_2$. An orthonormal matrix U satisfies $U^T U = I$. A square orthonormal matrix is *orthogonal*. The spectral norm and the Frobenius norm are *unitarily invariant* in the sense that $\|UA\|_2 = \|A\|_2$ and $\|UA\|_F = \|A\|_F$ for any orthogonal matrix U with proper dimensions. Typically, we use C to denote a subset of columns of A, written $C \subset A$, i.e. C is a column sub-matrix of A.

Every $A \in \mathbb{R}^{m \times n}$ of rank r admits a factorization of the form

$$A = QR,$$

where $Q \in \mathbb{R}^{m\times r}$ is an orthogonal matrix, and $R \in \mathbb{R}^{r\times n}$ is upper triangular. That is, the QR factorization reveals an orthonormal basis for the space spanned by the columns of a matrix.

The Singular Value Decomposition (SVD) of a matrix $A \in \mathbb{R}^{m\times n}$ of rank r is denoted by

$$A = U\Sigma V^T$$

where $U \in \mathbb{R}^{m\times r}$ is the orthonormal matrix of left singular vectors, $\Sigma \in \mathbb{R}^{r\times r}$ is the diagonal matrix containing the singular values of A in descending order, i.e. $\Sigma = diag(\sigma_1(A), \ldots, \sigma_r(A))$ where $\sigma_1(A) \geq \sigma_2(A) \ldots \geq \sigma_r(A) > 0$ are the singular values of A. $V \in \mathbb{R}^{r\times n}$ is the orthonormal matrix of right singular vectors. The SVD reveals a lot of information about the structure of a matrix through the singular values. Specifically, we have

$$\|A\|_2 = \sigma_1(A), \qquad \|A\|_F = \sqrt{\sum_{i=1}^{r} \sigma_i(A)^2}.$$

The "best" rank k approximation to A (with respect to a unitarily invariant norm) is $A_k = U_k\Sigma_k V_k^T$ where U_k and V_k are the first k columns of the corresponding matrices in the full SVD of A, and Σ_k is the $k \times k$ diagonal matrix of the first k singular values. To be more precise:

$$\min_{B\in\mathbb{R}^{m\times n}, rank(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}(A).$$

$$\min_{B\in\mathbb{R}^{m\times n}, rank(B)=k} \|A - B\|_F = \|A - A_k\|_F = \sum_{i=k+1}^{r} \sigma_i(A).$$

The pseudo-inverse of A is denoted by $A^+ = V\Sigma^+U^T$, where $\Sigma^+ = diag\left(\frac{1}{\sigma_1}, \ldots \frac{1}{\sigma_r}\right) = \Sigma^{-1}$. Let $C \in \mathbb{R}^{m\times k}$ be a full rank sub-matrix of $A \in \mathbb{R}^{m\times n}$, the best reconstruction of A from C (with respect to a unitarily invariant norm) is CC^+A. That is,

$$\min_{X \in \mathbb{R}^{k \times n}} \|A - CX\|_2 = \|A - CC^+A\|_2.$$

$$\min_{X \in \mathbb{R}^{k \times n}} \|A - CX\|_F = \|A - CC^+A\|_F.$$

2.2 Related Work

In this section, we review the basic related work about selecting a subset of columns of a matrix. It falls into three main categories as related to our investigation.

- *Results on Low Rank Matrix Approximation:* This body of work, mainly pursued in theoretical computer science, aims to construct an approximation to A_k up to an arbitrary precision ϵ by choosing as few columns as possible. The number of columns chosen is a function of k and ϵ.

- *Results on Rank Revealing QR (RRQR) Factorization:* An RRQR factorization reveals the rank of a matrix by re-ordering the columns via pivoting strategies. The algorithms developed for this purpose choose exactly k columns and the quality of approximation is a function of k and n (n is the number of columns of the matrix).

- *Results on Volume:* We review the results revealing the relationship between the volume of a subset of column vectors of a matrix and its low-rank approximations and RRQR factorizations.

2.2.1 Low-Rank Matrix Approximation

With the advent of massive data sets, much work in theoretical computer science has been on finding algorithms for matrix approximation by considering a careful choice of a subset of the columns of the data matrix. The seminal paper by Frieze, Kannan and Vempala [30] gives a randomized algorithm that chooses a subset of columns $C \in \mathbb{R}^{m \times c}$ of A such that $\|A - CC^+A\|_F \leq \|A - A_k\|_F + \epsilon\|A\|_F$, where $c = poly(k, 1/\epsilon, 1/\delta)$, and δ is the failure probability of the algorithm. Subsequent work [25, 26, 53] introduced several improvements on the dependence of c

on $k, 1/\epsilon$ and $1/\delta$ also extending the analysis to the spectral norm. These algorithms sample columns of the matrix with probability proportional to their Euclidean norm squared. The related information is stored in the memory after one pass over the matrix and the approximation is computed in the second pass. Later, [53] showed that the same sampling scheme also yields similar results in the spectral norm.

Recently, the effort has been towards eliminating the additive term in the inequality thereby yielding a relative approximation of the form $\|A - CC^+A\|_F \leq (1+\epsilon)\|A - A_k\|_F$. Along these lines, Deshpande et al. [24] first showed the existence of such approximations introducing a sampling technique related to the volume of the simplex defined by the column subsets of size k, without giving a polynomial time algorithm. Specifically, they showed that there exists k columns with which one can get a $\sqrt{k+1}$ relative error approximation in Frobenius norm, which is tight. Later, Deshpande and Vempala [24] and Drineas et al. [27] provided algorithms with different sampling schemes attaining the $(1+\epsilon)$ approximation where the number of columns is a function of k and ϵ. Other recent approaches for the low-rank approximation problem includes random projections [54], and sampling which exploits geometric properties of high dimensional spaces [55]. All of these algorithms exploit the power of randomization and they introduce a trade-off between the number of columns chosen, the error parameter and the failure probability of the algorithm. The proof techniques presented in these papers break when the random sampling approach is replaced by a deterministic column selection procedure. In contrast, our result in Chapter 6 provides a deterministic algorithm with approximation ratio $(1+\epsilon)$ together with a bound on the number of columns chosen. However, this bound is a function of the input matrix, which might be an evidence that deterministic solutions for this problem are difficult to analyze.

2.2.2 Rank Revealing QR (RRQR) Factorization

When it comes to deterministic approximation, no $(1+\epsilon)$ approximation algorithms are known. The linear algebra community has developed deterministic algorithms in the framework of *rank revealing QR (RRQR) factorizations* [16] which yield some approximation guarantees in spectral norm. Given a matrix $A \in \mathbb{R}^{n \times n}$,

consider the QR factorization of the form

$$A\Pi = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}, \tag{2.1}$$

where $R_{11} \in \mathbb{R}^{k\times k}$ and $\Pi \in \mathbb{R}^{n\times n}$ is a permutation matrix. By the interlacing property of singular values (see [33]), $\sigma_k(R_{11}) \leq \sigma_k(A)$ and $\sigma_1(R_{22}) \geq \sigma_{k+1}(A)$. If the numerical rank of A is k, i.e. $\sigma_k(A) \gg \sigma_{k+1}(A)$, then one would like to find a permutation Π for which $\sigma_k(R_{11})$ is sufficiently large and $\sigma_1(R_{22})$ is sufficiently small. A QR factorization is said to be a rank revealing QR (RRQR) factorization if $\sigma_k(R_{11}) \geq \sigma_k(A)/p(k,n)$ and $\sigma_1(R_{22}) \leq \sigma_{k+1}(A)p(k,n)$, where $p(k,n)$ is a low degree polynomial in k and n.

Much research on finding RRQR factorizations has yielded improved results for $p(k,n)$ [16, 17, 18, 19, 20, 21, 37, 39, 49]. These algorithms make use of the *local maximum volume* concept. Tight bounds for $p(k,n)$ can be used to give deterministic low rank matrix approximation with respect to the spectral norm, via the following simple fact.

Theorem 2.1. *Let Π_k be the matrix of first k columns of Π in (2.1). Then,*

$$\|A - (A\Pi_k)(A\Pi_k)^+ A\|_2 \leq p(k,n) \cdot \|A - A_k\|_2.$$

The best $p(k,n)$ was proposed by Gu and Eisenstat [37]. The authors show that there exists a permutation Π for which $p(k,n) = \sqrt{1+k(n-k)}$. It is not known whether such a permutation can be computed in polynomial time. Instead, algorithms with $p(k,n) = \sqrt{1+f^2k(n-k)}$ were given which run in $O((m+n\log_f n)n^2)$ time for $f > 1$ [37]. Hence, for constant f, the approximation ratio depends on n and the running time is $O(mn^2 + n^3 \log n)$. Note that, these algorithms consider choosing exactly k columns and the results are not directly comparable to ours as they provide bounds on the spectral norm. It is not clear whether these algorithmic results can be extended to give non-trivial bounds in Frobenius norm or to choose more than k columns so as to yield $(1+\epsilon)$ approximation.

Very recently, Boutsidis et al. [9] introduced a hybrid algorithm for the prob-

lem of choosing exactly k columns from a matrix A to approximate A_k, combining the random sampling schemes and the deterministic column pivoting strategies exploited by QR algorithms. Their algorithm provides a performance guarantee of $p(k,n) = O\left(k^{\frac{3}{4}}\log^{\frac{1}{2}}(k)(n-k)^{\frac{1}{4}}\right)$ for spectral norm, and $p(k,n) = O\left(k\sqrt{\log k}\right)$, with high probability.

2.2.3 Volume

There are a few results revealing the relationship between the volume of a subset of columns of a matrix and its approximation. In [23], the authors introduced 'volume sampling' to find low-rank approximation to a matrix where one picks a subset of columns with probability proportional to the volume of the simplex they define. Improving the existence results in [23], [24] also provided an adaptive randomized algorithm which includes repetitively choosing a small number of columns by approximating volume sampling, to find a low-rank approximation. Thus, sampling larger volume columns is good. A natural question is to ask what happens when one uses the columns with largest volume (deterministic), which is our problem MAX-VOL. For the algorithmic problem of obtaining the columns with largest volume, we rely on [24] as the qualitative intuition behind why obtaining the maximum volume sub-matrix should play an important role in matrix reconstruction.

Goreinov and Tyrtyshnikov [35] provided more explicit statements of how volume is related to low-rank approximations in the following theorems:

Theorem 2.2. [35] *Suppose that A is an $m \times n$ block matrix of the form*

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}$$

where A_{11} is nonsingular, $k\times k$, whose volume is at least μ^{-1} times the maximum volume among all $k\times k$ sub-matrices. Then [1] $\|A_{22} - A_{21}A_{11}^{-1}A_{12}\|_\infty \leq \mu(k+1)\sigma_{k+1}(A)$.

This theorem implies that if one has a good approximation to the maximum volume $k \times k$ sub-matrix, then the rows and columns corresponding to this sub-matrix can be used to obtain a good approximation to the entire matrix in the

[1] $\|B\|_\infty$ denotes the maximum modulus of the entries of a matrix B.

∞-norm. If $\sigma_{k+1}(A)$ is small for some small k, then this yields a low-rank approximation to A. Thus, finding maximum volume sub-matrices is important for matrix approximation. [36] also proves a similar result to Theorem 2.2.

Pan [48] unifies the main approaches developed for finding RRQR factorizations by defining the concept of *local maximum volume* and then gives a theorem relating it to the quality of approximation.

Definition 2.3. [48] *Let $A \in \mathbb{R}^{m\times n}$ and C be a sub-matrix of A formed by any k columns of A. $Vol(C)(\neq 0)$ is said to be local μ-maximum volume in A, if $\mu\, Vol(C) \geq Vol(C')$ for any C' that is obtained by replacing one column of C by a column of A which is not in C.*

Theorem 2.4. [48] *For a matrix $A \in \mathbb{R}^{n\times n}$, an integer k $(1 \leq k < n)$ and $\mu \geq 1$, let $\Pi \in \mathbb{R}^{n\times n}$ be a permutation matrix such that the first k columns of $A\Pi$ is a local μ-maximum in A. Then, for the QR factorization*

$$A\Pi = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix},$$

we have $\sigma_{min}(R_{11}) \geq (1/\sqrt{k(n-k)\,\mu^2+1})\sigma_k(A)$ and $\sigma_1(R_{22}) \leq \sqrt{k(n-k)\,\mu^2+1}\,\sigma_{k+1}(A)$.

We note that, MAX-VOL asks for a stronger property of the set of vectors to be chosen, i.e. it asks for a "good" set of vectors in a global sense rather than requiring local optimality. Nevertheless, it is clear that an approximation ratio for MAX-VOL translates to a result in the context of RRQR factorizations. Because, if one could obtain a subset which is μ-maximum (as opposed to local μ-maximum) then the same theorem would hold, since the volume of *any* new set of vectors which is a result of exchanging a column between the current set and the rest of the columns is smaller than the largest possible volume. This new result obtained via the approximation factor we provide for Greedy is an alternative to a mathematically different analysis provided by [37] which proves $p(k,n) = \sqrt{n-k}\,2^k$ [37]. We state the details in the relevant chapter.

CHAPTER 3
Approximating Multi-Dimensional Scaling

The main goal of this chapter is to present some of the motivations of the theoretical work which follows. We present a spectral graph drawing algorithm based on the well known method Multi-Dimensional Scaling (MDS). In this work, it became clear that very compact and accurate representations to the distance matrix can lead to very efficient algorithms for MDS with results comparable to pure MDS. This is what led to the further investigation of compact ways to represent data.

3.1 Introduction and Related Work

Spectral graph drawing formulates graph drawing as a problem of computing the eigenvalues and eigenvectors of certain matrices related to the structure of the graph. We will briefly review popular methods.

High-Dimensional Embedding (HDE) described in [38] by Harel and Koren embeds the graph in a high dimension (typically 50) with respect to carefully chosen pivot nodes. One then projects the coordinates into two dimensions by using a well-known multivariate analysis technique called principal component analysis (PCA), which involves computing the first few largest eigenvalues and eigenvectors of the covariance matrix of the points in the higher dimension.

ACE (Algebraic multigrid Computation of Eigenvectors) [43] minimizes Hall's Energy function $E = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(x_i - x_j)^2$ in each dimension, modulo some non-degeneracy and orthogonality constraints (n is the number of nodes, x_i is the one-dimensional coordinate of the i^{th} node and w_{ij} is the weight of the edge between i and j). This minimization problem can be reduced to obtaining the eigen-decomposition of the Laplacian of the graph. A multi-scaling approach is also used, creating coarser levels of the graph and relating them to the finer levels using an interpolation matrix.

Being closer to what we propose, Landmark MDS (LMDS) [22] is also a fast method using MDS based on sparsification of the distance matrix. It distinguishes a few nodes in the graph as *landmarks* and performs a spectral decomposition on

the matrix of squared distances among those landmarks followed by positioning of the other nodes using a barycentic method. The algorithms presented in [28, 61] employ slightly different strategies for approximating the distance information. These algorithms have been shown to be extensions of the well known Nystrom approximation [51], which is a general approach to find the top eigenspace of a matrix approximately. Recently, a new method which the authors call Pivot MDS (PMDS) [10] was also introduced which is slightly different from LMDS in that it also uses the distance information between landmarks and non-landmarks.

3.2 MDS

The idea of reproducing the pair-wise distances of a metric space (specifically a graph metric) in ${l_2}^2$ has been known for a long time as Multi-Dimensional Scaling (MDS) [7]. This idea was first used for graph drawing[2] purposes by [44]. Here, we give a more general and complete formulation of this technique along the ways introduced in [14]. Different formulations appeared in [42, 58].

Given an n-point metric space (X, D), let $A \in \mathbb{R}^{n \times n}$ be the distance matrix keeping the pair-wise distances between the points in X. We are trying to find an embedding of points in d-dimensional Euclidean space, $x_1, \ldots, x_n$ where $x_i \in \mathbb{R}^d$ for $n \geq i \geq 1$, so as to approximate the pair-wise distances in A, i.e.

$$\|x_i - x_j\|_2 \approx A_{ij}, \qquad \text{for } i, j = 1, 2, \ldots, n. \tag{3.1}$$

Taking squares of both sides, we have

$${x_i}^2 + {x_j}^2 - 2x_i \cdot x_j \approx A_{ij}^2. \tag{3.2}$$

Let L be an $n \times n$ symmetric matrix such that $L_{ij} = A_{ij}^2$, for $i, j = 1, 2, \ldots, n$. Let X, Q and 1_n be defined as follows:

$$X^T = [x_1, \ldots, x_n], Q^T = [\|x_1\|^2, \ldots, \|x_n\|^2], {1_n}^T = [1, \ldots, 1].$$

[2] In graph drawing, (roughly speaking) one tries to assign coordinates to the nodes of the given graph in the Euclidean plane.

Now (3.2) can be written as

$$[Q1_n{}^T]_{ij} + [Q1_n{}^T]_{ji} - 2[XX^T]_{ij} \approx L_{ij}. \tag{3.3}$$

Since $A_{ij} = A_{ji}{}^T$, and $(Q1_n{}^T)^T = 1_n Q^T$, the entire set of equations in matrix form is

$$Q1_n{}^T + 1_n Q^T - 2XX^T \approx L. \tag{3.4}$$

From this equation, since $rank(X) \leq d$, it immediately follows that the rank of L is at most $d + 2$. That is L has low rank (when $d \ll n$). As it stands, (3.4) is hard to solve on account of the dependance of Q on X. We massage it into a more convenient form by using a projection matrix

$$\gamma = I_n - \frac{1}{n} 1_n 1_n{}^T, \tag{3.5}$$

where I_n is the $n \times n$ identity matrix. Multiplying both sides of (3.4) by γ from left and right, we obtain

$$\gamma Q1_n{}^T\gamma + \gamma 1_n Q^T\gamma - 2\gamma XX^T\gamma \approx \gamma L\gamma. \tag{3.6}$$

Since γ is a projection operator, (3.4) becomes

$$(\gamma X)(\gamma X)^T \approx -\frac{1}{2}\gamma L\gamma, \tag{3.7}$$

where we have used the fact that $\gamma = \gamma^T$. We may interpret this equation more easily by setting

$$Y = \gamma X = (X - \frac{1}{n} 1_n 1_n{}^T X). \tag{3.8}$$

Y is an $n \times d$ matrix containing the same vectors of X in a different coordinate system; one in which $mean(Y) = 0$; in this centered coordinate system, (3.4) is considerably simplified. Letting $M = -\frac{1}{2}\gamma L\gamma$, we get

$$YY^T \approx M. \tag{3.9}$$

Note that Y has rank d. If A were a true Euclidean distance matrix, then M would have rank at most d and we could exactly recover Y, solving our problem. Since A may not be a true Euclidean distance matrix, i.e. A may not be isometrically embeddable in $\mathbb{R}^d$, M will generally have rank greater than d. Naturally, we want to approximate M as closely as possible. The metric MDS chooses is the spectral norm, so we wish to find the best rank-d approximation to M with respect to the spectral norm. This is a well-known problem, which is equivalent to finding the largest d eigenvalues of M. Specifically, order the eigenvalues of M such that $|\lambda_1| \geq |\lambda_2| \geq \cdots \geq |\lambda_n|$ and let $u_1, u_2, \ldots, u_n$ be the associated eigenvectors. Then, the spectral decomposition of M yields $M = \sum_{k=1}^{n} \lambda_k u_k {u_k}^T$, and the rank-$d$ approximation of M is $M_d = \sum_{k=1}^{d} \lambda_k u_k {u_k}^T$. It is well known that M_d is the best rank-d approximation to M with respect to the spectral norm. The final centralized coordinates are then given by $Y = [\sqrt{\lambda_1}u_1, \ldots, \sqrt{\lambda_d}u_d]$. This fact can easily be turned into a graph drawing algorithm for $d = 2$. Specifically, there are two stages:

(i) computing all-pairs shortest path lengths to obtain L.

(ii) finding a rank-d approximation of $M = -\frac{1}{2}\gamma L \gamma$ which corresponds to computing the largest d eigenvalues and eigenvectors.

In order to implement (i), we run a BFS for each node. The complexity of this step is $O(|V||E|)$. For (ii), rather than using an exact algorithm which will be $O(|V|^3)$, we use a standard procedure typically referred to as the power iteration to compute the eigenvalues and eigenvectors of M. The power iteration typically produces results as good as the exact algorithm, but much more efficiently. It starts with some random initial vectors and iteratively multiplies them with the relevant matrix modulo orthogonality constraints. The procedure stops when some termination condition is met, for example, when the change in direction of the eigenvector is negligible, i.e., the cosine of the dot product of the previous estimate and the newly computed estimate is above $1 - \epsilon$ for some small ϵ. We impose one additional condition for termination, which ensures that the ratio of the direction change between two consecutive iterations is above some value, $1 + \epsilon$ in our case. The convergence of the power iteration depends on the eigenvalues of the matrix; in practice it takes

some constant number of iterations to compute the eigenvalues to some precision, since convergence is exponentially fast. The matrix multiplications we perform in the power iteration take $O(|V|^2)$ time, which makes the overall complexity of the power iteration $O(d|V|^2)$. Thus, the complexity of our algorithm is $O(|V||E| + d|V|^2)$, which is equal to $O(|V||E|)$ for $d = 2$. The space complexity is $O(|V|^2)$ since we need to store all the pair-wise distances.

Algorithm 1 MDS($G = (V, E)$)

Compute the distance matrix D using an APSP algorithm on G
Define matrix L such that $L_{ij} = D_{ij}^2$.
return Y = PowerIteration($-\frac{1}{2}\gamma L\gamma, \epsilon$) % epsilon is a tolerance

Algorithm 2 PowerIteration(M,ϵ)

$current \leftarrow \epsilon$; $y_1 \leftarrow random/\|random\|$
repeat
  $prev \leftarrow current$
  $u_1 \leftarrow y_1$
  $y_1 \leftarrow Mu_1$
  $\lambda_1 \leftarrow u_1 \cdot y_1$ % compute the eigenvalue
  $y_1 \leftarrow y_1/\|y_1\|$
  $current \leftarrow u_1 \cdot y_1$
until $current \geq 1 - \epsilon$ or $|current/prev| \leq 1 + \epsilon$
$current \leftarrow \epsilon$; $y_2 \leftarrow random/\|random\|$
repeat
  $prev \leftarrow current$
  $u_2 \leftarrow y_2$
  $u_2 \leftarrow u_2 - u_1(u_1 \cdot u_2)$ % orthogonalize against u_1
  $y_2 \leftarrow Mu_2$
  $\lambda_2 \leftarrow u_2 \cdot y_2$
  $y_2 \leftarrow y_2/\|y_2\|$
  $current \leftarrow u_2 \cdot y_2$
until $current \geq 1 - \epsilon$ or $|current/prev| \leq 1 + \epsilon$
return $(\sqrt{\lambda_1}y_1 \;\; \sqrt{\lambda_2}y_2)$

3.2.1 Performance Analysis of MDS

The formal problem that MDS id attempting to solve is a problem in minimum distortion finite metric embedding. MDS approaches this problem using a spectral

decomposition of the matrix of squared distances. We now give some results that explain the intuition behind why and when MDS will work well in practice. The distance matrix D represents a finite metric space. MDS uses a spectral technique to estimate L, where $L_{ij} = D_{ij}^2$. Suppose that the optimal embedding (which we define below) is given by the coordinates $z_1, \ldots, z_n$, which we collect in the matrix Z (analogous to X, Y). Let D_Z and L_Z be the distance matrix and the matrix of squared distances implied by Z. We can then write

$$L = L_Z + \epsilon. \tag{3.10}$$

We refer ϵ as the metric embedding error. Z is optimal in that $\|\epsilon\|_2$ is infimum over all possible Z. L is *embeddable* if $\epsilon = 0$.

Theorem 3.1. *If L is embeddable, then, up to an orthogonal transformation, MDS returns $Z - \frac{1}{n}1_n{1_n}^T Z$.*

Proof. Multiplying both sides of (3.10) by γ from the left and right, we obtain

$$\begin{aligned} \gamma L \gamma &= \gamma L_Z \gamma \\ &= -2(Z - \frac{1}{n}1_n{1_n}^T Z)(Z - \frac{1}{n}1_n{1_n}^T Z)^T \\ &= -2AA^T, \end{aligned} \tag{3.11}$$

where $A = Z - \frac{1}{n}1_n{1_n}^T Z$ is rank-d. Since MDS computes a rank-d approximation A_d of the matrix $-\frac{1}{2}\gamma L \gamma$, and the right hand side of (3.11) is rank-d, it exactly recovers $-\frac{1}{2}\gamma L \gamma$, i.e., $A_d{A_d}^T = -\frac{1}{2}\gamma L \gamma = AA^T$. Since the singular values of a matrix A are the nonnegative square roots of the eigenvalues of AA^T and the left singular vectors of that matrix are the eigenvectors of AA^T, we can write the singular value decompositions of A_d and A as $A_d = U\Sigma{V_1}^T$ and $A = U\Sigma{V_2}^T$. Multiplying the equation for A_d from left by V_1, we have $A_d V_1 = U\Sigma$ since V_1 is orthogonal. Substituting this expression in the equation for A, we obtain $A = A_d V_1 {V_2}^T$. Note that $V = V_1{V_2}^T$ is also orthogonal since $VV^T = I_d$ and therefore A_d differs from A by at most an orthogonal transformation. □

Since the distance matrix is invariant to orthogonal transformations, we obtain

the following corollary, which is the basic intuition behind MDS:

Corollary 3.2. *When the distance matrix is embeddable, the coordinates recovered by MDS exactly reproduce the distance matrix.*

When the distance matrix is not exactly embeddable, but the embedding error ϵ is small, MDS should approximately reproduce the distance matrix. Suppose that $\epsilon \neq 0$, and let

$$\begin{aligned} M &= -\frac{1}{2}\gamma L_Z \gamma \\ &= -2(Z - \frac{1}{n}1_n 1_n{}^T Z)(Z - \frac{1}{n}1_n 1_n{}^T Z)^T. \end{aligned} \tag{3.12}$$

Let $\epsilon_1 = -\frac{1}{2}\gamma\epsilon\gamma$. Then, using (3.10), $-\frac{1}{2}\gamma L\gamma = M + \epsilon_1$. The next theorem states that the optimal rank-d approximation for $-\frac{1}{2}\gamma L\gamma$ is a also a good approximation to M (up to the embedding error ϵ).

Theorem 3.3. *Let M_d be the optimal rank-d approximation to $-\frac{1}{2}\gamma L\gamma = M + \epsilon_1$. Then, $\|M_d - M\|_2 \leq \|\epsilon\|_2$.*

Proof. By the triangle inequality, we have

$$\begin{aligned} \|M_d - M\|_2 &= \|M_d - M - \epsilon_1 + \epsilon_1\|_2 \\ &\leq \|M_d - M - \epsilon_1\|_2 + \|\epsilon_1\|_2. \end{aligned} \tag{3.13}$$

Since M_d is the best rank-d approximation to $M + \epsilon_1$ and M is itself rank-d,

$$\begin{aligned} \|M_d - M - \epsilon_1\|_2 &\leq \|M - M - \epsilon_1\|_2 \\ &= \|\epsilon_1\|_2. \end{aligned} \tag{3.14}$$

Thus, $\|M_d - M\|_2 \leq 2\|\epsilon_1\|_2$. To conclude, note that since γ is a projection matrix, $\|\gamma\|_2 \leq 1$, so by sub-multiplicativity, $\|\epsilon_1\|_2 = \| - \frac{1}{2}\gamma\epsilon\gamma\|_2 \leq \frac{1}{2}\|\gamma\|_2^2\|\epsilon\|_2 \leq \frac{1}{2}\|\epsilon\|_2$. □

Thus, if MDS returns the coordinates Y, then $\|\gamma YY^T\gamma - \gamma ZZ^T\gamma\|_2$ is small provided that L is nearly embeddable (which depends on the perturbation ϵ). Unfortunately this alone does not guarantee that $\gamma Y \approx \gamma ZO$ for some orthogonal $d \times d$ matrix O. In general, the eigenvectors are not stable to perturbations. The next theorem relates the change in the eigenvectors to the spectrum.

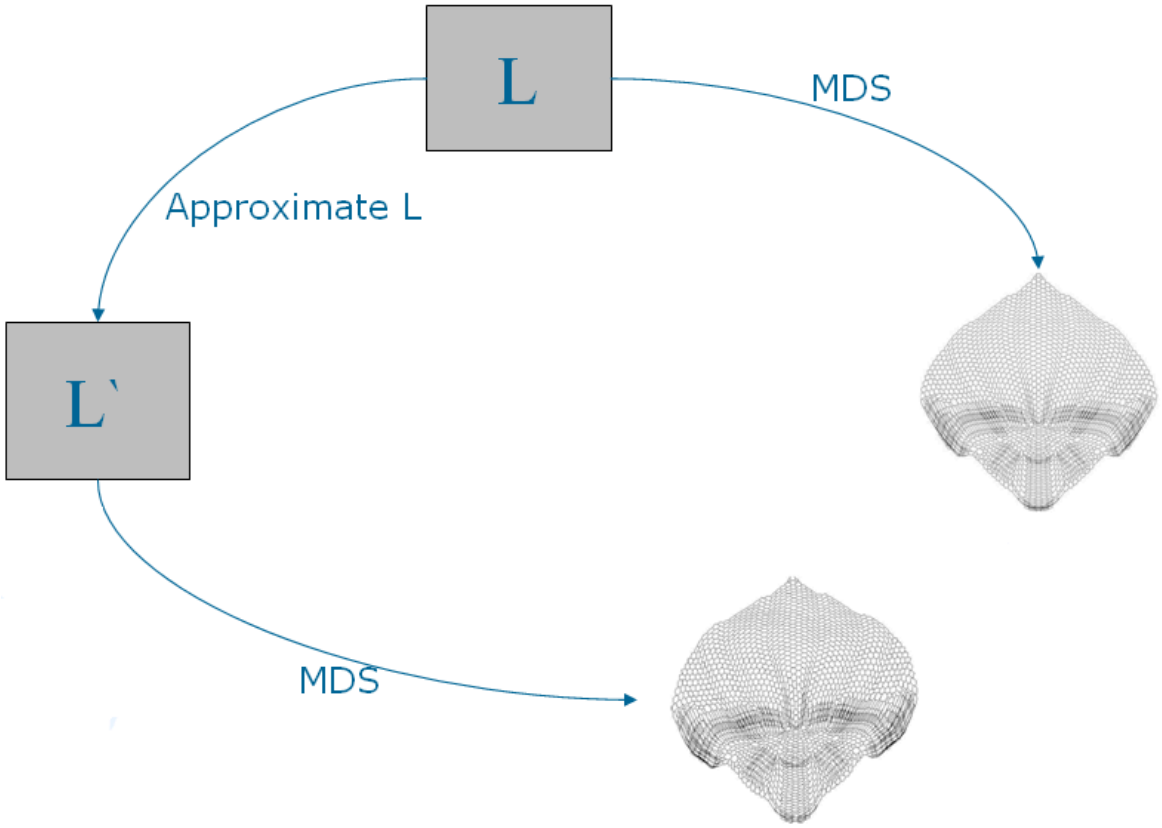
Theorem 3.4 (Corollary 7.2.6 in [33]). *Let $\hat{e}_1$ and $\hat{e}_2$ be the top two eigenvectors of $M+\epsilon_1$, and e_1 and e_2 be the top two eigenvectors of M. Then there exists functions f_1 and f_2 of M and ϵ_1 such that $\|e_1 - \hat{e}_1\|_2 \leq f_1(M, \epsilon_1)$ and $\|e_2 - \hat{e}_2\|_2 \leq f_1(M, \epsilon_1)$. Furthermore, these functions depend on how well separated the eigenvalues of M are, the size of the error ϵ_1, and how well aligned the error is to M.*

The functions f_1, f_2 in Theorem 3.4 have the expected dependence on M, ϵ_1. In particular, the better separated the spectrum of M is and the smaller ϵ_1 is, the smaller the values of f_1, f_2 will be. An additional concern is how well aligned ϵ_1 is to M. The precise formulation of this can be found in [33], but in a nutshell, the perturbations in the small singular value subspaces of M should be small (i.e. the relative perturbations should be small). According to Theorem 3.4, when the eigenvalues of M are well separated and ϵ is small, then not only will M_d be close to M, but so will the top two eigenvectors match closely. Thus, MDS will recover a close approximation to ZO for some orthogonal matrix O, i.e., the distance matrix is nearly recovered. Note that the separation of the eigenvalues is a necessary condition for the power iteration to converge quickly. Thus, when the power iteration fails to converge quickly, it is already a sign that MDS may produce undesirable results. However, it also means that any rank-2 decomposition, including the optimal embedding must have a high relative embedding error. To see this, note that $\|\epsilon_1\|_F^2 \geq \sum_{i=3}^{n} \sigma_j^2(L)$, and since $\sigma_3^2(L) \approx \sigma_{1,2}^2(L)$ (because the singular values are not well separated), the embedding error will be a constant fraction of $\|L\|_F$.

3.3 Approximation of the Distance Matrix

Our goal is to develop an efficient algorithm for MDS while preserving its accuracy. There are some properties of MDS which make it an interesting problem for our investigation:

Figure 3.1: The main approach for approximating MDS



1. It embeds using the most important spectral information.

2. It uses the whole matrix L to get this information.

A convenient low-rank representation for L which we will use was presented in [50]. We will approximate L by a product of three smaller matrices. Here, we briefly review the approach, following the notation used in [15]. Our general approach is depicted in Figure 3.1.

Let $i_1, i_2, \ldots, i_c$ be a set of distinct indices where c is a predefined positive integer smaller than n and $1 \leq i_k \leq n$ for $k = 1, \ldots, c$. Let $C = [L^{(i_1)}, L^{(i_2)}, \ldots, L^{(i_c)}]$, where $L^{(i)}$ denotes the i^{th} column of L ($L_{(i)}$ also denotes the i^{th} row of L). If C is chosen carefully, any column $L^{(i)}$ can approximately be written as a linear combination of the columns of C, i.e.

$$L^{(i)} \approx C\alpha^{(i)} \qquad \text{for } \; i = 1, 2, \ldots, n, \tag{3.15}$$

where $\alpha^{(i)}$ is a $c \times 1$ vector. Denoting $\alpha = [\alpha^{(1)}, \alpha^{(2)}, \ldots, \alpha^{(n)}]$, we have

$$L \approx C\alpha. \tag{3.16}$$

Let Φ be the $c \times c$ matrix such that $\Phi_{jk} = L_{i_j\, i_k}$ for $j, k = 1, \ldots c$. Note that since we also have $C^T = [L_{(i_1)}, L_{(i_2)}, \ldots, L_{(i_c)}]$, Φ can be interpreted as the intersection of C and C^T on the matrix L. Now, since the columns of L can approximately be expressed as a linear combination of the columns of C, the columns of C^T can also be expressed as a linear combination of the columns of Φ. This gives

$$C^T \approx \Phi\alpha, \tag{3.17}$$

where α is the same matrix as we defined above. If Φ has full rank, (3.17) yields $\alpha = \Phi^{-1}C^T$. Combining this with (3.16), we have $L = C\Phi^{-1}C^T$. More generally, we do not assume that Φ has full rank, so we have

$$L \approx C\Phi^{+}C^T. \tag{3.18}$$

The last expression indicates that we can approximate the distance matrix L by the multiplication of three smaller matrices, which all have at most linear size in terms of n. Note that C is $n \times c$ and Φ is $c \times c$. Hence, the matrix-vector multiplications in the power iteration procedure take linear time.

The main question is, how to choose the sub-matrix C. The following strategy was heuristically proposed in [15]: The first point is chosen uniformly at random. Then, at each step, we choose the furthest point to the set of points that have already been chosen until c nodes are chosen. And then, C is formed with the corresponding columns. Note that, this algorithm is also known to be a 2-approximation algorithm for the k-center problem [60]. We begin presenting a practical implementation of our algorithm for $d = 2$, by describing this procedure below. The number of pivot nodes, c can be treated as a parameter to the algorithm; but we have experienced that setting $c = 50$ is enough for getting good results on practically all graphs. The sampling step, overall requires $O(c|E|)$ time as we initiate a BFS from c nodes in the graph.

Algorithm 3 ComputeCandPhi(G, method, c)

$i_1 \leftarrow unifrnd(1, |V|)$ % choose uniformly at random
$C^1 \leftarrow dist(i_1, V)$ % BFS
for $k = 2$ to c **do**
  $i_k \leftarrow \max_{1 \le j \le n} \min_{1 \le l \le k} \{C_{jl}\}$ %choose the furthest node
  $C^k \leftarrow dist(i_k, V)$ % BFS
end for
Compute Φ % $\Phi_{kj} = C_{i_k j}$
return (C, Φ)

We find the pseudo-inverse Φ^+ by first computing the singular value decomposition $\Phi = U\Sigma V^T$, which can be performed in $O(c^3)$ time using standard procedures (See for example [33]). The pseudo-inverse can then be computed by the expression $\Phi^+ = V\Sigma^+ U^T$. Here, Σ^+ is the diagonal matrix keeping the reciprocals of the non-zero singular values, which are stored in Σ. Unfortunately, in order to get numerically stable results, it is not enough to compute the reciprocals of the singular values, since the small singular values which are close to zero should actually be ignored, as they may be the result of numerical imprecision and will result in huge instability in Σ^+. To prevent such instability, we use a *regularization* method which uses the expression

$$\frac{\sigma_i}{{\sigma_i}^2 + \alpha/{\sigma_i}^2} \tag{3.19}$$

for the reciprocals in Σ^+, where σ_i is the i^{th} diagonal entry in Σ. The parameter α is the regularization parameter, which must be chosen judiciously in order not to distort the reciprocals of the large singular values too much. On the other hand, it should result in values close to zero for the small singular values. Our experiments revealed that $\alpha = {\sigma_1}^3$ is good enough for practical purposes where σ_1 is the largest singular value. However, we keep it as a parameter of the procedure, which is presented below.

Having computed the pseudo-inverse of Φ, we compute $\hat{L} = C\Phi^+ C^T$ from which we obtain $\hat{M} = -\frac{1}{2}\gamma\hat{L}\gamma$. Then, analogous to MDS, we obtain the coordinates of the points in the embedding using the spectral decomposition of $\hat{M}$, which approximates M. This requires computing the top 2 eigenvalues and eigenvectors, for

Algorithm 4 Regularize(Σ, α)

for $i = 1$ to c **do**
  ${\Sigma_{ii}}^{+} \leftarrow \frac{\Sigma_{ii}}{\Sigma_{ii}^2} + \alpha/\Sigma_{ii}^2$
end for
return Σ^{+}

which we use the power iteration similar to the one used for MDS. In this procedure, starting from the right, the matrix-vector multiplications can be performed using $O(c|V|)$ scalar additions and multiplications. The total number of iterations until a predefined convergence condition holds, depends on the matrix processed. But, since the convergence is exponential, in practice, a constant number of iterations is enough. Overall, the running time of the power iteration step of the algorithm is $O(c|V|)$, which overall means that our algorithm runs in linear time.

Intuitively, the method that chooses the sub-matrix C tries to make the dot product between two column vectors corresponding to the nodes as small as possible while choosing the most distant node from the already chosen. Because, the nodes closer to the already chosen node are far away from the one to be chosen. Hence, small entries in one column vectors is multiplied by large entries in the other one. In contrast, if two closer nodes are chosen, large entries are multiplied by other large ones. In Figures 3.2 and 3.3, we provide a comparison between MDS and its approximation for several graphs. As it is clearly seen, the information contained in the sub-matrix C of the distance matrix provides a good approximation to the space spanned by the first two eigenvectors. The striking similarity of the pictures of a finite element mesh of a cow, provided by MDS and Approximate MDS also indicates that our method is well suited to accelerate drawings of mesh-like real objects (see Figure 3.4). Whereas, other fast spectral methods have problems visualizing this graph, which is depicted in Figure 3.5. Table 3.1 gives the running time comparison of the two methods on a 2 GB Pentium 4HT 3.0 Ghz machine. The experiments reveal that our approximation is a vast improvement over MDS in terms of running time. Overall, this simple heuristic is empirically successful at approximating MDS by choosing columns that(roughly speaking)

- are far away from each other

- capture the relevant spectral information of the matrix

Inspired from this two simple observations, we investigate the problem of choosing subsets of columns of a matrix so that they satisfy certain related properties to the goals above. So the problem we would like to formulate based on our success with MDS is:

- Can we recover the important subspace of L efficiently so that its properties with respect to MDS is unchanged?
- Can we do this without knowing L (i.e. with an online manner)?

As we will see Problem (i) is already hard enough to analyze theoretically. A theoretical result on Problem (ii) would be icing on the cake, since in any practical problem, it may be too costly to sample all the entries in L. Our heuristic algorithm solves both (i) and (ii) which leaves hope that at least for some classes of distance matrices, we can accomplish (i) and (ii).

Table 3.1: Running time comparison between MDS and Approximate MDS on several graphs (Most of these graphs can be downloaded from [1], [2] and [3]). Missing entries are graphs where it was too costly to compute the entire distance matrix.

Graph	|V|	|E|	MDS	A-MDS(c=50)
3elt	4720	13722	8.47	0.04
sierpinski08	9843	19683	24.72	0.07
Grid 100x100	10000	19800	29.73	0.06
crack	10240	30380	45.00	0.10
4elt2	11143	32818	48.77	0.14
4elt	15606	45878	133.33	0.25
sphere	16386	49152	136.69	0.27
finan512	74752	261120	-	1.43
ocean	143437	409593	-	3.56
144	144649	1074393	-	6.03
wave	156317	1059331	-	4.78
auto	448695	3314611	-	21.67
Florida	1048506	1330551	-	23.45
California	1613325	1989149	-	36.13
Texas	2073870	2584159	-	45.89

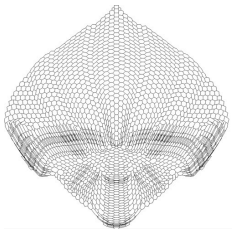
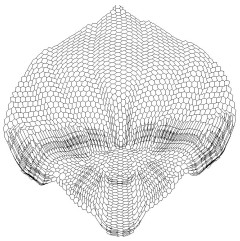
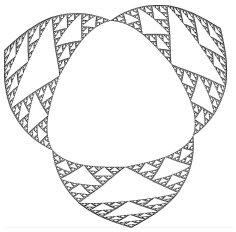
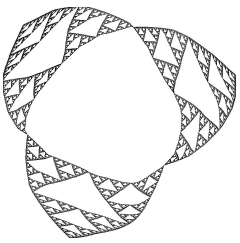
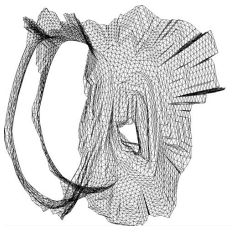
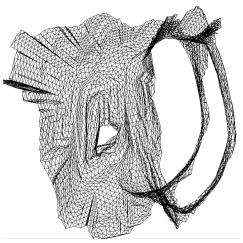
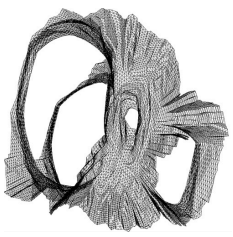
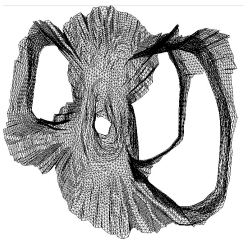
	MDS	**A-MDS (c=25)**
4970 $\|V\| = 4970$ $\|E\| = 7400$		
	running time = 5.04 sec.	running time = 0.01 sec.
sierpinski08 $\|V\| = 9843$ $\|E\| = 19683$		
	running time = 24.72 sec.	running time = 0.03 sec.
3elt $\|V\| = 4720$ $\|E\| = 13722$		
	running time = 8.47 sec.	running time = 0.015 sec.
4elt $\|V\| = 15606$ $\|E\| = 45878$		
	running time = 133.33 sec.	running time = 0.13 sec.

Figure 3.2: Comparison of Layouts Computed by MDS and Approximate-MDS I

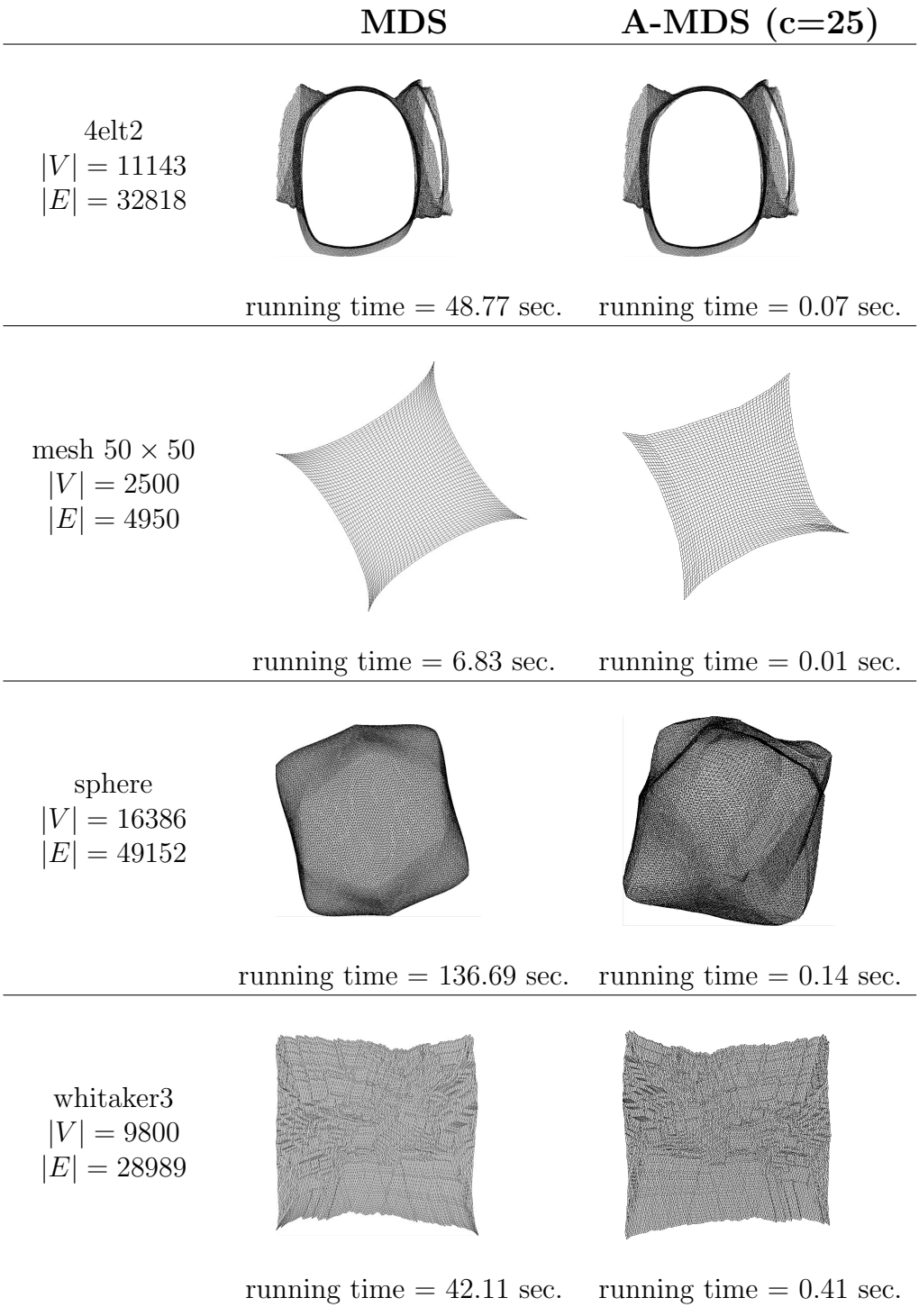


Figure 3.3: Comparison of Layouts Computed by MDS and Approximate-MDS II

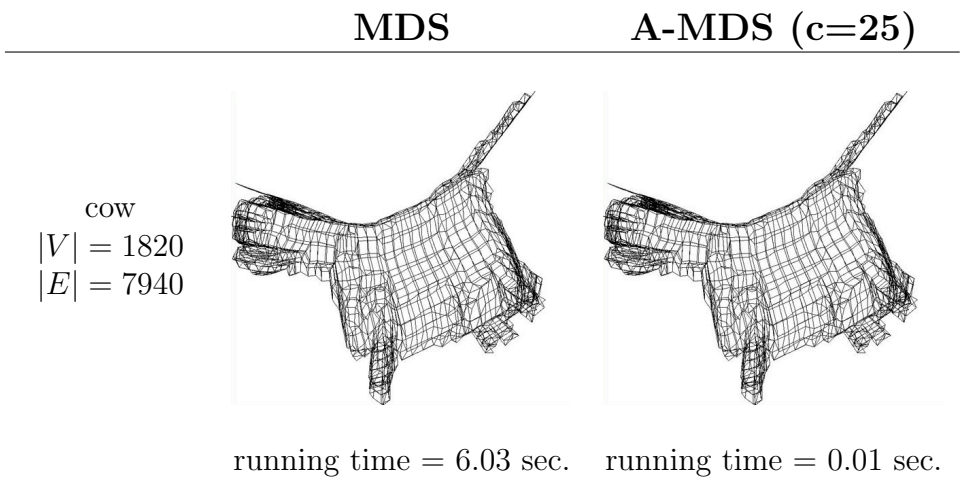


Figure 3.4: Comparison of MDS and Approximate-MDS on a finite element mesh of a cow with $|V| = 1820, |E| = 7940$.

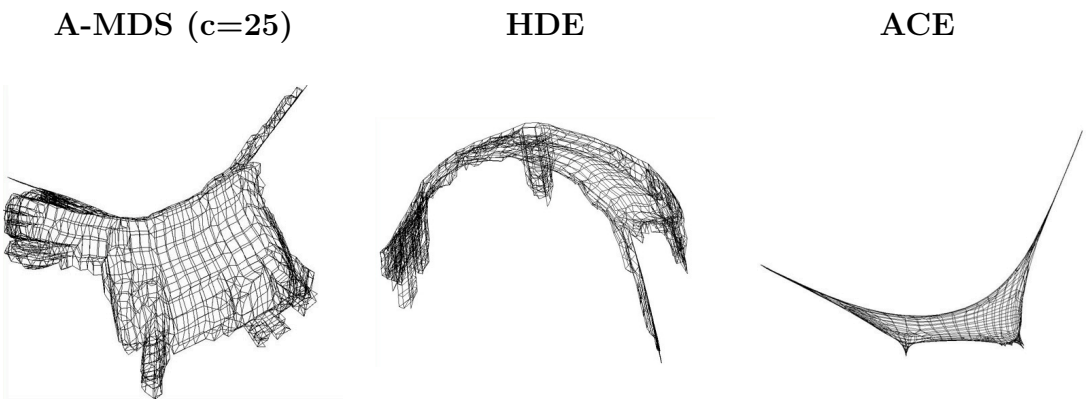


Figure 3.5: Comparison of A-MDS with other spectral methods (HDE and ACE) on the finite element mesh of a cow with $|V| = 1820, |E| = 7940$.

CHAPTER 4
MAX-VOL and Related Problems

4.1 Introduction

As a reminder to the example presented in the introduction, consider the set of three vectors

$$\left\{ e_1 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, e_2 = \begin{pmatrix} 0 \\ 1 \end{pmatrix}, u = \begin{pmatrix} \sqrt{1-\epsilon^2} \\ \epsilon \end{pmatrix} \right\},$$

which are clearly dependent, and any two of which are a basis. Thus any pair can serve to reconstruct all vectors. Suppose we choose e_1, u as the basis, then $e_2 = (1/\epsilon)u - (\sqrt{1-\epsilon^2}/\epsilon)e_1$, and we have a numerical instability in this representation as $\epsilon \to 0$. Such problems get more severe as the dimensionality of the space gets large (curse of dimensionality), and it is natural to ask the representatives to be "as far away from each other as possible". A natural formalization of this problem is to find the representatives which span the largest volume, since the volume is a quantification of how far the vectors are from each other. Another would be to choose them so that the matrix that they form is well conditioned, i.e. its condition number is small which intuitively means that the matrix is as close as possible to an orthogonal one and its smallest singular value is large with respect to the largest singular value. Thus, given a set of n vectors in $\mathbb{R}^m$ represented as a matrix $A \in \mathbb{R}^{m \times n}$ and a positive integer k, we discuss four distinct problems in which we ask for a subset $C \in \mathbb{R}^{m \times k}$ satisfying some spectral optimality condition:

1. MinMaxSingularValue: $\sigma_1(C)$ is minimum;
2. MaxMinSingularValue: $\sigma_k(C)$ is maximum;
3. MinSingularSubset: $\kappa(C) = \sigma_1(C)/\sigma_k(C)$ is minimum;
4. MAX-VOL: $Vol(C) = \prod_{i=1}^{k} \sigma_i(C)$, the volume of the parallelepiped defined by the column vectors of C is maximum.

4.1.1 Contributions of This Chapter

First, we establish the NP-hardness of these four problems. We then obtain the following inapproximability results:

1. MinMaxSingularValue is inapproximable to within $2/\sqrt{3} - \epsilon$;
2. MAX-VOL is inapproximable to within $2\sqrt{2}/3 + \epsilon$.

Here $\epsilon > 0$ is an arbitrarily small constant.

Next, we consider a simple (deterministic) greedy algorithm for the last problem and show that it achieves a $1/k!$ approximation to the optimal volume when selecting k columns. We also construct an explicit example for which the greedy algorithm gives no better than a $1/2^{k-1}$ approximation ratio, thus proving that our analysis of the greedy algorithm is almost tight (to within a logarithmic factor in the exponent). An important property of the approximation ratio for the greedy algorithm is that it is independent of n, and depends only on the number of columns one wishes to select.

We then consider the related problem of choosing the maximum number of vectors with a given volume, in the case when all columns in A have unit norm. If the optimal algorithm loses a constant factor with every additional vector selected (which is a reasonable situation), then the optimal volume will be $2^{-\Omega(k)}$. When the optimal volume for k vectors is $2^{-\Omega(k)}$ as motivated above, we prove that the greedy algorithm chooses $\Omega(k/\log k)$ columns having at least that much volume. Thus, the greedy algorithm is within a $\log k$-factor of the maximum number of vectors which can be selected given a target volume.

The $1/k!$ approximation ratio of the greedy algorithm is a general result about selecting sequences of vectors which are near-orthogonal sequentially. Each step in the sequential algorithm turns out to be successively worse which is what leads to the $1/k!$ approximation. It turns out that even this approximation ratio has had some unexpected consequences: indeed, [56] shows that vectors constructed by Greedy can be used to obtain approximate Fekete points. Further, a $1/k!$ approximation ratio is only $O(\log k)$ (in the exponent) worse than the optimal as we will show in the next chapter.

4.1.2 Preliminaries and Notation

Let $A = \{v_1, v_2, \ldots, v_n\}$ be given in column notation. The volume of A, $Vol(A)$ can be recursively defined as follows: if A contains one column, i.e. $A = \{v_1\}$, then $Vol(A) = \|v\|$, where $\|\cdot\|$ is the Euclidean norm. If A has more than one column, $Vol(A) = \|v - \pi_{(A-\{v\})}(v)\| \cdot Vol(A - \{v\})$ for any $v \in A$, where $\pi_A(v)$ is the projection of v onto the space spanned by the column vectors of A. It is well known that $\pi_{(A-\{v\})}(v) = A_v A_v^+ v$, where A_v is the matrix whose columns are the vectors in $A - \{v\}$, and A_v^+ is the pseudo-inverse of A_v (see for example [33]). Using this recursive expression, we have

$$Vol(S) = Vol(A) = \|v_1\| \cdot \prod_{i=1}^{n-1} \|v_{i+1} - A_i A_i^+ v_{i+1}\|$$

where $A_i = [v_1 \cdots v_i]$ for $\leq i \leq n-1$.

4.2 Hardness of Subset Selection Problems

We define four decision problems:

Problem: **Min-MaxSingularValue**

Instance: A matrix $A \in \mathbb{R}^{m \times n}$ of rank at least k, and $M \in \mathbb{R}$.

Question: Does there exist a sub-matrix $C \in \mathbb{R}^{m \times k}$ of A such that $\sigma_1(C) \leq M$.

Problem: **Max-MinSingularValue**

Instance: A matrix $A \in \mathbb{R}^{m \times n}$ of rank at least k, and $M \in \mathbb{R}$.

Question: Does there exist a sub-matrix $C \in \mathbb{R}^{m \times k}$ of A such that $\sigma_k(C) \geq M$.

Problem: **Min-SingularSubset**

Instance: A matrix $A \in \mathbb{R}^{m \times n}$ of rank at least k, and $M \in \mathbb{R}$.

Question: Does there exist a sub-matrix $C \in \mathbb{R}^{m \times k}$ of A such that $\sigma_1(C)/\sigma_k(C) \leq M$.

Problem: **MAX-VOL**

Instance: A matrix $A \in \mathbb{R}^{m \times n}$ with normalized columns and of rank at least k, and $M \in [0, 1]$.

Question: Does there exist a sub-matrix $C \in \mathbb{R}^{m \times k}$ of A such that $Vol(C) \geq M$?

Theorem 4.1. *Max-MinSingularValue, Min-MaxSingularValue, Min-SingularSubset and MAX-VOL are NP-Hard.*

Proof. We give a reduction from 'exact cover by 3-sets', which is known to be NP-complete (See for example [31, 41]). This reduction will provide the NP-hardness result for all the problems.

Problem: **Exact cover by 3-sets (X3C)**

Instance: A set Q and a collection C of 3-element subsets of Q.

Question: Does there exist an exact cover for Q, i.e. a sub-collection $C' \subseteq C$ such that every element in Q appears exactly once in C'?

We use the following reduction from X3C: let $Q = \{q_1, q_2, \ldots q_m\}$ and $C = \{c_1, c_2, \ldots c_n\}$ be given as an instance of X3C. We construct the matrix $A \in \mathbb{R}^{m \times n}$, in which each column $A^{(j)}$ corresponds to the 3-element set c_j. The non-zero entries in $A^{(j)}$ correspond to the elements in c_j. Specifically, set

$$A_{ij} = \begin{cases} 1/\sqrt{3} & \text{if } q_i \in c_j \\ 0 & \text{otherwise} \end{cases}$$

(Note that every $A^{(j)}$ has exactly 3 non-zero entries and has unit norm.) For the reduced instances, we set $k = m/3$ and $M = 1$.

It is clear that the reduction is polynomial time. All that remains is to show that the instance of X3C is true if and only if the corresponding instances of the four decision problems are true.

Suppose the instance of X3C is true. Then, there is a collection $C' = \{c_{i_1}, c_{i_2}, \ldots, c_{i_{m/3}}\}$ of cardinality $m/3$, which exactly covers Q. (Note that, m should be a multiple of

3, otherwise no solution exists.) Consider the sub-matrix C of A corresponding to the 3-element sets in C'. Since the cover is exact, $c_{i_j} \cap c_{i_k} = \emptyset \; \forall j, k \in \{1, \ldots, m/3\}$ where $j \neq k$, which means that $A^{(i_j)} \cdot A^{(i_k)} = 0$. Hence, C is orthonormal and all its singular values are 1, which makes the instances of all four problems we consider true.

Conversely, suppose the instance of Min-MaxSingularValue is true, i.e. there exists C such that $\sigma_1(C) \leq 1$. We have $\sigma_1(C) = \|C\|_2 \geq \|C\|_F/\sqrt{k} = 1$, which gives $\sigma_1(C) = 1$. On the other hand, $\sum_{i=1}^{k} \sigma_i(C)^2 = \|C\|_F^2 = k$. Thus, all the singular values of C are equal to 1, i.e. C is an orthogonal matrix. Now, suppose the instance of Max-MinSingularValue is true, namely there exists C such that $\sigma_k(C) \geq 1$. Then, the volume defined by the vectors in C, $Vol(C) = \prod_{i=1}^{k} \sigma_i(C) \geq 1$. Since the vectors are all normalized, we also have $Vol(C) \leq 1$, which gives $\prod_{i=1}^{k} \sigma_i(C) = 1$. Thus, all the singular values of C are equal to 1, which means that C is an orthogonal matrix. If the instance of Min-SingularSubset is true, i.e. there exists C such that $\sigma_1(C)/\sigma_k(C) \leq 1$, we immediately have that C is an orthogonal matrix. Finally, suppose that the instance of MAX-VOL is true, this means that the columns are pair-wise orthogonal and we have the desired result.

Thus, if any of the reduced instances are true, then there is a C in A whose columns are pairwise orthogonal. We will now show that if such a C exists, then the instance of X3C is true. Let u, v be two columns in C; we have $u \cdot v = 0$. Since the entries in C are all non-negative, $u_i \cdot v_i = 0 \; \forall i \in [1, m]$, i.e. u and v correspond to 3-element sets which are disjoint. Hence, the columns in C correspond to a sub-collection C' of 3-element sets, which are pair-wise disjoint. Therefore, every element of Q appears at most once in C'. C' contains m elements corresponding to the m non-zero entries in C. It follows that every element of Q appears exactly once in C', concluding the proof. □

Our reduction in the NP-hardness proofs yields gaps, which also provides hardness of approximation results for the optimization versions of the problems.

Theorem 4.2. *Min-MaxSingularValue(k) is NP-hard to approximate within* $2/\sqrt{3} - \epsilon$.

Proof. We will provide a lower bound for $\sigma_1(C)$ for the reduced instance of X3C when it is false, which will establish the hardness result. Assume the X3C instance is not true. Then any collection of size $m/3$ has at least two sets which have non-empty intersection. Let s_i and s_j be two sets such that $|s_i \cap s_j| = 1$. And, let v_i, v_j be the corresponding vectors in the Min-MaxSingularValue instance. Then, we have $v_1 \cdot v_2 = 1/3$. Hence, v_1 and v_2 correspond to the following matrix V up to rotation:

$$V = \begin{pmatrix} 1 & \frac{1}{3} \\ 0 & \frac{2\sqrt{2}}{3} \end{pmatrix}$$

Note that, if $|s_i \cap s_j| > 1$, then the largest singular value of the corresponding matrix will be greater than that of V as $v_1 \cdot v_2$ will have a greater value. Also, it is a well known fact that the largest eigenvalue of any symmetric matrix A is greater than that of any principal sub-matrix of A. Thus, if we consider a matrix W of more than two vectors which also contain v_i and v_j, its largest singular value (which is the square root of the largest eigenvalue of W^TW) will be greater than that of V. Hence, in order to find a lower bound for $\sigma_1(C)$, it suffices to analyze V. This amounts to finding the square roots of the eigenvalues of V^TV. Hence, we are seeking λ such that

$$\det(V^TV - \lambda I) = \begin{vmatrix} \frac{10}{9} - \lambda & \frac{2\sqrt{2}}{9} \\ \frac{2\sqrt{2}}{9} & \frac{8}{9} - \lambda \end{vmatrix} = 0. \tag{4.1}$$

$\lambda = 4/3$ and $\lambda = 2/3$ satisfy (4.1). Hence, $\sigma_1(C) \geq 2/\sqrt{3}$, which concludes the proof. □

Theorem 4.3. *MAX-VOL(k) is NP-Hard to approximate within* $2\sqrt{2}/3 + \epsilon$.

Proof. Assume the X3C instance is not true. Then, we have at least one overlapping element between two sets. Any collection of size $m/3$ will have two sets v_1, v_2 which have non-zero intersection. The corresponding columns in A' have $d(v_1, v_2) = \|v_1 - (v_1 \cdot v_2)v_2\| = \|v_1 - (1/3)v_2\| \leq 2\sqrt{2}/3$, where $d(v_1, v_2)$ is the orthogonal part of v_1 with respect to v_2. Since $Vol(A') \leq d(v_1, v_2)$, we have $Vol(A') \leq 2\sqrt{2}/3$. A polynomial time algorithm with a $2\sqrt{2}/3 + \epsilon$ approximation factor for MAX-VOL would thus decide X3C, which would imply $P = NP$. □

4.3 The Greedy Approximation Algorithm for MAX-VOL

Having shown that the decision problem MAX-VOL is NP-hard, it has two natural interpretations as an optimization problem for a given matrix A:

1. *MAX-VOL(k)*: Given k, find a subset of size k with maximum volume.
2. *MaxSubset(V)*: Given V and that A has unit norm vectors, find the largest subset $C \subseteq A$ with volume at least V.

The natural question is whether there exists a simple heuristic with some approximation guarantee. One obvious strategy is the following greedy algorithm which was also proposed in [11] to construct QR factorizations of matrices:

Algorithm 5 Greedy

```
C ← ∅
while |C| < k do
    Select largest norm vector v ∈ A
    Remove the projection of v from every element of A
    C ← C ∪ v
end while
```

We would like to note that one can obtain a result related to the approximation ratio of Greedy which is implicit in [37] via the following theorem:

Theorem 4.4. [37] *For a matrix $A \in \mathbb{R}^{n\times n}$ and an integer $k(1 \leq k < n)$, let the first k columns of $A\Pi$ be the columns chosen by Greedy where $\Pi \in \mathbb{R}^{n\times n}$ is a permutation matrix and*

$$A\Pi = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix}$$

Then, $\sigma_i(R_{11}) \geq \sigma_i(A)/(\sqrt{n-i}\,2^i)$ for $1 \leq i \leq k$.

Based on this theorem, one can easily derive the following result.

Theorem 4.5. *Greedy has approximation ratio $O(2^{-k(k-1)/2}n^{-k/2})$.*

Proof. Let C be the first k columns of $A\Pi$, i.e. the columns chosen by Greedy. Since, Q is orthogonal, we have

$$\begin{aligned} Vol(C) = Vol(R_{11}) &= \prod_{i=1}^{k} \sigma_i(R_{11}) \geq \prod_{i=1}^{k} \sigma_i(A)/(\sqrt{n-i}\,2^i) \\ &\geq 2^{-k(k-1)/2} \left(\prod_{i=1}^{k} \sigma_i(A)/n^{1/2} \right) \\ &\geq 2^{-k(k-1)/2} n^{-k/2} \left(\prod_{i=1}^{k} \sigma_i(A) \right) \\ &\geq 2^{-k(k-1)/2} n^{-k/2} \cdot Vol_{max} \end{aligned}$$

where Vol_{max} is the maximum possible volume a subset can attain. □

This analysis is loose as the volume $\prod_{i=1}^{k} \sigma_i(A)$ may not be attainable using k columns of A. One major problem with this bound is that it has exponential dependence on n. Our (almost) tight analysis will provide an improvement on the approximation ratio of the theorem above in two ways: first, we will remove the dependence on n, and second we get better than quadratic dependence on k in the exponent. The outline of the remainder of this section is as follows: In Section 3.1, we analyze the performance ratio of Greedy. Section 3.2 presents an explicit example for which Greedy is bad. We analyze Greedy for MaxSubset(V) in Section 3.3 where we require the columns of the matrix be unit norm, in which case the volume is monotonically non-increasing or non-decreasing in the number of vectors chosen by any algorithm.

4.3.1 Approximation Ratio of Greedy

We consider Greedy after k steps. First, we assume that the dimension of the space spanned by the column vectors in A is at least k, since otherwise there is nothing to prove. Let $span(S)$ denote the space spanned by the vectors in the set S, and let $\pi_S(v)$ be the projection of v onto $span(S)$. In this section, let $d(v, S) = \|v - \pi_S(v)\|$ be the norm of the part of v orthogonal to $span(S)$. Let $V_k = \{v_1, \ldots, v_k\}$ be the set of vectors in order that have been chosen by the greedy algorithm at the

end of the k^{th} step. Let $W_k = \{w_1, \ldots, w_k\}$ be a set of k vectors of maximum volume. Our main result in this subsection is the following theorem:

Theorem 4.6. $Vol(V_k) \geq 1/k! \cdot Vol(W_k)$.

We prove the theorem through a sequence of lemmas. The basic idea is to show that at the j^{th} step, Greedy loses a factor of at most j to the optimal. Theorem 4.6 then follows by an elementary induction. First, define $\alpha_i = \pi_{(V_{k-1})}(w_i)$ for $i = 1, \ldots, k$. α_i is the projection of w_i onto $span(V_{k-1})$ where $V_{k-1} = \{v_1, \ldots, v_{k-1}\}$. Let $\beta_i = w_i - \pi_{(V_{k-1})}(w_i)$. Hence, we have

$$w_i = \alpha_i + \beta_i \qquad for\ i = 1, \ldots, k. \tag{4.2}$$

Note that the dimension of $span(V_{k-1})$ is $k-1$, which means that the α_i's are linearly dependent. We will need some stronger properties of the α_i's.

Definition 4.7. *A set of m vectors is said to be in general position, if they are linearly dependent and any $m-1$ element subset of them are linearly independent.*

It's immediate from Definition 4.7 that

Remark 4.8. *Let $U = \{\gamma_1, \ldots, \gamma_m\}$ be a set of m vectors in general position. Then, γ_i can be written as a linear combination of the other vectors in U, i.e.*

$$\gamma_i = \sum_{l \neq i} \lambda_l^i \gamma_l \tag{4.3}$$

for $i = 1, \ldots, m$. λ_l^i's are the coefficients of γ_l in the expansion of γ_i.

Lemma 4.9. *Let $U = \{\gamma_1, \ldots, \gamma_m\}$ be a set of m vectors in general position. Then, there exists a γ_i such that $|\lambda_j^i| \leq 1$ for all $j \neq i$.*

Proof. Assume, without loss of generality that $A = \{\gamma_2, \gamma_3, \ldots, \gamma_m\}$ has the greatest volume among all possible $m-1$ element subsets of U. We claim that γ_1 has the desired property. Consider the set $B_j = \{\gamma_1, \ldots \gamma_{j-1}, \gamma_{j+1}, \ldots, \gamma_m\}$ for $2 \leq j \leq m$. Let $C_j = A - \{\gamma_j\} = B_j - \{\gamma_1\}$. Then, since A has the greatest volume, $Vol(A) = Vol(C_j) \cdot d(\gamma_j, C_j) \geq Vol(B_j) = Vol(C_j) \cdot d(\gamma_1, C_j)$. Hence, we have $d(\gamma_j, C_j) \geq d(\gamma_1, C_j)$. Then, using (4.3), we can write

$$\gamma_1 = \lambda_j^1 \gamma_j + \sum_{l \neq j, l \neq 1} \lambda_l^1 \gamma_l \tag{4.4}$$

Denoting $\delta_j = \pi_{C_j}(\gamma_j)$ and $\theta_j = \gamma_j - \delta_j$, (4.4) becomes

$$\gamma_1 = \left(\lambda_j^1 \delta_j + \sum_{l \neq j, l \neq 1} \lambda_l^1 \gamma_l \right) + \lambda_j^1 \theta_j$$

where the term in parentheses is in $span(C_j)$. Hence, the part of γ_1 which is not in $span(C_j)$, $\theta_1 = \gamma_1 - \pi_{C_j}(\gamma_1) = \lambda_j^1 \theta_j$ and so $\|\theta_1\| = |\lambda_j^1| \|\theta_j\|$. Note that $\|\theta_1\| = d(\gamma_1, C_j)$ and $\|\theta_j\| = d(\gamma_j, C_j)$, so $d(\gamma_1, C_j) = |\lambda_j^1| d(\gamma_j, C_j)$. Since $d(\gamma_1, C_j) \leq d(\gamma_j, C_j)$, we have $|\lambda_j^1| \leq 1$. □

Lemma 4.10. *If $\|\alpha_i\| > 0$ for $i = 1, \ldots, k$ and $k \geq 2$, then there exists a set of m vectors $U = \{\alpha_{i_1}, \ldots, \alpha_{i_m}\} \subseteq \{\alpha_1, \ldots, \alpha_k\}$ with $m \geq 2$ that are in general position.*

Proof. Note that the cardinality of a set U with the desired properties should be at least 2, since otherwise there is nothing to prove. We argue by induction on k. For the base case $k = 2$, we have two vectors α_1 and α_2 spanning a 1-dimensional space and clearly any one of them is linearly independent since neither is 0. Assume that, as the induction hypothesis, any set of $k \geq 2$ non-zero vectors $\{\alpha_1, \ldots, \alpha_k\}$ spanning at most a $k-1$ dimensional space has a non-trivial subset in general position. Consider a $k+1$ element set $A = \{\alpha_1, \ldots, \alpha_{k+1}\}$ with $dim(span(A)) \leq k$. If the vectors in A are not in general position, then there is a k element subset A' of A which is linearly dependent. Hence, $dim(span(A')) \leq k-1$ for which, by the induction hypothesis, we know that there exists a non-trivial subset in general position. □

The existence of a subset in general position guaranteed by Lemma 4.10 will be needed when we apply the next lemma.

Lemma 4.11. *Assume $\|\alpha_i\| > 0$ for $i = 1, \ldots, k$. Then, there exists an α_{i_j} such that $d(\alpha_{i_j}, W'_{k-1}) \leq (m-1) \cdot d(v_k, V_{k-1})$, where $W'_{k-1} = W_k - \{w_{i_j}\}$.*

Proof. Let $U = \{\alpha_{i_1}, \ldots, \alpha_{i_m}\} \subseteq \{\alpha_1, \ldots, \alpha_k\}$ be in general position where $m \geq 2$ (the existence of U is given by Lemma 4.10). Assume α_{i_1} has the property given

by Lemma 4.9. Let $U' = \{w_{i_2}, \ldots, w_{i_m}\}$. We claim that α_{i_1} has the desired property. First, note that $d(\alpha_{i_1}, W'_{k-1}) \leq d(\alpha_{i_1}, U')$, since $span(U')$ is a subspace of $span(W'_{k-1})$. We seek a bound on $d(\alpha_{i_1}, W'_{k-1})$. Using (4.3) and (4.2), we have

$$\alpha_{i_1} = \sum_{l \neq 1} \lambda^1_{i_l} \alpha_{i_l} = \sum_{l \neq 1} \lambda^1_{i_l} (w_{i_l} - \beta_{i_l}).$$

where α_{i_l}'s are the vectors in U and β_{i_l}'s are their orthogonal parts. Rearranging,

$$\sum_{l \neq 1} \lambda^1_{i_l} \beta_{i_l} = \left(\sum_{l \neq 1} \lambda^1_{i_l} w_{i_l} \right) - \alpha_{i_1}.$$

Note that the right hand side is an expression for the difference between a vector in $span(U')$ and α_{i_1}. Hence,

$$\begin{aligned}
d(\alpha_{i_1}, W'_{k-1}) \leq d(\alpha_{i_1}, U') &= \min_{v \in span(U')} \|v - \alpha_{i_1}\| \\
&\leq \| \sum_{l \neq 1} \lambda^1_{i_l} w_{i_l} - \alpha_{i_1} \| \\
&= \| \sum_{l \neq 1} \lambda^1_{i_l} \beta_{i_l} \| \\
&\leq \sum_{l \neq 1} \lambda^1_{i_l} \|\beta_{i_l}\| \\
&\leq (m-1) \cdot max_{1 \leq l \leq m} \|\beta_{i_l}\| \\
&\leq (m-1) \cdot d(v_k, V_{k-1}).
\end{aligned}$$

where the last two inequalities follow from Lemma 4.9 and the greedy property of the algorithm, respectively. □

Before stating the final lemma, which gives the approximation factor of Greedy at each round, we need the following simple observation.

Lemma 4.12. *Let u be a vector, V and W be subspaces and $\alpha = \pi_V(u)$. Then $d(u, W) \leq d(u, V) + d(\alpha, W)$.*

Proof. Let $\gamma = \pi_W(\alpha)$. By triangle inequality for vector addition, we have

$\|u - \gamma)\| \leq \|u - \alpha\| + \|\alpha - \gamma\| = d(u, V) + d(\alpha, W)$. The result follows since $d(u, W) \leq \|u - \gamma\|$. □

Lemma 4.13. *At the k^{th} step of Greedy, there exists a w_i such that $d(w_i, W'_{k-1}) \leq k \cdot d(v_k, V_{k-1})$ where $W'_{k-1} = W_k - \{w_i\}$.*

Proof. For $k = 1$, there's nothing to prove. For $k \geq 2$, there are two cases.

1. One of the w_i's is orthogonal to V_{k-1} ($\|\alpha_i\| = 0$). In this case, by the greedy property, $d(v_k, V_{k-1}) \geq \|w_i\| \geq d(w_i, W'_{k-1})$, which gives the result.

2. For all w_i, $\|\alpha_i\| > 0$, i.e., all w_i have non-zero projection on V_{k-1}. Assuming that $\alpha_1 = \pi_{V_{k-1}}(w_1)$ has the desired property proved in Lemma 4.11, we have for the corresponding w_1

$$\begin{aligned} d(w_1, W'_{k-1}) &\leq d(w_1, V_{k-1}) + d(\alpha_1, W'_{k-1}) \\ &\leq \|\beta_1\| + d(\alpha_1, W'_{k-1}) \\ &\leq \|\beta_1\| + (m-1) \cdot d(v_k, V_{k-1}) \\ &\leq m \cdot d(v_k, V_{k-1}). \end{aligned}$$

The first inequality is due to Lemma 4.12. The last inequality follows from the greedy property of the algorithm, i.e. the fact that $d(v_k, V_{k-1}) \geq \|\beta_1\|$. The lemma follows since $m \leq k$. □

The last lemma immediately leads to the result of Theorem 4.6, with a simple inductive argument as follows:

Proof. The base case is easily established since $Vol(V_1) = Vol(W_1)$. Assume that $Vol(V_{k-1}) \geq 1/(k-1)! \cdot Vol(W_{k-1})$ for some $k > 2$. By Lemma 4.13, we have a w_i such that $d(w_i, W'_{k-1}) \leq k \cdot d(v_k, V_{k-1})$ where $W'_{k-1} = W_k - \{w_i\}$. It follows that

$$\begin{aligned} Vol(V_k) &= d(v_k, V_{k-1}) \cdot Vol(V_{k-1}) \\ &\geq \frac{d(w_i, W'_{k-1})}{k} \cdot \frac{Vol(W_{k-1})}{(k-1)!} \\ &\geq \frac{d(w_i, W'_{k-1})}{k!} \cdot Vol(W'_{k-1}) \\ &= \frac{Vol(W_k)}{k!}. \end{aligned}$$

□

Theorem 4.6 combined with Theorem 2.4, we obtain the following result for the RRQR factorization provided by the greedy algorithm:

Theorem 4.14. *For a matrix $A \in \mathbb{R}^{n\times n}$, and an integer k $(1 \leq k < n)$, let $\Pi \in \mathbb{R}^{n\times n}$ be a permutation matrix such that the first k columns of $A\Pi$ are chosen by Greedy k steps run. Then, for the QR factorization*

$$A\Pi = Q \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix},$$

we have $\sigma_{min}(R_{11}) \geq (1/\sqrt{k(n-k)(k!)^2 + 1})\sigma_k(A)$ and $\sigma_1(R_{22}) \leq \sqrt{k(n-k)(k!)^2 + 1}\,\sigma_{k+1}(A)$.

4.3.2 Lower Bound for Greedy

We give a lower bound of $1/2^{k-1}$ for the approximation factor of Greedy by explicitly constructing a bad example. We will inductively construct a set of unit vectors satisfying this lower bound. It will be the case that the space spanned by the vectors in the optimal solution is the same as the space spanned by the vectors chosen by Greedy. An interesting property of our construction is that both the optimal volume and the volume of the vectors chosen by Greedy approach 0 in the limit of a parameter δ, whereas their ratio approaches to $1/2^{k-1}$.

We will first consider the base case $k = 2$: let the matrix $A = [v_1 w_1 w_2]$ where $dim(A) = 2$ and $d(v_1, w_1) = d(v_1, w_2) = \delta$ for some $1 > \delta > 0$ such that θ, the angle between w_1 and w_2 is twice the angle between v_1 and w_1, i.e. v_1 is 'between' w_1

and w_2. If the greedy algorithm first chooses v_1, then $\lim_{\delta \to 0} Vol(V_2)/Vol(W_2) = 1/2 \cos\theta/2 = 1/2$. Hence, for $k = 2$, there's a set of vectors for which $Vol(W_2) = (2-\epsilon) \cdot Vol(V_2)$ for arbitrarily small $\epsilon > 0$.

For arbitrarily small $\epsilon > 0$, assume that there is an optimal set of k vectors $W_k = \{w_1, \ldots, w_k\}$ such that $Vol(W_k) = (1-\epsilon)2^{k-1} \cdot Vol(V_k)$ where $V_k = \{v_1, \ldots, v_k\}$ is the set of k vectors chosen by Greedy. The vectors in W_k and V_k span a subspace of dimension k, and assume $w_i \in \mathbb{R}^d$ where $d > k$. Let $d(v_2, V_1) = \epsilon_1 = \delta$ for some $1 > \delta > 0$, and $d(v_{i+1}, V_i) = \epsilon_i = \delta\epsilon_{i-1}$ for $i = 2, \ldots, k-1$. Thus, $Vol(V_k) = \delta^{k(k-1)/2}$ and $Vol(W_k) = (1-\epsilon)2^{k-1}\delta^{k(k-1)/2}$. Assume further that for all w_i in W_k, $d(w_i, V_j) \leq \epsilon_j$ for $j = 1, \ldots, k-2$ and $d(w_i, V_{k-1}) = \epsilon_{k-1}$ so that there exists an execution of Greedy where no $\{w_1, \ldots, w_k\}$ is chosen.

We will now construct a new set of vectors $W_{k+1} = W'_k \cup \{w_{k+1}\} = \{w'_1, \ldots w'_k, w_{k+1}\}$ which will be the optimal solution. Let $w_i^j = \pi_{V_j}(w_i)$, and let $e_i^j = \pi_{V_j}(w_i) - \pi_{V_{j-1}}(w_i)$ for $j = 2, \ldots, k$ and $e_i^1 = w_i^1$. Namely, e_j^i is the component of w_i which is in V_j, but perpendicular to V_{j-1} and e_i^1 is the component of w_i which in in the span of v_1. (Note that $\|e_i^k\| = \epsilon_{k-1}$.) Let u be a unit vector perpendicular to $span(W_k)$. For each w_i we define a new vector $w'_i = (\sum_{j=1}^{k-1} e_i^j) + \sqrt{1-\delta^2} e_i^k + \delta\epsilon_{k-1}u$. Intuitively, we are defining a set of new vectors which are first rotated towards V_{k-1} and then towards u such that they are $\delta\epsilon_{k-1}$ away from V_k. Introduce another vector $w_{k+1} = \sqrt{1-\delta^2} v_1 - \delta\epsilon_{k-1}u$. Intuitively, this new vector is v_1 rotated towards the negative direction of u. Note that, in this setting $\epsilon_k = \delta\epsilon_{k-1}$. We finally choose $v_{k+1} = w_{k+1}$.

Lemma 4.15. *For any $w \in W_{k+1}$, $d(w, V_j) \leq \epsilon_j$ for $j = 1, \ldots, k-1$ and $d(w, V_k) = \epsilon_k$.*

Proof. For $w = w_{k+1}$, $d(w_{k+1}, V_j) = \epsilon_k \leq \epsilon_j$ for $j = 1, \ldots, k$. Let $w = w'_i$ for some $1 \leq i \leq k$. Then, for any $1 \leq j \leq k-1$, we have $d(w'_i, V_j)^2 = \sum_{l=j+1}^{k-1} \|e_i^l\|^2 + (1-\delta^2)\|e_i^k\|^2 + \delta^2\|e_i^k\|^2 = \sum_{l=j+1}^{k} \|e_i^l\|^2 = d(w_i, V_j)^2 \leq {\epsilon_j}^2$ by the induction hypothesis. □

Lemma 4.15 ensures that $\{v_1, \ldots, v_{k+1}\}$ is a valid output of Greedy. What remains is to show that for any $\epsilon > 0$, we can choose δ sufficiently small so that

$Vol(W_{k+1}) \geq (1-\epsilon)2^k \cdot Vol(V_{k+1})$. In order to show this, we will need the following lemmas.

Lemma 4.16. $\lim_{\delta \to 0} Vol(W_{k+1}) = 2\epsilon_k \cdot Vol(W_k)$.

Proof. With a little abuse of notation, let W_{k+1} denote the matrix of coordinates for the vectors in the set W_{k+1}.

$$W_{k+1} = \begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,k} & \sqrt{1-\delta^{2k}} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,k} & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sqrt{1-\delta^2}w_{k,1} & \sqrt{1-\delta^2}w_{k,2} & \cdots & \sqrt{1-\delta^2}w_{k,k} & 0 \\ \delta^k & \delta^k & \cdots & \delta^k & -\delta^k \end{pmatrix}$$

where $w_{i,j}$ is the i^{th} coordinate of w_j, which is in W_k. (Note that this is exactly how U is constructed in the inductive step). Expanding on the right-most column of the matrix, we have

$$Vol(W_{k+1}) = |det(W_{k+1})| = |\sqrt{1-\delta^{2k}} \cdot det(A) + (-1)^{k+1}\delta^k \cdot det(B)| \tag{4.5}$$

where A and B are the corresponding minors of the coefficients, i.e. the left-most lower and upper $k \times k$ sub-matrices of W_{k+1}, respectively. Clearly, we have $det(B) = \sqrt{1-\delta^2} \cdot det(W_k)$ where W_k is the matrix of coordinates for the vectors in the set W_k. Let C be the matrix obtained by replacing each $w_{1,i}$ by 1 in W_k. Then, using row interchange operations on A, we can move the last row of A to the top. This gives a sign change of $(-1)^{k-1}$. Then, factoring out $\sqrt{1-\delta^2}$ and δ^k in the first and last rows respectively, we have $det(A) = (-1)^{k-1}\delta^k\sqrt{1-\delta^2} \cdot det(C)$. Hence, (4.5) becomes

$$|det(W_{k+1})| = (\delta^k\sqrt{1-\delta^2})|\sqrt{1-\delta^{2k}} \cdot det(C) + det(W_k)| \tag{4.6}$$

We will need the following lemma to compare $det(W_k)$ and $det(C)$.

Lemma 4.17. $\lim_{\delta\to 0} \det(C)/\det(W_k) = 1$.

Proof. For $i > 1$, the elements of the i^{th} rows of both W_k and C has δ^{i-1} as a common coefficient by construction. Factoring out these common coefficients, we have $det(W_k) = \delta^{k(k-1)/2} \cdot det(U)$ and $det(C) = \delta^{k(k-1)/2} \cdot det(U')$ where U and U' are matrices with non-zero determinants as δ approaches 0. Furthermore, $lim_{\delta\to 0} det(U) = det(U')$ as the elements in the first row of U approaches 1. The result then follows. □

Using Lemma 4.17 and (4.6), we have

$$lim_{\delta\to 0} Vol(W_{k+1}) = lim_{\delta\to 0} |det(W_{k+1})| = 2\delta^k |det(W_k)| = 2\epsilon_k \cdot Vol(W_k)$$

□

Theorem 4.18. *$Vol(W_{k+1}) \geq (1-\epsilon)2^k \cdot Vol(V_{k+1})$ for arbitrarily small $\epsilon > 0$.*

Proof. Given any $\epsilon' > 0$ we can choose δ small enough so that $Vol(W_{k+1}) \geq 2\epsilon_k(1-\epsilon') \cdot Vol(W_k)$, which is always possible by Lemma 4.16. Given any ϵ'', we can apply induction hypothesis to obtain V_k and W_k such that $Vol(W_k) \geq (1-\epsilon'')2^{k-1} \cdot Vol(V_k)$. Thus,

$$\begin{aligned} Vol(W_{k+1}) &\geq 2\epsilon_k(1-\epsilon') \cdot Vol(W_k) \\ &\geq 2\epsilon_k(1-\epsilon')(1-\epsilon'')2^{k-1} \cdot Vol(V_k) \\ &= (1-\epsilon')(1-\epsilon'')2^k \cdot Vol(V_{k+1}), \end{aligned}$$

where we have used $Vol(V_{k+1}) = \epsilon_k \cdot Vol(V_k)$. Choosing ϵ' and ϵ'' small enough such that $(1-\epsilon')(1-\epsilon'') > 1-\epsilon$ gives the result. □

4.3.3 Maximizing the Number of Unit norm Vectors Attaining A Given Volume

In this section, we give a result on approximating the maximum number of unit norm vectors which can be chosen to have at least a certain volume. This

result is essentially a consequence of the previous approximation result. We assume that all the vectors in A have unit norm, hence the volume is non-increasing in the number of vectors chosen by Greedy. Let OPT_k denote the optimal volume for k vectors. Note that $OPT_k \geq OPT_{k+1}$ and the number of vectors m, chosen by Greedy attaining volume at least OPT_k is not greater than k. Our main result states that, if the optimal volume of k vectors is $2^{-\Omega(k)}$, then Greedy chooses $\Omega(k/\log k)$ vectors having at least that volume. Thus, Greedy gives a $\log k$-approximation to the optimal number of vectors. We prove the result through a sequence of lemmas. The following lemma is an immediate consequence of applying Greedy on W_k.

Lemma 4.19. *Let $W_k = \{w_1, \ldots, w_k\}$ be a set of k vectors of optimal volume OPT_k. Then there exists a permutation π of the vectors in W_k such that $d_{\pi(k)} \leq d_{\pi(k-1)} \leq \ldots \leq d_{\pi(2)}$ where $d_{\pi_i} = d(w_{\pi_i}, \{w_{\pi_1}, \ldots, w_{\pi_{i-1}}\})$ for $k \geq i \geq 2$.*

We use this existence result to prove the following lemma.

Lemma 4.20. *$OPT_m \geq (OPT_k)^{(m-1)/(k-1)}$ where $m \leq k$.*

Proof. Let $W_k = \{w_1, \ldots, w_k\}$ be a set of vectors of optimal volume OPT_k. By Lemma 4.19, we know that there exists an ordering of vectors in W_k such that $d_{\pi(k)} \leq d_{\pi(k-1)} \leq \ldots \leq d_{\pi(2)}$ where $d_{\pi_i} = d(w_{\pi_i}, \{w_{\pi_1}, \ldots, w_{\pi_{i-1}}\})$ for $k \geq i \geq 2$. Let $W_m{}' = \{w_{\pi(1)}, \ldots, w_{\pi(m)}\}$. Then, we have $OPT_m \geq Vol(W_m{}') = \prod_{i=2}^{m} d_{\pi_i} \geq (\prod_{i=2}^{k} d_{\pi_i})^{(m-1)/(k-1)} = (OPT_k)^{(m-1)/(k-1)}$. □

Lemma 4.21. *Suppose $OPT_k \leq 2^{(k-1)m\log m/(m-k)}$. Then, the greedy algorithm chooses at least m vectors whose volume is at least OPT_k.*

Proof. We are seeking a condition for OPT_k which will provide a lower bound for m such that $OPT_m/m! \geq OPT_k$. If this holds, then $Vol(Greedy_m) \geq Opt_m/m! \geq OPT_k$ and so Greedy can choose at least m vectors which have volume at least OPT_k. It suffices to find such an m satisfying $(OPT_k)^{(m-1)/(k-1)}/m! \geq OPT_k$ by Lemma 4.20. This amounts to $1/m! \geq (OPT_k)^{1-(m-1)/(k-1)}$. Since $1/m! \geq 1/m^m$ for $m \geq 1$, we require $1/m^m \geq (OPT_k)^{1-(m-1)/(k-1)}$. Taking logarithms of both sides and rearranging, we have $-(k-1)m\log m/(k-m) \geq \log OPT_k$. Taking exponents of both sides yields $2^{(k-1)m\log m/(m-k)} \geq OPT_k$. □

In order to interpret this result, we will need to restrict OPT_k. Otherwise, for example if $OPT_k = 1$, the greedy algorithm may never get more than 1 vector to guarantee a volume of at least OPT_k since it might be possible to misguess the first vector. In essence, the number of vectors chosen by the algorithm depends on OPT_k. First, we discuss what is a reasonable condition on OPT_k. Consider n vectors in m dimensions which defines a point in $\mathbb{R}^{m\times n}$. The set of points in which any two vectors are orthogonal has measure 0. Thus, define $2^{-\alpha} = \max_{ij} d(v_i, v_j)$. Then, it is reasonable to assume that $\alpha > 0$, in which case $OPT_k \leq 2^{-\alpha k} = 2^{-\Omega(k)}$. Hence, we provide the following theorem which follows from the last lemma under the reasonable assumption that the optimal volume decreases by at least a constant factor with the addition of one more vector.

Theorem 4.22. *If $OPT_k \leq 2^{-\Omega(k)}$, then the greedy algorithm chooses $\Omega(k/\log k)$ vectors having volume at least OPT_k.*

Proof. For some α, $OPT_k \leq 2^{-\alpha k}$. Thus, we solve for m such that $2^{-\alpha k} \leq 2^{(k-1)m\log m/(m-k)}$. Suitable rearrangements yield

$$m \leq \frac{\alpha k(k-m)}{(k-1)\log m} \leq \frac{2\alpha k}{\log m}$$

.

For m, the largest integer such that $m \leq 2\alpha k/\log m$, we have

$$m \approx \frac{2\alpha k}{\log(2\alpha k/\log m)} = \frac{2\alpha k}{\log(2\alpha k) - \log\log m} = \Omega\left(\frac{k}{\log k}\right).$$

□

In reality, for a random selection of n vectors in m dimensions, α will depend on n and so the result is not as strong as it appears.

4.4 Discussion

Our analysis of the approximation ratio relies on finding the approximation factor at each round of Greedy. Indeed, we have found examples for which the volume of the vectors chosen by Greedy falls behind the optimal volume by as large

a factor as $1/k$, making Lemma 4.13 tight. But it might be possible to improve the analysis by correlating the 'gains' of the algorithm between different steps. Hence, one of the immediate questions is that whether one can close the gap between the approximation ratio and the lower bound for Greedy. We conjecture that the approximation ratio is $1/2^{k-1}$.

We list other open problems as follows:

- Do there exist efficient non-greedy algorithms with better guarantees for MAX-VOL?

- Volume seems to play an important role in constructing a low-rank approximation to a matrix. Can the result of [35] be extended to yield a direct relationship between low-rank approximations and large volume $m \times k$ sub-matrices of a matrix? Or, can we establish a result stating that there must exist a large volume $k \times k$ sub-matrix of a large volume $m \times k$ sub-matrix such that one can find an approximation to the maximum volume $k \times k$ sub-matrix by running the same algorithm again on the $m \times k$ sub-matrix?

We would like to note that the approximation ratio of Greedy algorithm is considerably small because of the 'multiplicative' nature of the problem. Another important problem which resembles MAX-VOL in terms of behavior (but not necessarily in nature) is the Shortest Vector Problem (SVP), which is not known to have a polynomial factor approximation algorithm. Indeed, the most common algorithm which works well in practice has a $2^{O(n)}$ approximation ratio [46] and non-trivial hardness results for this problem are difficult to find.

CHAPTER 5
Exponential Inapproximability of MAX-VOL

This chapter proves the exponential inapproximability for MAX-VOL. We provide a gap preserving reduction from the well known Label-Cover problem using the Parallel Repetition Theorem [52]. In doing so, we will establish that the greedy algorithm from previous chapter is about the best we could hope for.

5.1 Introduction

First, we briefly review simple facts about the inapproximability of NP-hard problems. The following information can be found in a standard text about approximation algorithms, e.g. [60]. Since we only consider maximization problems in this chapter, we give all the definitions accordingly.

Consider a maximization problem Π for which we want to show an inapproximability result. Namely, we would like to prove that there is no polynomial time c approximation algorithm for Π assuming a complexity theoretic conjecture[3]. Assuming x is an instance of Π, let $OPT(x)$ denote the optimal value of problem Π on x. There are two ways of proving inapproximability results:

1. *Gap producing reduction:* This is a polynomial time reduction from an NP-complete problem to Π. Assume, for example, a reduction from SAT to Π. This type of reduction maps an instance ϕ of SAT to x of Π. It comes with two parameters f and α, and it satisfies

   - if ϕ is satisfiable, $OPT(x) \geq f(x)$,
   - if ϕ is not satisfiable, $OPT(x) < \alpha \cdot f(x)$.

   Note that this is the type of reduction that we provided in the previous chapter to show inapproximability for MAX-VOL. This type of reduction shows α inapproximability for Π.

[3]The conjecture is generally $P \neq NP$ in most cases. But weaker assumptions are also common. c can be a function of the input

2. *Gap preserving reduction:* This is a polynomial time reduction from a maximization problem Π_1 to another maximization problem Π_2. It has four parameters f_1, α, f_2 and β. Given an instance x of Π_1, it computes an instance y of Π_2 such that

   - if $OPT(x) \geq f_1(x)$, then $OPT(y) \geq f_2(y)$,
   - if $OPT(x) < \alpha \cdot f_1(x)$, then $OPT(y) < \beta \cdot f_2(y)$.

   Given that Π_1 is α inapproximable, this reduction shows β inapproximability for Π_2.

The following can be easily obtained and is quite frequently used in proving inapproximability results:

Fact 5.1. *If there is a gap producing reduction Γ from SAT to Π_1 with parameters f_1 and α, and a gap preserving reduction Γ' from Π_1 to Π_2 with parameters f_1, α, f_2 and β, then there exists a gap producing reduction Γ'' from SAT to Π_2 with parameters f_2 and β.*

5.1.1 Preliminaries and Notation

As a reminder, we introduce some preliminary notation. Let $A = \{v_1, v_2, \ldots, v_n\}$ be given in column notation. The volume of A, $Vol(A)$ can be recursively defined as follows: if A contains one column, i.e. $A = \{v_1\}$, then $Vol(A) = \|v\|_2$, where $\|\cdot\|_2$ is the Euclidean norm. If A has more than one column, $Vol(A) = \|v - \pi_{(A-\{v\})}(v)\|_2 \cdot Vol(A - \{v\})$ for any $v \in A$, where $\pi_A(v)$ is the projection of v onto the space spanned by the column vectors of A. It is well known that $\pi_{(A-\{v\})}(v) = A_v A_v^+ v$, where A_v is the matrix whose columns are the vectors in $A - \{v\}$, and A_v^+ is the pseudo-inverse of A_v (see for example [33]). Using this recursive expression, we have

$$Vol(S) = Vol(A) = \|v_1\|_2 \cdot \prod_{i=1}^{n-1} \|v_{i+1} - A_i A_i^+ v_{i+1}\|_2$$

where $A_i = [v_1 \cdots v_i]$ for $\leq i \leq n-1$.

We will also need the definition of "distance" of a vector to a subspace, and the following lemma which will simplify the final proof:

Lemma 5.2 (Union Lemma)**.** *Given two sets of vectors P and $Q = \{q_1, \ldots, q_m\}$, let $d(q, P) = \|q - \pi_P(q)\|_2$ denote the distance of $q \in Q$ to the space spanned by the vectors in P. Then, $Vol(P \cup Q) \leq Vol(P) \cdot \prod_{i=1}^{n} d(q_i, P)$.*

Proof. We argue by induction on m. For $m = 1$, Q has one element and the statement trivially holds. Assume that it is true for $n = k$ where $Q = \{q_1, \ldots, q_k\}$. Then, for any q_{k+1}

$$\begin{aligned} Vol(P \cup Q \cup \{q_{k+1}\}) &=_{(a)} Vol(P \cup Q) \cdot d(q_{k+1}, P \cup Q) \\ &\leq_{(b)} Vol(P \cup Q) \cdot d(q_{k+1}, P) \\ &\leq Vol(P) \cdot \prod_{i=1}^{k} d(q_i, P) \cdot d(q_{k+1}, P) \\ &= Vol(P) \cdot \prod_{i=1}^{k+1} d(q_i, P). \end{aligned}$$

(a) follows because $d(q, A \cup B) \leq d(q, A)$ for any A, B and (b) follows by the induction hypothesis. □

5.2 Label-Cover Problem

Our reduction will be from the Label Cover problem. Label Cover combinatorially captures the expressive power of a 2-prover 1-round proof system for the problem Max-3SAT(5). Specifically, there exists a reduction from Max-3SAT(5) to Label Cover and the well known parallel repetition technique for the specified proof system corresponds to a new k-fold Label Cover instance. For simplicity, we prefer to state our reduction from Label Cover and for the sake of completeness, we provide the canonical reduction from Max-3SAT(5) to Label Cover.

Max-3SAT(5) is defined as follows: Given a set of $5n/3$ variables and n clauses in conjunctive normal form where each clause contains three distinct variables and each variable appears in exactly five clauses, find an assignment of variables such

that it maximizes the fraction of satisfied clauses. The following result is well known [4, 5]:

Theorem 5.3. *There is a constant $\epsilon > 0$, such that it is NP-hard to distinguish between the instances of Max-3SAT(5) having optimal value* 1 *and optimal value at most* $(1 - \epsilon)$.

Although the above result was proved for general 3CNF formulas, without the requirement that each variable appears exactly 5 times, there is a standard reduction from Max-3SAT to Max-3SAT(5) [29], which only results in a difference in the constant ϵ.

A Label Cover instance L is defined as follows: $L = (G(V, W, E), (\Sigma_V, \Sigma_W), \Pi)$ where

- $G(V, W, E)$ is a regular bipartite graph with vertex sets V and W, and the edge set E.
- Σ_V and Σ_W are the label sets associated with V and W, respectively.
- Π is the collection of constraints on the edge set, where the constraint on an edge e is defined as a function $\Pi_e : \Sigma_V \to \Sigma_W$.

A labeling is an assignment to the vertices of the graph, $\sigma : \{V \to \Sigma_V\} \cup \{W \to \Sigma_W\}$. It is said to satisfy an edge $e = (v, w)$ if $\Pi_e(\sigma(v)) = \sigma(w)$. The Label Cover problem asks for an assignment σ such that the fraction of the satisfied edges is maximum.

In what follows, we describe a standard reduction from Max-3SAT(5) to Label Cover: Given a Max-3SAT(5) instance with a 3CNF formula ϕ containing n clauses and $5n/3$ variables, let V be the vertices corresponding to each clause and W be the vertices representing the variables. Let there be an edge between $v \in V$ and $w \in W$ if the variable x_w corresponding to w is contained (in any form) in the clause C_v corresponding to v. Hence, we have defined the graph $G(V, W, E)$. Let $\Sigma_V = \{1, \ldots, 7\}$ and let $\Sigma_W = \{1, 2\}$. The labels for a vertex $v \in V$ stand for the 7 different satisfying assignment for C_v in some order. And, the two labels for a vertex $w \in W$ correspond to the assignment given to the variable x_w, i.e. *true* or *false*.

For an edge $e = (v, w)$ and for $7 \geq i \geq 1$, we define $\Pi_e(i) = 1$ if the *ith* satisfying assignment for C_v assigns *false* to x_w, $\Pi_e(i) = 2$ otherwise. Note that, in this Label Cover instance, $|V| = 5n/3$, $|W| = n$, $|E| = 5n$; the degrees of the vertices in V and W is 3 and 5, respectively. The following theorem can easily be derived and it essentially states that the reduction above is a gap preserving one.

Theorem 5.4. *There is a constant $\epsilon' > 0$, such that it is NP-hard to distinguish between the instances of Label Cover having optimal value 1 and optimal value at most $(1 - \epsilon')$.*

In order to amplify the gap, one can define a new Label Cover instance for which the vertex set is essentially a set Cartesian product of the original one. This instance, as follows, captures a standard 2-prover 1-round protocol with parallel repetition ℓ times applied. We first note that for a given set $S = \{s_1, \ldots, s_n\}$, S^ℓ consists of all ℓ-tuples of the form $(s_{i_1}, \ldots, s_{i_\ell})$ where $s_{i_j} \in S$ and i_j runs over $\{1, \ldots, n\}$ for $\ell \geq j \geq 1$. Given the original Label Cover instance $L = (G(V, W, E), (\Sigma_V, \Sigma_W), \Pi)$ reduced from Max-3SAT(5), let $L^\ell = (G^\ell(V^\ell, W^\ell, E^\ell), \Sigma_V^\ell, \Sigma_W^\ell, \Pi^\ell)$ where V^ℓ, W^ℓ, Σ_V^ℓ and Σ_W^ℓ are the ℓ times Cartesian products of the sets V, W, Σ_V and Σ_W, respectively as defined above. Let

- E^ℓ consist of all edges of the form $e = (v, w)$ where $v = (v_{i_1}, \ldots, v_{i_\ell})$ and $w = (w_{i_1}, \ldots, w_{i_\ell})$ satisfying $(v_{i_j}, w_{i_j}) \in E$ and for all $l \geq j \geq 1$.

- Π^ℓ be the collection of constraints on the edge set E^ℓ. The constraint on an edge $e = (v, w)$ where $v = (v_{i_1}, \ldots, v_{i_\ell})$ and $w = (w_{i_1}, \ldots, w_{i_\ell})$ is a function $\Pi_e^\ell : \Sigma_V^\ell \to \Sigma_W^\ell$ which is essentially an l-tuple constraint $(\Pi_{e_1}^\ell, \ldots \Pi_{e_l}^\ell)$, where $\Pi_{e_j}^\ell = \Pi_{(v_{i_j}, w_{i_j})}$ for $\ell \geq j \geq 1$.

A labeling σ of the vertices V^ℓ and W^ℓ is said to satisfy an edge $e = (v, w)$ where $v = (v_{i_1}, \ldots, v_{i_\ell})$ and $w = (w_{i_1}, \ldots, w_{i_\ell})$, if $\Pi_e^\ell(\sigma(v)) = \sigma(w)$. Note that this requirement is equal to $\Pi_{(v_{i_j}, w_{i_j})}(\sigma(v_{i_j})) = \sigma(w_{i_j})$ for all $\ell \geq j \geq 1$. It is easy to see that, in this new Label Cover instance, $|V| = (5n/3)^\ell$, $|W| = n^\ell$, $|E| = (5n)^\ell$, $|\Sigma_V^\ell| = 7^\ell$ and $|\Sigma_W^\ell| = 2^\ell$; the degrees of the vertices in V and W is 3^ℓ and 5^ℓ, respectively. The following theorem is a well known result by Raz [52]:

Theorem 5.5. *There is an absolute constant $\alpha > 0$, such that it is NP-hard to distinguish between the case that $OPT(L^\ell) = 1$ and $OPT(L^\ell) \leq 2^{-\alpha\ell}$.*

5.3 Exponential Inapproximability of MAX-VOL

5.3.1 The Basic Gadget

At the heart of our analysis is a set of vectors with a special property. We will use a set of vectors (composed of binary entries for simplicity of construction) such that any two of them have large dot-product. We will also require that the dot product of a vector and the orthogonal complement of any other vector is also large. More specifically, we need these dot products be proportional to the Euclidean norms squared of the vectors.

Given a vector $v = (v_1 \dots v_n)$ where $v_i \in \{0, 1\}$ for $n \geq i \geq 1$, we denote the orthogonal complement of v by $\overline{v} = (\overline{v_1} \dots \overline{v_n})$ where $\overline{v_i} = 1$ if $v_i = 0$, and $\overline{v_i} = 0$ otherwise. We begin with the following lemma:

Lemma 5.6. *There exists a set of vectors $B = \{b_1, \dots, b_{2n-1}\}$ of dimension 2^n with binary entries such that the following three conditions hold:*

- $\|b_i\|_2 = 2^{(m-1)/2}$ *for* $2m - 1 \geq i \geq 1$
- $b_i \cdot \overline{b_j} = 2^{m-2}$ *for* $2m - 1 \geq i > j \geq 1$.
- $b_i \cdot b_j = 2^{m-2}$ *for* $2m - 1 \geq i > j \geq 1$.

Proof. We argue by induction on m. For $m = 2$, consider the following 3 vectors which clearly satisfy the requirements:

$$p = 0011$$

$$q = 0101$$

$$r = 0110$$

We will use this simple observation in the inductive step. Now, assume the statement holds for $m = k$, i.e. that there exists $B_k = \{b_1, \dots, b_{2k-1}\}$ with the desired

properties. For a vector $v = (v_1 \dots v_{2^k})$ in B_k, define a new vector $v' = (v'_1 \dots v'_{2^{k+1}})$ such that $v'_{2i-1} = v'_{2i} = v_i$ for $2^k \geq i \geq 1$. In words, we define v' by repeating the elements of v twice in place. And, let B'_k be the set of all such vectors. To give an example, let B_2 be the set of three vectors described above. Then, representing each vector row-wise, we have

$$B_2 = \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}, \qquad B'_2 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Note that $\|v'\|_2 = 2^{k/2}$ and $v \cdot \overline{w} = 2^{k-1}$ for all $v', w' \in B'_k$. Consider the last two entries of all the vectors in B_k. By the inductive hypothesis, we have at least one vector in B'_k ending with 0011 (or 1100 which will not change our argument as it is symmetric). Consider the vectors

$$q' = \underbrace{(q \dots q)}_{2^{k-1} \text{ times}} \qquad r' = \underbrace{(r \dots r)}_{2^{k-1} \text{ times}}$$

where q and r are defined as above. It is clear that $\|q'\|_2 = \|r'\|_2 = 2^{k/2}$, $q' \cdot r' = 2^{k-1}$, and $q' \cdot \overline{r'} = 2^{k-1}$. We claim that the set $B_{k+1} = B'_k \cup \{q'\} \cup \{r'\}$ satisfies the desired properties. For convenience, we explicitly show B_3 continuing our example for $k = 2$:

$$B_3 = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

For $v' \in B'_k$, $v' \cdot q'$ and $v' \cdot r'$ is exactly half of the number of 1's in v' since the 0's of v' do not contribute to the dot product and every block of two 1's is multiplied by the block 01 or 10. It is easy to see that this holds for the complement for q' and r', too. Thus, $v' \cdot q' = v' \cdot r' = v' \cdot \overline{q'} = v' \cdot \overline{r'} = 2^{k-1}$ completing our argument. □

The proof of Lemma 5.6 actually yields an algorithm which starts with the vectors p, q, r and inductively constructing the desired set by following the procedure

in the proof. It clearly works in time $O(m2^m)$.

5.3.2 The Reduction

According to Lemma 5.6, for $m = 2^{\ell-1}+1$, there exists a set of binary vectors $B = \{b_1, \ldots, b_{2^\ell}\}$ of dimension $2^{(2^{\ell-1}+1)}$ such that the following three conditions hold:

- $\|b_i\|_2 = 2^{2^{\ell-2}}$ for $2^\ell \geq i \geq 1$
- $b_i \cdot \overline{b_j} = 2^{2^{\ell-1}-1}$ for $2^\ell \geq i > j \geq 1$.
- $b_i \cdot b_j = 2^{2^{\ell-1}-1}$ for $2^\ell \geq i > j \geq 1$.

B can be constructed in time $O(2^\ell 2^{2^\ell})$; note that this is a constant for constant ℓ. For the sake of simplicity of our argument, we normalize the vectors in B, which then clearly satisfies

- $\|b_i\|_2 = 1$ for $2^\ell \geq i \geq 1$
- $b_i \cdot \overline{b_j} = 1/2$ for $2^\ell \geq i > j \geq 1$.
- $b_i \cdot b_j = 1/2$ for $2^\ell \geq i > j \geq 1$.

Given a Max-3SAT(5) instance and the reduction described in the previous section, we will define a column vector for each vertex-label pair in L^ℓ, making $(35n/3)^\ell + (2n)^\ell$ vectors in total. (Note that $|V^\ell| = (5n/3)^\ell$, $|W^\ell| = n^\ell$, $\Sigma_V^\ell = \{1, \ldots, 7^\ell\}$ and $\Sigma_W^\ell = \{1, \ldots 2^\ell\}$). Each vector will be composed of $|E^\ell| = (5n)^\ell$ "blocks" which are either vectors from the set B or the zero vector according to the adjacency information. More specifically, let $A_{v,i}$ be the vector for the vertex label pair $v \in V^\ell$ and $i \in \Sigma_V^\ell$. Similarly let $A_{w,j}$ be the vector for the pair $w \in W$ and $j \in \Sigma_W^\ell$. Both of these vectors are $(5n)^\ell 2^{(2^{\ell-1}+1)}$ dimensional. The block of $A_{v,i}$ corresponding to an edge $e \in E^\ell$ is denoted by $A_{v,i}(e)$. The block of $A_{w,j}$ corresponding to an edge $e \in E^\ell$ is denoted by $A_{w,j}(e)$. We define

$$A_{v,i}(e) = \begin{cases} \dfrac{\overline{b_{\Pi_e^\ell(i)}}}{3^{\ell/2}} & \text{if } e \text{ is incident to } v \\ \\ \overrightarrow{0} & \text{if } e \text{ is not incident to } v. \end{cases}$$

$$A_{w,j}(e) = \begin{cases} \dfrac{b_j}{5^{\ell/2}} & \text{if } e \text{ is incident to } w \\ \\ \overrightarrow{0} & \text{if } e \text{ is not incident to } w \end{cases}$$

In order to show how our reduction works, we present a part of a simple bipartite graph in Figure 5.1 with all the edges drawn between two pairs of nodes, and the corresponding (row) vectors computed by the reduction in Figure 5.2. Note that $A_{v,i}$ has exactly 3^ℓ non-zero blocks, and $A_{w,j}$ has 5^ℓ non-zero blocks. Hence, according to the definition above, their Euclidean norm is 1. The column vector set for the MAX-VOL instance is defined as

$$A \in \mathbb{R}^{M \times N} = \{A_{v,i} | v \in V^\ell, i \in \Sigma_V^\ell\} \cup \{A_{w,j} | w \in W^\ell, j \in \Sigma_W^\ell\}.$$

Note that $M = (5n)^\ell 2^{(2^{\ell-1}+1)}$ and $N = (35n/3)^\ell + (2n)^\ell$, both having polynomial size in n for constant l. From an intuitive point of view, we define mutually orthogonal subspaces for each edge, and then we "spread" the Euclidean norm of each vector

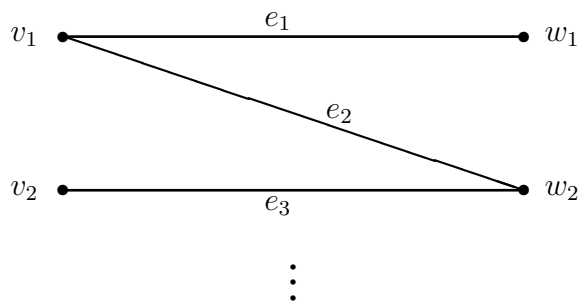


Figure 5.1: A part of a simple bipartite graph representing a Label-Cover instance

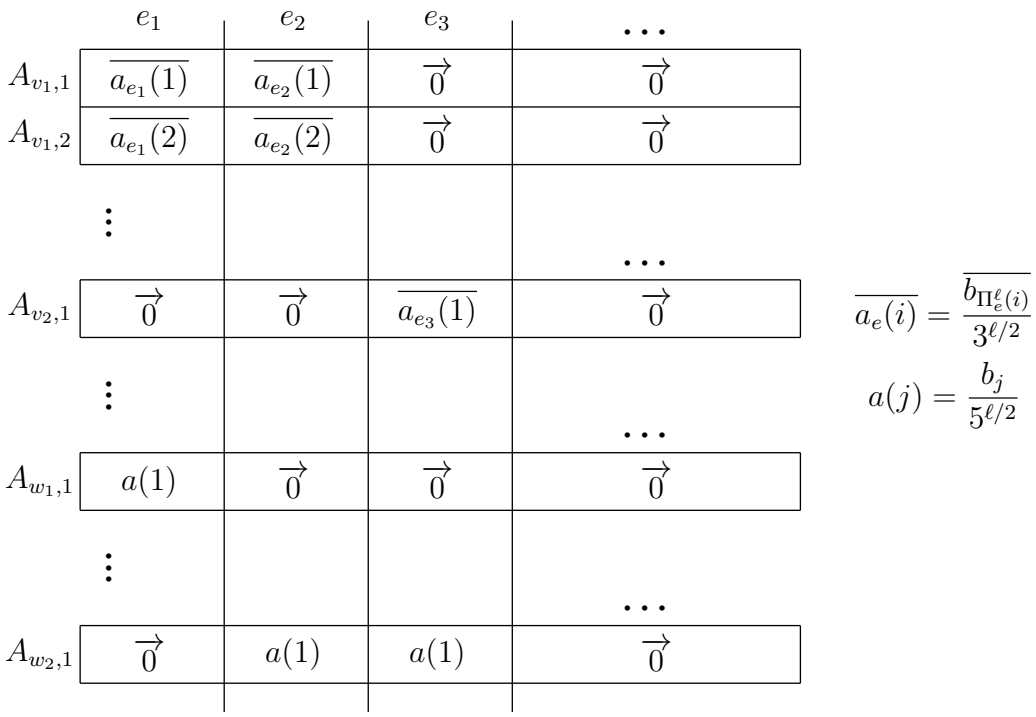


Figure 5.2: The resulting (row) vectors in MAX-VOL instance computed from the graph in Figure 5.1 by our reduction

to the subspaces corresponding to the edges incident to the vertex corresponding to the vector. A crucial observation for this construction is that, vectors A_{v_1,i_1} and A_{v_2,i_2} are orthogonal to each other for all $v_1, v_2 \in V^l$, and $i_1, i_2 \in \Sigma_V^\ell$, since there are no edges between the vertices in V^ℓ. The same result holds for the vertices in W^ℓ. From now on, this fact will be used frequently without explicit reference. We set the number of column vectors k to be chosen in the MAX-VOL instance to $|V^\ell| + |W^\ell| = (5n/3)^\ell + n^\ell$.

5.3.3 The Analysis

We start with the completeness of the reduction:

Theorem 5.7. *If the Label Cover instance L^ℓ has a labeling that satisfies all the edges, then in the MAX-VOL instance, there exists k column vectors with volume* 1.

Proof. We show that there are at least k orthogonal vectors. For an edge $e = (v, w)$, let $i \in \Sigma_V^\ell$ and $j \in \Sigma_W^\ell$ be the labeling of v and w assigned by the optimal labeling which satisfies all the edges. Then, in the MAX-VOL instance the dot product of the vectors $A_{v,i}$ and $A_{w,j}$ is

$$A_{v,i} \cdot A_{w,j} = \sum_{e \in E^\ell} A_{v,i}(e) \cdot A_{w,j}(e) = \overline{b_{\Pi_e^\ell(i)}} \cdot b_j = \overline{b_j} \cdot b_j = 0. \tag{5.1}$$

This is due to the fact that the labeling satisfies e, i.e. $b_{\Pi_e^\ell(i)} = b_j$. Since all the edges are satisfied, and there exists a vector from each vertex corresponding to the optimal labeling satisfying the equation (5.1), we have $|V^\ell| + |W^\ell|$ orthogonal vectors, i.e. we have k orthogonal vectors. □

Before proving the soundness of the reduction, which will prove hardness of approximation, we first give the intuition for the argument. According to our construction of the MAX-VOL instance, there is a set of vectors corresponding to each node in V^ℓ and W^ℓ. The set of vectors defined for a specific node has high pair-wise dot products whereas a vector from a node $v_1 \in V^\ell$ and another from v_2 in V^ℓ are orthogonal to each other. The same goes for the vectors defined for W^ℓ. Hence, if vectors are chosen from the same set corresponding to a single node, the total volume will decrease exponentially with respect to the number of such vectors. Let us call these vectors *duplicates* in V^ℓ and W^ℓ. The more intricate part of the analysis is due to the dot products *between* the vectors defined for V^ℓ and W^ℓ, which is enforced to be non-zero by the unsatisfied edges in the Label-Cover instance. We will show that, in case the Label-Cover instance has few satisfied edges, any k vectors chosen in the MAX-VOL instance should satisfy the following: either the number of duplicates in V^ℓ and W^ℓ is large enough so that the total volume is small, or the dot products between V^ℓ and W^ℓ leads to a small volume; this will prove our result.

Theorem 5.8. *There exists absolute constants α and c such that, if the Label Cover instance L^ℓ does not have any labeling that satisfies more than $2^{-\alpha\ell}$ of the edges, then the volume of any k vectors in the MAX-VOL instance is at most 2^{-ck}.*

Proof. Let $V^\ell = \{v_1, \ldots, v_{(5n/3)^\ell}\}$ and $W^\ell = \{w_1, \ldots, w_{n^\ell}\}$. Let A_v be the vectors corresponding to the vertex $v \in V^l$: $A_v = \{A_{v,i} | i \in \Sigma_V^\ell\}$. Similarly, let $A_w =$

$\{A_{w,j}|j \in \Sigma_W^\ell\}$ for $w \in W^\ell$. Let A_{V^ℓ} be the set of all vectors corresponding to the nodes in V^ℓ, and A_{W^ℓ} be the set of all vectors corresponding to the nodes in W^ℓ, i.e.

$$A_{V^\ell} = \bigcup_{i=1}^{(5n/3)^\ell} A_{v_i}, \qquad A_{W^\ell} = \bigcup_{i=1}^{n^\ell} A_{w_i}.$$

For a set of vectors C of size k, let $C_u = C \cap A_u$ for all $u \in \{V^\ell \cup W^\ell\}$, $C_{V^\ell} = C \cap A_{V^\ell}$ and $C_{W^\ell} = C \cap A_{W^\ell}$. Let $V^\ell(C)$ and $W^\ell(C)$ be the set of vectors for which C "selects" at least one vector from V^ℓ and W^ℓ, respectively.

$$V^\ell(C) = \{v \in V^\ell | C_v \neq \emptyset\}, \qquad W^\ell(C) = \{w \in W^\ell | C(A_w) \neq \emptyset\}.$$

For ease of notation, we let $k_{V_C} = |C_{V^\ell}|, k_{W_C} = |C_{W^\ell}|, d_{V_C} = k_{V_C} - |V^\ell(C)|, d_{W_C} = k_{W_C} - |W^\ell(C)|$. Note that k_{V_C} and k_{W_C} denote how many vectors are chosen by C from V^ℓ and W^ℓ, respectively. Whereas d_{V_C} and $d_{C(W)}$ are the total number of duplicates in C_{V^ℓ} and C_{W^ℓ}, respectively. The following lemma relates the number of duplicates on one side with its volume.

Lemma 5.9. *$Vol(C_{V^\ell}) \leq (\sqrt{3}/2)^{d_{V_C}}$ and $Vol(C_{W^\ell}) \leq (\sqrt{3}/2)^{d_{W_C}}$.*

Proof. Let P be the set of $|V^\ell(C)|$ elements which contains exactly one vector of the form $A_{v,i}$ for each $v \in V^\ell(C)$. In words, we consider the vectors of C corresponding to the nodes in the Label-Cover instance minus all the duplicates. For the duplicate vector $A_{v,j}$, we have $A_{v,i} \cdot A_{v,j} = 1/2$. Hence, $d(A_{v,j}, P) \leq d(A_{v,j}, A_{v,i}) = \sqrt{3}/2$. By the definition of d_{V_C} and by the Union Lemma, we get $Vol(C_{V^\ell}) \leq (\sqrt{3}/2)^{d_{V_C}}$. The argument for $Vol(C_{W^\ell})$ is similar. □

Let the constant $c = 1/(3 \cdot 5^{\ell+1})$. We will show that Theorem 5.8 holds for this value of c; we will prove that $Vol(C) \leq 2^{-ck}$ for any set C of k vectors. To this aim, we argue by contradiction. The next lemma roughly states that if the volume of C is large enough, then its vectors are almost equally distributed among the nodes of the Label-Cover instance. This condition will in turn imply a small volume completing our argument.

Claim 5.10. *If* $Vol(C) \geq 2^{-ck}$ *for* $c = 1/(3 \cdot 5^{l+1})$*, then*

$$(1-\epsilon_1)(5n/3)^\ell < k_{V_C} < (1+\epsilon_1)(5n/3)^\ell, \tag{5.2}$$

$$(1-\epsilon_2)n^\ell < k_{W_C} < (1+\epsilon_2)n^\ell, \tag{5.3}$$

where $\epsilon_1 = \frac{1}{3^{l+1}}\left((3/5)^\ell + (3/5)^{2\ell}\right)$ *and* $\epsilon_2 = \frac{1}{3^{l+1}}\left((3/5)^l + 1\right)$.

Proof. First, we note that $Vol(C) \leq Vol(C_{V^\ell})$ since all the vectors in the MAX-VOL instance have unit norm. Similarly, $Vol(C) \leq Vol(C_{W^\ell})$. Thus, by the premise of the claim, we have $Vol(C_{V^\ell}) \geq 2^{-ck}$ and $Vol(C_{W^\ell}) \geq 2^{-ck}$. By Claim 5.9, we get

$$(\sqrt{3}/2)^{d_{V_C}} = 2^{d_{V_C}(-1+\log 3/2)} = Vol(C_{V^\ell}) \geq 2^{-ck}$$

which implies $d_{V_C} \leq ck/(1-\log 3/2) < 5ck$ since $\log 3 < 1.6$. The analysis for d_{W_C} along exactly the same lines also yields $d_{W_C} < 5ck$. Noting the expressions for c and k, and following the definitions, we obtain

$$\begin{aligned} k_{V_C} = |V^\ell(C)| + d_{V_C} &< |V^\ell| + 5ck \\ &= (5n/3)^\ell + \frac{1}{3 \cdot 5^\ell}((5n/3)^\ell + n^\ell) \\ &= (1+\epsilon_1)(5n/3)^\ell. \end{aligned}$$

Similarly,

$$\begin{aligned} k_{W_C} = |W^\ell(C)| + d_{W_C} &< |W^\ell| + 5ck \\ &= n^\ell + \frac{1}{3 \cdot 5^\ell}((5n/3)^\ell + n^\ell) \\ &= (1+\epsilon_2)n^\ell \end{aligned}$$

which proves the right hand sides of (5.2) and (5.3). Noting that $k_{V_C} + k_{W_C} = k = (5n/3)^\ell + n^\ell$, we get

$$\begin{aligned} k_{V_C} = k - k_{W_C} &> (5n/3)^\ell + n^\ell - (1+\epsilon_2)n^\ell \\ &= (5n/3)^\ell - \frac{1}{3^{\ell+1}}((3n/5)^l + n^l) \\ &= (1-\epsilon_1)(5n/3)^\ell \end{aligned}$$

and

$$\begin{aligned} k_{W_C} = k - k_{V_C} &> (5n/3)^\ell + n^\ell - (1+\epsilon_1)(5n/3)^\ell \\ &= n^\ell - \frac{1}{3^{\ell+1}}((3n/5)^\ell + n^\ell) \\ &= (1-\epsilon_2)n^\ell \end{aligned}$$

which proves the left hand sides. □

Lemma 5.10 ensures that if the volume of a set of k vectors exceeds 2^{-ck}, then some certain concentration result should hold, namely Equation (5.2) and Equation (5.3). We will now show that, these equations imply $Vol(C) < 2^{-ck}$ which is our contradiction.

Without loss of generality, let $V^\ell(C)) = \{v_1, \ldots, v_q\}$, $W^\ell(C) = \{w_1, \ldots, w_p\}$. Note that these sets contain the nodes of the Label-Cover instance from which C "selects" at least one vector. Let $Q = \{A_{v_1,i_1}, \ldots, A_{v_q,i_q}\}$ where $A_{v_s,i_s} \in C_{v_s}$ for $s = 1, \ldots, q$. Let $P = \{A_{w_1,j_1}, \ldots, A_{w_p,j_p}\}$ where $A_{v_s,i_s} \in C_{v_s}$ for $s = 1, \ldots, p$. By definition,

$$q = k_{V_C} - d_{V_C} > (1-2\epsilon_1)(5n/3)^\ell, \qquad p = k_{W_C} - d_{W_C} > (1-2\epsilon_2)n^\ell.$$

In words, the set of nodes from which C selects at least one vector essentially covers V^ℓ and W^ℓ. These vectors are all orthogonal. From this point of view, $V^\ell(C))$ and $W^\ell(C))$ play an important role in our argument. Since C "covers" V^l and W^l and since the Label-Cover instance has many unsatisfied edges, it means

that the dot products of many vectors in C_{V^ℓ} with many vectors in C_{W^ℓ} will be large. This will lead to small volume. Hence, we are essentially interested in the number of unsatisfied edges between $V^\ell(C)$ and $W^\ell(C)$. Since there are at most $2^{-\alpha\ell}$ satisfied edges in the Label-Cover instance, and there are exactly 3^ℓ edges incident to a node in V^ℓ, the number of unsatisfied edges incident to $V^\ell(C)$ is greater than $(1-2\epsilon_1-2^{-\alpha\ell})(5n)^\ell$. Similarly, the number of unsatisfied edges incident to $W^\ell(C)$ is greater than $(1-2\epsilon_2-2^{-\alpha\ell})(5n)^\ell$. Thus, the number of unsatisfied edges whose end points are in $V^\ell(C)$ and $W^\ell(C)$, is greater than $(1-2\epsilon_1-2\epsilon_2-2^{-\alpha\ell+1})(5n)^\ell$.

We now give an upper bound for the distance of the vectors in Q to P, namely $\|A_{v_s,i_s}-\pi_P(A_{v_s,i_s})\|_2$ for each $A_{v_s,i_s}\in Q$. To this end, we define the set $N(A_{v_s,i_s})=\{C_w|e=(A_{v_s,i_s},w) \text{ is unsatisfied}\}$. Note that the vectors in different sets are mutually orthogonal, and by the reduction we have

$$A_{v_s,i_s}\cdot A_{w,j}=\sum_{e\in E^\ell}A_{v_s,i_s}(e)\cdot A_{w,j}(e)=\overline{b_{\Pi_e^\ell(i)}}\cdot b_j=\frac{1}{2\cdot 3^{\ell/2}\cdot 5^{\ell/2}}$$

for $A_{w,j}\in N(A_{v_s,i_s})$ since $e=(A_{v_s,i_s},A_{w,j})$ is unsatisfied. Thus, by the Pythagoras Theorem, we obtain

$$d(A_{v_s,i_s},P)=\|A_{v_s,i_s}-\pi_P(A_{v_s,i_s})\|_2<\left(1-\frac{|N(A_{v_s,i_s})|}{4\cdot 3^\ell\cdot 5^\ell}\right)^{\frac{1}{2}}.$$

Using the Union Lemma, we get

$$\begin{aligned}Vol(P\cup Q)&\leq Vol(P)\cdot\prod_{s=1}^{q}d(A_{v_s,i_s},P)\\&<Vol(P)\cdot\prod_{s=1}^{q}\left(1-\frac{|N(A_{v_s,i_s})|}{4\cdot 3^\ell\cdot 5^\ell}\right)^{\frac{1}{2}}.\end{aligned}$$

The product in the last expression is maximized when all the factors are equal to each other. We also previously showed that $\sum_{s=1}^{q}|N(A_{v_s,i_s})|>(1-2\epsilon_1-2\epsilon_2-2^{-\alpha\ell+1})(5n)^\ell$ and that q, the number of distinct nodes hit in V^ℓ satisfies, $q>(1-2\epsilon_1)(5n/3)^\ell$. Hence, we obtain

$$
\begin{aligned}
Vol(P \cup Q) &< Vol(P) \cdot \prod_{s=1}^{q} \left(1 - \frac{\sum_{s=1}^{q} |N(A_{v_s,i_s})|}{q \cdot 4 \cdot 3^{\ell} \cdot 5^{\ell}}\right)^{\frac{1}{2}} \\
&< Vol(P) \cdot \prod_{s=1}^{q} \left(1 - \frac{(1 - 2\epsilon_1 - 2\epsilon_2 - 2^{-\alpha\ell+1})(5n)^{l}}{(5n/3)^{\ell} \cdot 4 \cdot 3^{\ell} \cdot 5^{\ell}}\right)^{\frac{1}{2}} \\
&= Vol(P) \cdot \left(1 - \frac{(1 - 2\epsilon_1 - 2\epsilon_2 - 2^{-\alpha\ell+1})}{4 \cdot 5^{\ell}}\right)^{\frac{q}{2}} \\
&< Vol(P) \cdot \left(1 - \frac{(1 - 2\epsilon_1 - 2\epsilon_2 - 2^{-\alpha\ell+1})}{4 \cdot 5^{\ell}}\right)^{\frac{(1-2\epsilon_1)(5n/3)^{\ell}}{2}}.
\end{aligned}
$$

To simplify, let $t = \frac{4 \cdot 5^{\ell}}{(1-2\epsilon_1-2\epsilon_2-2^{-\alpha\ell+1})}$. For $\ell \geq 1$, we have $\epsilon_1 = \frac{1}{3^{l+1}}\left((3/5)^{\ell} + (3/5)^{2\ell}\right) \leq \frac{1}{3^2}\left((3/5) + (3/5)^2\right) < 3/20$. Noting that $\log e \geq 10/7$, we obtain $\log e(1 - 2\epsilon_1) \geq 10/7 \cdot 7/10 = 1$. Then, we have

$$
\begin{aligned}
Vol(P \cup Q) &< Vol(P) \cdot \left(1 - \frac{1}{t}\right)^{t \cdot \frac{(1-2\epsilon_1)(5n/3)^{\ell}}{2t}} \\
&\leq e^{-\frac{(1-2\epsilon_1)(5n/3)^{\ell}}{2t}} \\
&= 2^{-\log e \cdot \frac{(1-2\epsilon_1)(5n/3)^{\ell}}{2t}} \\
&\leq 2^{-\frac{(5n/3)^{\ell}}{2t}},
\end{aligned}
$$

where e is the base of the natural logarithm. For $\ell \geq 2$, we have $2\epsilon_1 < 1/27$ and $2\epsilon_2 < 3/27$. Let ℓ' be the smallest integer such that $2^{-\alpha\ell+1} < 11/27$. Then, for $\ell \geq \ell'$, we get $t = \frac{4 \cdot 5^{\ell}}{(1-2\epsilon_1-2\epsilon_2-2^{-\alpha\ell+1})} < \frac{4 \cdot 5^{\ell}}{(4/9)} = 9 \cdot 5^{\ell}$. Since $n^{\ell} = k/(1+(5n/3)^{\ell}) > k/(5/3)^{\ell+1}$, we finally obtain

$$
Vol(P \cup Q) < 2^{-\frac{(5n/3)^{\ell}}{9 \cdot 5^{\ell}}} = 2^{-\frac{n^{\ell}}{3^{\ell+2}}} < 2^{-\frac{k}{3 \cdot 5^{\ell+1}}} = 2^{-ck},
$$

which is our contradiction. Thus, the volume of a set of k vectors in a negative instance of MAX-VOL cannot exceed 2^{-ck} for $c = \frac{1}{3 \cdot 5^{\ell+1}}$. Combining this with Fact 5.1, we have shown that MAX-VOL does not admit a 2^{-ck} approximation algorithm, assuming $P \neq NP$. □

We have shown that

- if the optimal value of the Label Cover instance is 1, then the optimal value of the MAX-VOL instance is 1
- if the optimal value of the ℓ-fold Label Cover instance is less than $2^{-\alpha\ell}$, then the optimal value of the MAX-VOL instance is less than 2^{-ck}.

We know that there exists a gap producing reduction from SAT to ℓ-fold Label Cover by the combination of Theorem 5.3 and Theorem 5.5 with parameters 1 and $2^{-\alpha\ell}$. This means that there is a polynomial time reduction from a SAT to MAX-VOL such that, given a formula ϕ

- if ϕ is satisfiable , then $OPT(\text{MAX-VOL}) = 1$.
- if ϕ is not satisfiable , then $OPT(\text{MAX-VOL}) = 2^{-ck}$.

Now, assume that there exists an algorithm for MAX-VOL with approximation ratio greater than or equal to 2^{-ck}. Then,

- if ϕ is satisfiable , the algorithm returns a subset with volume at least 2^{-ck}.
- if ϕ is not satisfiable, then the volume of the subset returned by the algorithm is smaller than 2^{-ck}.

which means that we can solve SAT in polynomial time. Thus, unless $P = NP$, MAX-VOL is inapproximable to within 2^{-ck} for some $c > 0$.

5.4 Discussion

Our reduction heavily relies on the Raz' Parallel Repetition Theorem [52]. It is not possible to get an exponential inapproximability result without parallel repetition. But, since the degrees of the vertices in the Label-Cover instance exponentially increases with respect to the number of repetitions, our constant c depends on the constant α in Raz' result. It might be possible to improve this constant by making use of more sophisticated parallel repetition theorems, but we did not proceed so

far. Indeed, the exact analysis is irrelevant as the constant will be too small in all cases. Overall, the strength of our result is is directly related to the underlying theorems for the inapproximability of Label-Cover.

Another way of getting a stronger hardness result is to find a more sophisticated reduction. In our MAX-VOL instance, the subspaces "reserved" for each edge in the Label-Cover instance are orthogonal to each other. This dramatically simplifies the analysis, yielding perfect completeness, i.e. volume 1 in MAX-VOL. It might be possible to construct a MAX-VOL instance for which these subspaces have some pair-wise angle, so that we sacrifice the perfect completeness, but at the same time get a much smaller soundness. This would improve the inapproximability result.

The obvious open problem is whether the inapproximability can be strengthened to 2^{-k+1}. Recall that this is the lower bound for the greedy algorithm for MAX-VOL. It seems that with the techniques we have used, it is impossible to break the dependence on the constant in the parallel repetition theorems.

CHAPTER 6
Deterministic Low-Rank Approximation

6.1 Introduction

In this chapter, we give a deterministic greedy algorithm for the low-rank matrix approximation problem which is based on the sparse approximation of the SVD of A. We first generalize the problem of sparse approximation in [47] to one of approximating a subspace, using the columns from A. We then propose and analyze a greedy algorithm for this problem and get our main result in the special case where the subspace to be approximated is the best rank-k approximation to A.

In words, our algorithm first computes the top k left singular vectors of A, and then selects columns of A in a greedy fashion so as to "fit" the space spanned by the singular vectors, appropriately scaled according to the singular values. The performance characteristics of the algorithm depend on how well the greedy algorithm approximates the optimal choice of such columns from A, and on how good the optimal columns themselves are. We combine an existence result on the quality of the optimal columns with the analysis of the greedy algorithm to arrive at the following result:

Theorem 6.1. *For $\epsilon \leq \frac{\sqrt{k}\|A_k\|_F}{e\cdot\|A-A_k\|_F}$, the greedy algorithm chooses a column sub-matrix $C \subseteq A$ with $c = O\left(\frac{k^2 \log k}{\epsilon^2}\mu^2(A) \ln\left(\frac{\sqrt{k}\|A_k\|_F}{\epsilon\|A-A_k\|_F}\right)\right)$ columns such that*

$$\|A - CC^+A\|_F \leq (1+\epsilon)\|A - A_k\|_F.$$

For $\epsilon > \frac{\sqrt{k}\|A_k\|_F}{e\cdot\|A-A_k\|_F}$, the greedy algorithm chooses a column sub-matrix $C \subseteq A$ with $c = O\left(\frac{k\log k\|A-A_k\|_F^2}{\|A_k\|_F^2}\mu^2(A)\right)$ columns such that

$$\|A - CC^+A\|_F \leq (1+\epsilon)\|A - A_k\|_F.$$

In the first case (for small values of ϵ), the term $\frac{k^2 \log k}{\epsilon^2}$ arises from an upper bound on the number of columns the optimal solution would choose (the existence

result), and the remaining terms are contributed by the analysis of the greedy algorithm. The coherence parameter $\mu(A)$, restricts the class of matrices for which the bound is useful. Note that, in order to achieve this approximation ratio, we choose more than k columns, which is reminiscent of the results provided by the theoretical computer science community. The second case in the theorem makes sure that the logarithmic expression does not evaluate to a negative value.

The term $\mu(A)$, which is related to the smallest singular value of a certain submatrix of A, arises from the sparse approximation problem. We believe that a result without the parameter $\mu(A)$ should be possible, however we have not been able to construct one. Improving either the upper bound on the optimal reconstruction of the singular vectors, or improving the analysis of the greedy algorithm would yield a tighter result.

6.1.1 Preliminaries and Notation

For this chapter, we define the maximum column norm of a matrix A, $\|A\|_{col} = \max_{i=1}^{n}\{\|A^{(i)}\|_2\}$. We also define $S^{\perp}$, the space orthogonal to the space spanned by the vectors in S.

6.2 Generalized Sparse Approximation

Instead of seeking sparse approximation to a single vector [47], we propose the following generalization: given a matrix $A \in \mathbb{R}^{m\times n}$, a set of vectors $B \in \mathbb{R}^{m\times k}$, and $\epsilon > 0$, find a matrix $X \in \mathbb{R}^{n\times k}$ satisfying

$$\|AX - B\|_F \leq \epsilon, \tag{6.1}$$

such that $\sum_{i=1}^{n} \nu_i(X)$ is minimum over all possible choices of X, where $\nu_i(X) = 1$ if the row $X_{(i)}$ contains non-zero entries, $\nu_i(X) = 0$ if $X_{(i)} = \overrightarrow{0}$. Intuitively, the problem asks for a minimum number of column vectors of A whose span is "close" to the span of B.

6.2.1 The Algorithm

A greedy strategy for solving this problem is to choose the column v from A at each iteration, for which $\|B^T v\|_2$ is maximum, and project the column vectors of B and the other column vectors of A onto the space orthogonal to the chosen column. The algorithm proceeds greedily on these residual matrices until the norm of the residual B drops below the required threshold ϵ. Naturally, if the error ϵ cannot be attained, the algorithm will fail after selecting a maximal independent set of columns.

Algorithm 6 Greedy

1: normalize each column of A to have norm 1.
2: $l \leftarrow 0$, $\Lambda \leftarrow \emptyset$, $A_0 \leftarrow A$, $B_0 \leftarrow B$.
3: **while** $\|B_l\|_F > \epsilon$ **do**
4: choose $i \in \{1, \ldots, n\} - \Lambda$ such that $\|B_l^T A_l^{(i)}\|_2$ is maximum
5: $B_{l+1}^{(j)} \leftarrow B_l^{(j)} - \left(B_l^{(j)^T} A_l^{(i)}\right) A_l^{(i)}$ for $i = 1, \ldots, k$, i.e. project $B_l^{(j)}$'s onto $\{A_l^{(i)}\}^{\perp}$.
6: $\Lambda \leftarrow \Lambda \cup \{i\}$.
7: $A_{l+1}^{(j)} \leftarrow A_l^{(j)} - \left(A_l^{(j)^T} A_l^{(i)}\right) A_l^{(i)}$ for $j \in \{1, \ldots, n\} - \Lambda$, i.e. project $A_l^{(j)}$'s onto $\{A_l^{(i)}\}^{\perp}$.
8: normalize $A_{l+1}^{(j)}$ for $j \in \{1, \ldots, n\} - \Lambda$.
9: $l \leftarrow l + 1$.
10: **end while**
11: return $C = \Lambda(A)$, the selected columns.

6.2.2 Implementation Details and Running Time Analysis

Line 1,7 and 8 of Greedy takes $O(mn)$ time. Line 5 takes $O(mk)$ time since $B \in \mathbb{R}^{m \times k}$. The computationally intensive part of the algorithm in the while loop is the 4th step, which takes $O(mnk)$ time with a naive implementation, since there are n matrix-vector multiplications of cost $O(mk)$. This makes a total of $O(mnkc)$ running time complexity which amounts to $O(mnk^2)$ in case k vectors are chosen. We make note of a simple observation which is akin to the pivoted QR algorithms and is called a norm update: instead of performing matrix-vector multiplications at each iteration, we remember the dot products of the chosen column with the columns of B and the other columns in A. We also introduce a matrix $D \in \mathbb{R}^{k \times n}$, where $(D_l)_{ij}$ denotes the the dot product of the i^{th} column of B and the j^{th} column

of A in the l^{th} iteration, i.e. $(D_l)_{ij} = {B_l^{(i)}}^T A_l^{(j)}$. At the end of each iteration, we update D_l to get D_{l+1} where the update of each entry requires constant time. Hence, the 4th step takes $O(nk)$ time complexity for each iteration. Overall, the running time of the algorithm is $O((2mn + mk + nk)c) = O(mnc)$.

The norm update is as follows: Suppose a column vector v is chosen at iteration l. Noting that $v^T v = 1$, D_{l+1} satisfies

$$
\begin{aligned}
(D_{l+1})_{ij} = {B_{l+1}^{(i)}}^T A_{l+1}^{(j)} &= \frac{\left(B_l^{(i)} - \left({B_l^{(i)}}^T v\right) v\right)^T \left(A_l^{(j)} - \left({A_l^{(j)}}^T v\right) v\right)}{\left\| A_l^{(j)} - \left({A_l^{(j)}}^T v\right) v \right\|_2} \\
&= \frac{{B_l^{(i)}}^T A_l^{(j)} + \left({B_l^{(i)}}^T v\right)\left({A_l^{(j)}}^T v\right)\left(v^T v - 2\right)}{\left\| A_l^{(j)} - \left({A_l^{(j)}}^T v\right) v \right\|_2} \\
&= \frac{(D_l)_{ij} - \left({B_l^{(i)}}^T v\right)\left({A_l^{(j)}}^T v\right)}{\left\| A_l^{(j)} - \left({A_l^{(j)}}^T v\right) v \right\|_2}.
\end{aligned}
$$

The update per entry can be performed in constant time given the other values in the last expression, which are already computed.

6.2.3 Performance Analysis

Our analysis yields an approximation factor which includes a term related to the smallest singular value of a certain sub-matrix, which is relevant to the analysis. We begin with the following definition, which provides a general upper bound for the spectral norm of the pseudo-inverse of any sub-matrix of a matrix. We would like to note that similar definitions have appeared in [59] while analyzing algorithms for the sparse approximation problem.

Definition 6.2. *[**Coherence**] Given a matrix $A \in \mathbb{R}^{m \times n}$ of rank r, let $\mathbf{A}$ be the matrix A with normalized columns. Then, $\mu(A)$ is the maximum of the inverse of the least singular value over all $m \times r$ full-rank sub-matrices of $\mathbf{A}$. Namely,*

$$\mu(A) = \max_{\substack{\mathbf{C} \subseteq \mathbf{A} \\ \mathbf{C} \in \mathbb{R}^{m \times r} \\ rank(\mathbf{C}) = r}} \frac{1}{\sigma_r(\mathbf{C})}. \tag{6.2}$$

Remark 6.3. $1 \leq \mu(A) < \infty$. *Small values of* $\mu(A)$ *indicate a matrix with near orthonormal columns.*

Theorem 6.4. *The number of columns chosen by Greedy is at most*

$$O\left(Opt(\epsilon/2)\mu^2(A) \ln\left(\frac{\|B\|_F}{\epsilon}\right)\right),$$

where $Opt(\epsilon/2)$ *is the optimal number of columns at error* $\epsilon/2$.

We will establish Theorem 6.4 through a sequence of lemmas. The proof follows similar reasoning to the proof in [47]. Let t be the total number of iterations of Greedy. At the beginning of the l^{th} iteration of the algorithm, for $0 \leq l < t$, let U_l be an optimal solution to the generalized sparse approximation problem with error parameter $\epsilon/2$, i.e. U_l minimizes $\sum_{i=1}^{n} \nu_i(X)$ over $X \in \mathbb{R}^{n \times k}$ such that $\|A_l U_l - B_l\|_F \leq \epsilon/2$, where $\nu_i(X) = 1$ if the row $X_{(i)}$ contains non-zero entries, $\nu_i(X) = 0$ if $X_{(i)} = \overrightarrow{0}$. Let $N_l = \sum_{i=1}^{n} \nu_i(U_l)$ and $Q_l = A_l U_l$. Define

$$\lambda = 4 \max_{0 \leq l < t} \frac{N_l \|U_l\|_F^2}{\|B_l\|_F^2}. \tag{6.3}$$

Assuming that the Greedy has not terminated, the following lemma states that the next step makes significant progress.

Lemma 6.5. *For the* l^{th} *iteration of Greedy,* $\|B_l^T A_l\|_{col} \geq \frac{\|B_l\|_F^2}{2\sqrt{N_l}\|U_l\|_F}$.

Proof. Let $E \in \mathbb{R}^{m \times k}$ be a generic error matrix such that $\|E\|_F \leq \epsilon/2$, and Let $\|E^{(j)}\|_2 = \epsilon_j/2$ for $i = 1, \ldots, k$. Hence, $\sum_{i=1}^{k} \epsilon_j^2 \leq \epsilon^2$. Now, we can write $B_l^{(j)} = \left(\sum_{i=1}^{n} A_l^{(i)} U_{lij}\right) + E^{(j)}$ for $j = 1, \ldots, k$. Then,

$$\|B_l\|_F^2 = \sum_{j=1}^{k} {B_l^{(j)}}^T B_l^{(j)} = \sum_{j=1}^{k} \sum_{i=1}^{n} U_{lij} {B_l^{(j)}}^T A_l^{(i)} + \sum_{j=1}^{k} {B_l^{(j)}}^T E^{(j)}. \tag{6.4}$$

We will first bound the double summation in the above expression.

$$\begin{aligned}\sum_{j=1}^{k}\sum_{i=1}^{n} U_{lij} B_l^{(j)^T} A_l^{(i)} \leq & \sum_{i=1}^{n}\left(\left(\sum_{j=1}^{k} U_{lij}^2\right)^{1/2}\left(\sum_{j=1}^{k}\left(B_l^{(j)^T} A_l^{(i)}\right)^2\right)^{1/2}\right) \\ & \leq \max_{1\leq i\leq n}\left\{\left(\sum_{j=1}^{k}\left(B_l^{(j)^T} A_l^{(i)}\right)^2\right)^{1/2}\right\}\sum_{i=1}^{n}\left(\sum_{j=1}^{k} U_{lij}^2\right)^{1/2} \\ & \leq \|B_l^T A_l\|_{col}\sqrt{N_l}\|U_l\|_F.\end{aligned}$$

The first line is due to Cauchy-Schwartz inequality. The last inequality bounds the double summation in the second line as follows. Define n dimensional vectors a and b such that $a_i = \left(\sum_{j=1}^{k} U_{lij}^2\right)^{1/2}$ and $b_i = 1$ if there exists a non-zero entry in the i^{th} row of U_l, and $b_i = 0$ if all the elements in the i^{th} row of U_l are zero, for $i = 1, \ldots, n$. Then, applying Cauchy-Schwartz inequality to a and b, we obtain $\sum_{i=1}^{n}\left(\sum_{j=1}^{k} U_{lij}^2\right)^{1/2} = \sum_{i=1}^{n} a_i b_i \leq \left(\sum_{i=1}^{n} a_i^2\right)^{1/2}\left(\sum_{i=1}^{n} b_i^2\right)^{1/2}$. Since $\sum_{i=1}^{n} a_i^2 = \sum_{i=1}^{n}\sum_{j=1}^{k} U_{lij}^2 = \|U_l\|_F{}^2$, and $\sum_{i=1}^{n} b_i^2 = N_l$, we have that $\sum_{i=1}^{n}\left(\sum_{j=1}^{k} U_{lij}^2\right)^{\frac{1}{2}} \leq \sqrt{N_l}\|U_l\|_F$.

We will now bound the second term in (6.4).

$$\begin{aligned}\sum_{j=1}^{k} B_l^{(j)^T} E^{(j)} &\leq \sum_{j=1}^{k} \|B_l^{(j)^T}\|_2 \|E^{(j)}\|_2 \quad (Cauchy - Schwartz) \\ &= \frac{1}{2}\sum_{j=1}^{k} \epsilon_j \|B_l^{(j)^T}\|_2 \\ &\leq \frac{1}{2}\left(\sum_{j=1}^{k}\epsilon_j^2\right)^{1/2}\left(\sum_{j=1}^{k}\|B_l^{(j)^T}\|_2^2\right)^{1/2} \quad (Cauchy - Scwartz) \\ &\leq \frac{1}{2}\epsilon\|B_l\|_F \\ &\leq \frac{1}{2}\|B_l\|_F^2,\end{aligned}$$

where the last inequality is due to the fact that $\|B_l\|_F > \epsilon$, i.e. the algorithm is still running.

Combining these bounds in (6.4), we have $\|B_l\|_F^2 \leq \|B_l^T A_l\|_{col}\sqrt{N_l}\|U_l\|_F + 1/2\|B_l\|_F^2$, which gives $\|B_l\|_F^2 \leq 2\|B_l^T A_l\|_{col}\sqrt{N_l}\|U_l\|_F$. The lemma then immedi-

ately follows. □

Thus, there exists a column in the residual A_l which will reduce the residual B_l significantly, because B_l has a large projection onto this column. Therefore, since every step of Greedy makes significant progress, there cannot be too many steps, which is the content of the next lemma.

Lemma 6.6. *$t \leq \left\lceil 2\lambda \ln\left(\frac{\|B\|_F}{\epsilon}\right)\right\rceil$, where t is the number of Greedy iterations.*

Proof. Let i be the index of the chosen column at step l and let j be a column index of B. Then, by the execution of the algorithm, $B_{l+1}^{(j)} = B_l^{(j)} - \left(B_l^{(j)^T} A_l^{(i)}\right) A_l^{(i)}$. Since $B_{l+1}^{(j)}$ is orthogonal to $A_l^{(i)}$ and $\|A_l^{(i)}\|_2 = 1$, we can write $\|B_{l+1}^{(j)}\|_2^2 = \|B_l^{(j)}\|_2^2 - |B_l^{(j)^T} A_l^{(i)}|^2$. Summing over all column indices of B_{l+1}, we obtain

$$\begin{aligned}
\|B_{l+1}\|_F^2 = \sum_{j=1}^{k} \|B_{l+1}^{(j)}\|_2^2 &= \sum_{j=1}^{k} \|B_l^{(j)}\|_2^2 - \sum_{j=1}^{k} |B_l^{(j)^T} A_l^{(i)}|^2 \\
&= \|B_l\|_F^2 - \|B_l^T A_l^{(i)}\|_2^2 \\
&= \|B_l\|_F^2 - \|B_l^T A_l\|_{col}^2 \\
&\leq \|B_l\|_F^2 - \frac{\|B_l\|_F^4}{4N_l \|U_l\|_F^2} \quad (Lemma\,6.5) \\
&= \|B_l\|_F^2 \left(1 - \frac{1}{\lambda}\right) \quad (Equation\,(6.3)),
\end{aligned}$$

where the third line follows since the algorithm chooses i to maximize $\|B_l^T A_l^{(i)}\|_2$. Hence, $\|B_l\|_F^2 \leq (1 - 1/\lambda)\|B_0\|_F^2$. Since the algorithm stops when $\|B_t\|_F^2 \leq \epsilon^2$, it suffices for t to satisy $(1 - 1/\lambda)^t \|B_0\|_F^2 \leq \epsilon^2$. Rearranging, and taking logarithms we obtain $t \ln(1 - 1/\lambda) \leq \ln(\epsilon^2/\|B_0\|_F^2)$. Since $\ln(1 - 1/\lambda) \leq -1/\lambda$, we get that $t \geq \lambda \ln(\|B\|_F^2/\epsilon^2) = 2\lambda \ln(\|B\|_F/\epsilon)$ iterations are enough for Greedy to terminate. □

What remains is to bound λ. First, we will bound $\|U_l\|_F$ in terms of $\|B_l\|_F$, both of which appear in the expression for λ. Let $\pi_l = \{i | U_{l(i)} \neq \overrightarrow{0}\}$ be the indices of rows of U_l which are not all zero. Recall that these indices denote which columns are chosen by the optimal solution for A_l. Let $\tau_l = \{i_1, i_2, \ldots, i_l\}$ be the indices of

the first l columns picked by the algorithm. Given an index set γ, let the set of column vectors $\{A^{(i)}|i \in \gamma\}$ be denoted by $\gamma(A)$.

Lemma 6.7. *$\pi_l(A) \cup \tau_l(A)$ is a linearly independent set for all $l \geq 0$.*

Proof. Note that for $l = 0$, we only have $\sigma_0(A)$ and by the definition of the optimality of U_0, this set should be linearly independent. For $l \geq 1$, we will argue by contradiction. Assume that the given set, $\pi_l(A) \cup \tau_l(A)$ is not a linearly independent set. Hence, some linear combination of some vectors from the set sum to 0. Since, by the execution of the algorithm, $\tau_l(A)$ is a linearly independent set, at least one of these vectors should be from $\pi_l(A)$, and this vector u can be written as a linear combination of some other vectors in $\pi_l(A) \cup \tau_l(A)$. To this end, recall that π_l denotes the indices of columns of A_l chosen by the optimal solution U_l, and $\pi_l(A)$ is the set of columns of A with these indices. Consider a column vector v in $\pi_l(A)$. According to the algorithm, at the end of the l^{th} iteration, the residual vector v_l (which is in $\pi_l(A_l)$) is precisely the projection of v onto the space orthogonal to the vectors chosen by the algorithm, namely $\tau_l(A)$. Since this is the case for all possible v's, we have that $\pi_l(A_l)$ is the projection of $\pi_l(A)$ onto the space orthogonal to $\tau_l(A)$. Hence, according to our last assumption, u_l which is the projection of u onto the space orthogonal to $\tau_l(A)$ can be expressed as a linear combination of some other vectors in $\pi_l(A_l)$ since no vector from $\tau_l(A)$ can contribute in the expansion of u_l. This contradicts the optimality of U_l, i.e. that the number of columns it "selects" from A_l is the fewest among all possible choices. □

Lemma 6.8. *For $0 \leq l < t$, $\|U_l\|_F \leq \frac{3}{2}\mu(A)\|B_l\|_F$.*

Proof. Consider the column indices $\{i_1, i_2, \ldots, i_l\}$ of the first l vectors chosen by the algorithm. Specifically, let $\tau_l(A_l) = \{A_l^{(i_1)}, A_l^{(i_2)}, \ldots, A_l^{(i_l)}\}$ be the columns in A_l chosen by the algorithm in the order selected. Note that these vectors are orthogonal due to the algorithm. At the end of the l^{th} iteration of the algorithm, for $i \in \pi_l$, we can write

$$A_l^{(i)} = \frac{A_{l-1}^{(i)} - v_l^{(i)}}{\sqrt{1 - \|v_l^{(i)}\|_2^2}}, \tag{6.5}$$

where $v_l^{(i)}$ is in the span of $A_l^{(i_l)}$. Similarly, we can express $A_{l-1}^{(i)}$ in terms of $A_{l-2}^{(i)}$, i.e.

$$A_{l-1}^{(i)} = \frac{A_{l-2}^{(i)} - v_{l-1}^{(i)}}{\sqrt{1 - \|v_{l-1}^{(i)}\|_2^2}},$$

where $v_{l-1}^{(i)}$ is in the span of $A_l^{(i_{l-1})}$. Note that, since the vectors in $\tau_l(A_l)$ are orthogonal, we have $\|v_l^{(i)} + v_{l-1}^{(i)}\|_2^2 = \|v_l^{(i)}\|_2^2 + \|v_{l-1}^{(i)}\|_2^2$. Using this, we can recursively express $A_l^{(i)}$ in (6.5) as

$$A_l^{(i)} = \frac{A^{(i)} - v^{(i)}}{\sqrt{1 - \|v^{(i)}\|_2^2}}, \tag{6.6}$$

for some $v^{(i)} \in span(\tau_l(A))$. (Note that $span(\tau_l(A_l) = span(\tau_l(A_0) = span(\tau_l(A)$ and the columns of A are normalized). Thus, noting that $Q_l^{(j)} = \sum_{i \in \pi_l} A_l^{(i)} U_{lij}$, and $v^{(i)}$ can be expressed as a linear combination of the column vectors of $\tau_l(A)$, we have

$$Q_l^{(j)} = \sum_{i \in \pi_l} U_{lij} \frac{A^{(i)} - v^{(i)}}{\sqrt{1 - \|v^{(i)}\|_2^2}} = \sum_{i \in \pi_l} \frac{U_{lij}}{\sqrt{1 - \|v^{(i)}\|_2^2}} A^{(i)} + \sum_{i \in \tau_l} \delta_i A^{(i)}, \tag{6.7}$$

where δ_i's are appropriate coefficients in the expansion of $v^{(i)}$. Now, let S_l be the matrix of the columns from $\pi_l(A) \cup \tau_l(A)$. Note that, S_l is a column sub-matrix of A which has full rank by Lemma 6.7. Since S_l is a linearly independent set, Q_l has a unique expansion in the basis S_l given by $W_l = S_l^+ Q_l$. Specifically, for $i \in \pi_l$, $W_{lij} = U_{lij}/\sqrt{1 - \|v^{(i)}\|_2^2}$, and for $i \in \tau_l$, $W_{lij} = \delta_i$. Since $\sqrt{1 - \|v^{(i)}\|_2^2} < 1$, $|U_{lij}| \leq |W_{lij}|$ for $i \in \pi_l$. For $i \in \tau_l$, we have $U_{lij} = 0$ and hence trivially $|U_{lij}| \leq |W_{lij}|$. Applying this inequality to the j^{th} column of U_l, we obtain $\|U_l^{(j)}\|_2 \leq \|W_l^{(j)}\|_2 \leq \|S_l^+\|_2 \|Q_l^{(j)}\|_2$. The last inequality is due to sub-multiplicativity of the spectral norm. Noting that $Q_l^{(j)} = B_l^{(j)} + E^{(j)}$, where E is a generic error matrix with $\|E\|_F \leq \epsilon/2$, we obtain

$$\begin{aligned}
\|U_l\|_F^2 &= \sum_{j=1}^{k} \|U_l^{(j)}\|_2^2 \\
&\leq \|S_l^+\|_2^2 \sum_{j=1}^{k} \|Q_l^{(j)}\|_2^2 \\
&\leq \|S_l^+\|_2^2 \sum_{j=1}^{k} \left(\|B_l^{(j)} + E^{(j)}\|_2^2 \right) \\
&\leq \|S_l^+\|_2^2 \sum_{j=1}^{k} \left(\|B_l^{(j)}\|_2 + \|E^{(j)}\|_2 \right)^2,
\end{aligned}$$

where the last step is due to the triangle inequality. We continue by expanding the last expression and note that $\|E\|_F \leq \epsilon/2 = \sum_{j=1}^{k} \|E^{(j)}\|_2^2 \leq \epsilon^2/4$:

$$\begin{aligned}
\|U_l\|_F^2 &\leq \|S_l^+\|_2^2 \sum_{j=1}^{k} \left(\|B_l^{(j)}\|_2 + \|E^{(j)}\|_2 \right)^2 \\
&= \|S_l^+\|_2^2 \left(\sum_{j=1}^{k} \|B_l^{(j)}\|_2^2 + \sum_{j=1}^{k} \|E^{(j)}\|_2^2 + 2 \sum_{j=1}^{k} \|B_l^{(j)}\|_2 \|E^{(j)}\|_2 \right) \\
&\leq \|S_l^+\|_2^2 \left(\|B_l\|_F^2 + \frac{\epsilon^2}{4} + 2 \sum_{j=1}^{k} \|B_l^{(j)}\|_2 \|E^{(j)}\|_2 \right) \\
&\leq \|S_l^+\|_2^2 \left(\frac{5}{4} \|B_l\|_F^2 + 2 \sum_{j=1}^{k} \|B_l^{(j)}\|_2 \|E^{(j)}\|_2 \right) \quad (\|B_l\|_F > \epsilon).
\end{aligned}$$

Applying the Cauchy-Schwartz inequality to the second term in the parentheses, we obtain

$$\begin{aligned}
\|U_l\|_F^2 &\leq \|S_l^+\|_2^2 \left(\frac{5}{4}\|B_l\|_F^2 + 2 \left(\sum_{j=1}^{k} \|B_l^{(j)}\|_2^2 \right)^{1/2} \left(\sum_{j=1}^{k} \|E^{(j)}\|_2^2 \right)^{1/2} \right) \\
&= \|S_l^+\|_2^2 \left(\frac{5}{4}\|B_l\|_F^2 + 2\|B_l\|_F \|E\|_F \right) \\
&\leq \|S_l^+\|_2^2 \left(\frac{5}{4}\|B_l\|_F^2 + \epsilon\|B_l\|_F \right) \quad (\|E\|_F \leq \epsilon/2) \\
&\leq \|S_l^+\|_2^2 \left(\frac{5}{4}\|B_l\|_F^2 + \|B_l\|_F^2 \right) \quad (\|B_l\|_F > \epsilon) \\
&= \frac{9}{4}\|S_l^+\|_2^2 \|B_l\|_F^2.
\end{aligned}$$

Hence, we have $\|U_l\|_F \leq \frac{3}{2}\|S_l^+\|_2 \|B_l\|_F$. Now, note that the rank of S_l is less than or equal to r, the rank of A. S_l can be obtained by deleting columns of a full-rank sub-matrix Z of A, which has exactly r columns. $\|S_l^+\|_2$, which is the inverse of the least singular value of S_l is smaller than that of such a matrix Z (see [33]). Then, by the definition of $\mu(A)$, we clearly have $\|S_l^+\|_2 \leq \|Z^+\|_2 \leq \mu(A)$ and the lemma follows. □

We can now prove the main theorem.

Proof of Theorem 6.4: First, we note that the number of non-zero rows in the optimal solution is non-increasing as the algorithm proceeds, that is $N_l \geq N_{l+1}$ for $l > 0$, which follows from an argument identical to the proof of Lemma 3 in [47]. Since $Opt(\epsilon/2) = N_0$, we have

$$\lambda \leq 4 \max_{0 \leq l < t} \frac{N_0 \|U_l\|_F^2}{\|B_l\|_F^2} \leq 9 Opt(\epsilon/2)\mu^2(A),$$

where the last inequality is due to the result of Lemma 6.8. Combining this with Lemma 6.6, we have that the number of iterations of the algorithm is bounded by

$$t \leq \left\lceil 18 Opt(\epsilon/2)\mu^2(A) \ln \left(\frac{\|B\|_F}{\epsilon} \right) \right\rceil.$$

□

6.3 Deterministic Low-Rank Matrix Approximation

In this section, we give a deterministic algorithm for low-rank matrix approximation based on the greedy approach that we have introduced and analyzed for the generalized sparse approximation problem.

Algorithm 7 The low-rank approximation algorithm

1: compute U_k and Σ_k of A
2: return Greedy(A, $U_k\Sigma_k$, $\epsilon\|A - A_k\|_F$)

The algorithm first computes U_k, the top k left singular vectors of A and Σ_k the first k singular values of A, which can be performed by standard methods like Lanczos. The columns of A are then selected in a greedy fashion so as to "fit" them to the subspace spanned by the columns of $U_k\Sigma_k$. Intuitively, we select columns of A which are close to the columns of $U_k\Sigma_k$ and our analysis will show that the sub-matrix C of A we obtain is provably close to the "best" rank-k approximation to A. The error parameter which is given as an input to Greedy is $\epsilon\|A - A_k\|_F$. The following result provides an upper bound on the number of columns of the optimal solution at error $\epsilon\|A - A_k\|_F/2$.

Lemma 6.9. *There exists a column sub-matrix C of A with $c = O(k \log k/\epsilon^2)$ columns such that $\|U_k\Sigma_k - CC^+U_k\Sigma_k\|_F \leq \epsilon\|A - A_k\|_F/2$.*

Proof. We will make use of the following result which is proved in [27]. They give a randomized algorithm which constructs, with non-zero probability a set of columns with a particular approximation property which immediately translates to an existence result. For a set of columns $C \in A$, denote the sampling matrix which selects the columns by S; so $C = AS$. Let V_k be the matrix of the first k right singular vectors of A. Let V_{r-k} be the matrix containing the last $r - k$ right singular vectors of A, and let Σ_k and Σ_{r-k} be the diagonal matrices containing the first k and the last $r - k$ singular values of A.

Lemma 6.10 ([27])**.** *There exists a set of $c = O(k \log k/\epsilon^2)$ columns from A and corresponding sampling matrix S, with $C = AS$ such that $rank(V_k^T S) = rank(V_k)$, $\|\Sigma_{r-k}V_{r-k}^T S(V_k^T S)^+\|_F \leq \epsilon\|A - A_k\|_F$ where Σ_{r-k} is the diagonal matrix containing*

the smallest $r-k$ singular values of A, and V_{r-k} is the matrix containing the last $r-k$ right singular vectors of A.

To proceed with the proof of the main lemma, let $C = AS$ be the column sub-matrix whose existence is guaranteed by the theorem above. We have

$$\begin{aligned}\epsilon^2\|A - A_k\|_F^2 &\geq \|\Sigma_{r-k}V_{r-k}^T S(V_k^T S)^+\|_F^2 \\ &= \|\Sigma_k - \Sigma_k V_k^T S(V_k^T S)^+\|_F^2 + \|\Sigma_{r-k}V_{r-k}^T S(V_k^T S)^+\|_F^2,\end{aligned}$$

where the first term in the last expression is just 0 as $V_k^T S(V_k^T S)^+ = I_k$. Combining the last two terms into one expression, we have

$$\begin{aligned}\epsilon^2\|A - A_k\|_F^2 &\geq \left\| \begin{pmatrix} \Sigma_k \\ 0 \end{pmatrix} - \begin{pmatrix} \Sigma_k V_k^T \\ \Sigma_{r-k}V_{r-k}^T \end{pmatrix} S(V_k^T S)^+ \right\|_F^2 \\ &= \left\| \begin{pmatrix} \Sigma_k \\ 0 \end{pmatrix} - \begin{pmatrix} \Sigma_k & 0 \\ 0 & \Sigma_{r-k} \end{pmatrix} \begin{pmatrix} V_k^T \\ V_{r-k}^T \end{pmatrix} S(V_k^T S)^+ \right\|_F^2 \\ &= \left\| \begin{pmatrix} \Sigma_k \\ 0 \end{pmatrix} - (\Sigma V^T S)(\Sigma_k V_k^T S)^+ \Sigma_k \right\|_F^2 \\ &= \left\| \begin{pmatrix} \Sigma_k \\ 0 \end{pmatrix} - (\Sigma V^T S)Y \right\|_F^2,\end{aligned}$$

where $Y = (\Sigma_k V_k^T S)^+ \Sigma_k$. Let A, B be arbitrary matrices. Then, $\min_X \|A - BX\|_F{}^2 = \|A - BB^+A\|_F{}^2$ (see [33]). We continue as follows:

$$\begin{aligned}
\left\| \begin{pmatrix} \Sigma_k \\ 0 \end{pmatrix} - (\Sigma V^T S)Y \right\|_F^2 &\geq \min_{X \in \mathbb{R}^{c \times k}} \left\| \begin{pmatrix} \Sigma_k \\ 0 \end{pmatrix} - (\Sigma V^T S)X \right\|_F^2 \\
&= \left\| \begin{pmatrix} \Sigma_k \\ 0 \end{pmatrix} - (\Sigma V^T S)(\Sigma V^T S)^+ \begin{pmatrix} \Sigma_k \\ 0 \end{pmatrix} \right\|_F^2 \\
&= \left\| \begin{pmatrix} I_k \\ 0 \end{pmatrix} \Sigma_k - (\Sigma V^T S)(\Sigma V^T S)^+ \begin{pmatrix} I_k \\ 0 \end{pmatrix} \Sigma_k \right\|_F^2 \\
&= \left\| U \begin{pmatrix} I_k \\ 0 \end{pmatrix} \Sigma_k - (U \Sigma V^T S)(\Sigma V^T S)^+ U^T U_k \Sigma_k \right\|_F^2 \\
&= \left\| U_k \Sigma_k - (U \Sigma V^T S)(U \Sigma V^T S)^+ U_k \Sigma_k \right\|_F^2 \\
&= \left\| U_k \Sigma_k - C C^+ U_k \Sigma_k \right\|_F^2,
\end{aligned}$$

where we have used $U\Sigma V^T = A$ and $C = AS$. Choosing an error parameter $\epsilon' = \epsilon/2$ gives the desired result. □

We now give the proof of Theorem 6.1.

Proof of Theorem 6.1: By the algorithm, we have

$$U_k \Sigma_k = C C^+ U_k \Sigma_k + E,$$

for some generic error matrix E satisfying $\|E\|_F \leq \epsilon \|A - A_k\|_F$. Multiplying both sides by V_k^T, we get

$$U_k \Sigma_k V_k^T = C C^+ U_k \Sigma_k V_k^T + E V_k^T,$$

which is clearly

$$A_k = C C^+ A_k + E V_k^T.$$

Rearranging and adding A to the both sides of the equation, we obtain $A - CC^+ A_k =$

$A - A_k + EV_k^T$. Taking norms of both sides, and noting that C^+A is the minimizer of $\|A - CX\|_F$ over X, we obtain

$$\begin{aligned}
\|A - CC^+A\|_F &\leq \|A - CC^+A_k\|_F \\
&= \|A - A_k + EV_k^T\|_F, \\
&\leq \|A - A_k\|_F + \|E\|_F\|V_k^T\|_F \\
&\leq \|A - A_k\|_F + \epsilon\sqrt{k}\|A - A_k\|_F \\
&= (1 + \epsilon\sqrt{k})\|A - A_k\|_F.
\end{aligned}$$

The third line follows due to the triangle inequality and sub-multiplicativity of the Frobenius norm. The fourth line is due to the fact that $\|E\|_F \leq \epsilon\|A - A_k\|_F$ and $\|V_k^T\|_F \leq \sqrt{k}$. Choosing an error parameter $\epsilon' = \epsilon/\sqrt{k}$ and combining Theorem 6.4 and Lemma 6.9, we have that the algorithm chooses $c = O\left(\frac{k^2 \log k}{\epsilon^2}\mu^2(A) \ln\left(\frac{\sqrt{k}\|A_k\|_F}{\epsilon\|A - A_k\|_F}\right)\right)$ columns such that

$$\|A - CC^+A\|_F \leq (1 + \epsilon)\|A - A_k\|_F.$$

Note that, in order this statement to be meaningful, one should require $\epsilon \leq \frac{\sqrt{k}\|A_k\|_F}{e \cdot \|A - A_k\|_F}$, so that the logarithmic expression evaluates to at least 1. In the complementary case, where $\epsilon > \frac{\sqrt{k}\|A_k\|_F}{e \cdot \|A - A_k\|_F}$, the number of columns chosen by the algorithm cannot exceed the number of columns chosen for $\epsilon = \frac{\sqrt{k}\|A_k\|_F}{e \cdot \|A - A_k\|_F}$ since the set of columns returned by the algorithm for this value of ϵ is enough to guarantee a $(1 + \epsilon)$ approximation ratio for larger values of ϵ. Hence, in this case, putting the value of ϵ into the expression above, we have that the algorithm chooses at most $c = O\left(\frac{k \log k\|A - A_k\|_F^2}{\|A_k\|_F^2}\mu^2(A)\right)$ columns such that

$$\|A - CC^+A\|_F \leq (1 + \epsilon)\|A - A_k\|_F.$$

□

6.4 Numerical Results

In this section, we present numerical experiments using our algorithm Greedy, comparing it to a few other significant algorithms providing bounds for the performance metric we have analyzed. We report the error ratios $\|A - CC^+A\|_2/\|A - A_k\|_2$, $\|A - CC^+A\|_F/\|A - A_k\|_F$ for various matrices and different values of k along with the running times on one of the matrices. We make use of 3 different types of $n \times n$ matrices for $n = 400$ and $n = 1000$, a total of 6 different matrices. Running times are only reported for $n = 1000$. Below are the matrices that are used in our experiments:

- *Log:* a random matrix A with singular values equally spaced between 1 and $10^{-\log n}$. More specifically, $A = U\Sigma V^T$, where Σ is the diagonal matrix with entries of the logarithmic distribution, and U and V are random orthogonal matrices.

- *Scaled Random:* a random matrix A created by assigning each entry a number between -1 and 1 from uniform distribution, and then scaling the i^{th} row of that matrix by $(20\epsilon)^{i/n}$ where ϵ is the machine precision. In our case, $\epsilon = 2.22 \cdot 10^{-16}$. This matrix was utilized in [37].

- *Kahan:* a matrix

$$A = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & \zeta & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \zeta^{n-1} \end{pmatrix} \cdot \begin{pmatrix} 1 & -\phi & \dots & -\phi \\ 0 & 1 & \dots & -\phi \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix}$$

  with $\zeta, \phi > 0$, and $\phi^2 + \zeta^2 = 1$. Kahan matrices are first mentioned in [40]. These matrices are low-rank and they provably yield bad results for the commonly used pivoted QR algorithm [11]. Along the same lines in [37], we set $\phi = 0.285$ for our experiments.

The variation in the results were negligible with respect to the random choices in the construction of the first two classes of matrices, hence we report results of

one randomly generated matrix in each class. We have implemented the following 3 algorithms in C++ along with Greedy and performed experiments on an Intel Core 2 Duo T4200 at 2.16 Ghz, 4 GB machine:

- Pivoted-QR: The algorithm of Golub and Businger [11]. [37] shows that it chooses a sub-matrix C satisfying $\|A - CC^+A\|_2 \leq 2^k\sqrt{n-k}\|A - A_k\|_2$. We report the running times of choosing exactly k columns, not of a complete decomposition.

- Low-RRQR: The algorithm introduced by Chan and Hansen in [17], which provides $\|A - CC^+A\|_2 \leq 2^{k+1}\sqrt{(k+1)n}\|A - A_k\|_2$. This algorithm also involves computation of a singular vector at each iteration, and requires a full QR decomposition as a preliminary step. We report the running times including this preliminary step for which we used pivoted-QR, followed by k iterations of the algorithm.

- Hybrid: The algorithm by Boutsidis et al. [9], which combines random sampling techniques and deterministic approaches. It guarantees $\|A - CC^+A\|_2 \leq O\left(k^{\frac{3}{4}}\log^{\frac{1}{2}}(k)(n-k)^{\frac{1}{4}}\right)\|A - A_k\|_2$, and $\|A - CC^+A\|_F = O\left(k\sqrt{\log k}\right)\|A - A_k\|_F$, by using different sampling distributions for spectral and Frobenius norms. We report the error ratios of the algorithm run using the specific sampling distribution tailored to the norm. This algorithm first chooses (on average) c columns randomly of the matrix A. These columns are related to the right singular vectors of A. It then makes use of a deterministic procedure to cut down the number of columns to k. The number c is theoretically of order $O(k\log k)$, but in practice the authors suggest to use a value between $2k$ and $10k$ [8]. We have chosen $c = 6k$ and used Pivoted-QR algorithm as the deterministic step. We run the algorithm 40 times to boost the success probability and get the best error ratio, as suggested in [9].

For the computation of a partial SVD (top k singular values and singular vectors), which are required for Hybrid and Greedy, we have used a C version of the SVDPACK library [6], which utilizes Lanczos methods.

We show the error ratios of the algorithms on matrices of size 400×400 in Tables 6.1, 6.2 and 6.3. The behavior of the algorithms on the matrices of size 1000×1000 are quite similar, and for convenience we give the results for these matrices in Figures 6.1 to 6.6. In Frobenius norm, Greedy consistently outperforms the other algorithms tested, especially when k is small. This is due to the rationale of the algorithm, that it is trying to choose column vectors whose span is as close as possible to A_k. It is intuitively reasonable to expect that the distance between any column vector to the subspace chosen by Greedy should be close to the distance between that vector to the optimal subspace, which is quantitatively expressed via the ratio of the Frobenius norm errors. The only exception is the matrix *Scaled Random* for large values of k. Note that, even the Pivoted-QR algorithm works very well for this type of matrix. Greedy also presents very good results in spectral norm except small values of k on *Kahan*. Low-RRQR gives the best results for small k on this matrix. We would like to note that, Low-RRQR is an algorithm that greedily selects a column which is close to the singular vector associated with the largest singular value of the "uncovered" space at each step, whereas our algorithm computes the k dimensional space to be approximated at the beginning. Hence, Low-RRQR gives better results in spectral norm for low-rank matrices with rapidly decreasing singular values, like *Kahan*. Pivoted-QR performs poorly on *Kahan* as expected.

Table 6.4 gives the running times of the algorithms on the 1000×1000 *Scaled Random* matrix. Pivoted-QR is the fastest algorithm, and Greedy is faster than Low-RRQR. If the time-consuming preliminary decomposition in Low-RRQR is disregarded, these two algorithms have quite similar behavior in terms of running time. Hybrid is the slowest of all due to the large number of repetitions.

6.5 Discussion

We have presented an algorithm that approximates the space spanned by the top k left singular vectors of a matrix by introducing a generalization of the sparse approximation problem. The analysis of the algorithm is based on the generalized case of approximating an arbitrary subspace. Hence, the term $\mu(A)$ that appears in

k	$\|A - CC^+A\|_2/\|A - A_k\|_2$				$\|A - CC^+A\|_F/\|A - A_k\|_F$			
	P-QR	L-RRQR	Hybrid	Greedy	P-QR	L-RRQR	Hybrid	Greedy
1	1.035	1.035	1.035	1.035	1.035	1.035	1.035	1.035
2	1.042	1.058	1.007	**1.003**	1.030	1.029	1.039	**1.020**
3	1.069	1.093	1.019	**1.005**	1.042	1.045	1.049	**1.034**
4	1.105	1.101	1.060	**1.045**	1.055	1.062	1.068	**1.042**
5	1.137	1.116	1.117	**1.035**	1.072	1.074	1.072	**1.051**
6	1.130	1.144	1.079	**1.042**	1.089	1.092	1.089	**1.064**
7	1.153	1.160	1.114	**1.093**	1.098	1.104	1.109	**1.075**
8	1.192	1.195	1.125	**1.094**	1.111	1.120	1.114	**1.083**
9	1.233	1.213	1.189	**1.110**	1.128	1.136	1.145	**1.097**
10	1.275	1.220	1.202	**1.130**	1.145	1.147	1.151	**1.107**
20	1.500	1.404	1.409	**1.256**	1.274	1.266	1.296	**1.222**
30	1.508	1.678	1.533	**1.406**	1.372	1.395	1.404	**1.327**
40	1.813	1.678	1.668	**1.536**	1.483	1.509	1.522	**1.432**
50	1.935	1.896	1.851	**1.612**	1.596	1.621	1.582	**1.539**

Table 6.1: Error ratios of Low-Rank Approximation Algorithms for *Log* 400×400. In bold for each k is the best method.

the analysis is a general bound for a term for which there is not an "easy" function to bound. We believe that a more refined analysis focusing on the specific problem of approximating A_k will yield much better theoretical guarantees. The experiments also suggest that one might get bounds in spectral norm. In practice, the algorithm gives far superior results to the theoretical guarantees, which suggests that smoothed analysis [57] might give further insight into the performance of the algorithm.

k	$\|A - CC^+A\|_2/\|A - A_k\|_2$				$\|A - CC^+A\|_F/\|A - A_k\|_F$			
	P-QR	L-RRQR	Hybrid	Greedy	P-QR	L-RRQR	Hybrid	Greedy
1	1.015	1.015	1.015	1.015	1.080	1.080	1.080	1.080
2	1.119	1.042	1.067	**1.016**	1.067	1.048	1.073	**1.040**
3	1.118	1.060	1.115	**1.024**	1.085	1.080	1.097	**1.069**
4	1.231	1.185	1.108	**1.042**	1.119	1.121	1.129	**1.095**
5	1.101	1.164	1.120	**1.078**	1.135	1.136	1.160	**1.111**
6	1.183	1.213	1.192	**1.079**	1.154	1.158	1.218	**1.142**
7	1.225	1.173	1.213	**1.132**	1.191	1.167	1.223	**1.168**
8	1.276	1.234	1.161	**1.090**	1.219	1.192	1.215	**1.190**
9	1.339	1.257	1.305	**1.158**	1.249	1.210	1.258	**1.231**
10	1.317	1.328	**1.307**	**1.307**	1.265	**1.233**	1.282	1.241
20	1.577	1.597	1.676	**1.417**	1.450	**1.435**	1.451	1.456
30	1.673	2.137	**1.527**	1.723	**1.621**	1.705	1.695	1.708
40	2.067	2.171	1.997	**1.912**	**1.753**	1.833	1.845	1.905
50	2.222	1.939	**1.936**	2.244	1.936	**1.935**	1.929	2.085

Table 6.2: Error ratios of Low-Rank Approximation Algorithms for *Scaled Random* 400×400. In bold for each k is the best method.

k	$\|A - CC^+A\|_2/\|A - A_k\|_2$				$\|A - CC^+A\|_F/\|A - A_k\|_F$			
	P-QR	L-RRQR	Hybrid	Greedy	P-QR	L-RRQR	Hybrid	Greedy
1	10.343	10.343	10.343	10.343	4.383	4.383	4.383	4.383
2	8.539	1.342	1.314	**1.308**	2.759	1.064	1.103	**1.063**
3	9.401	**1.308**	1.387	1.381	2.879	1.084	1.069	**1.068**
4	9.806	**1.320**	1.388	1.381	2.989	1.083	**1.068**	**1.068**
5	10.218	**1.343**	1.394	1.381	3.102	1.083	**1.068**	**1.068**
6	10.638	**1.264**	1.383	1.381	3.216	1.103	1.069	**1.068**
7	11.063	**1.273**	1.594	1.381	3.332	1.101	1.123	**1.068**
8	11.496	**1.296**	1.619	1.381	3.450	1.099	1.121	**1.068**
9	11.988	**1.248**	1.622	1.381	3.579	1.122	1.141	**1.068**
10	12.449	**1.250**	1.642	1.381	3.705	1.119	1.113	**1.068**
20	17.340	**1.321**	1.612	1.381	5.055	1.136	1.114	**1.068**
30	18.477	1.406	1.612	**1.382**	5.373	1.183	1.117	**1.068**
40	18.215	1.589	1.612	**1.382**	5.300	1.181	1.133	**1.068**
50	15.633	1.437	1.612	**1.382**	4.580	1.141	1.114	**1.068**

Table 6.3: Error ratios of Low-Rank Approximation Algorithms for *Kahan* 400×400. In bold for each k is the best method.

Figure 6.1: Error ratios of Low-Rank Approximation Algorithms for *Log* 1000×1000 in Spectral Norm

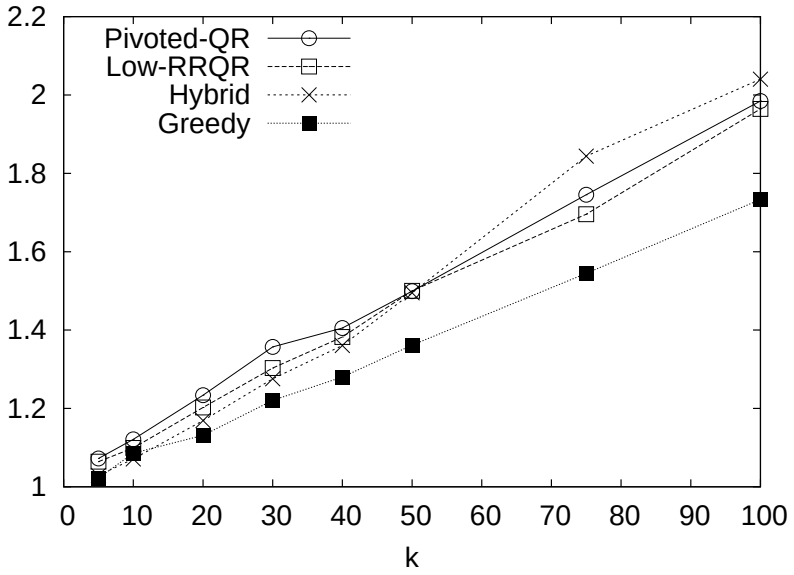


Figure 6.2: Error ratios of Low-Rank Approximation Algorithms for *Log* 1000×1000 in Frobenius Norm

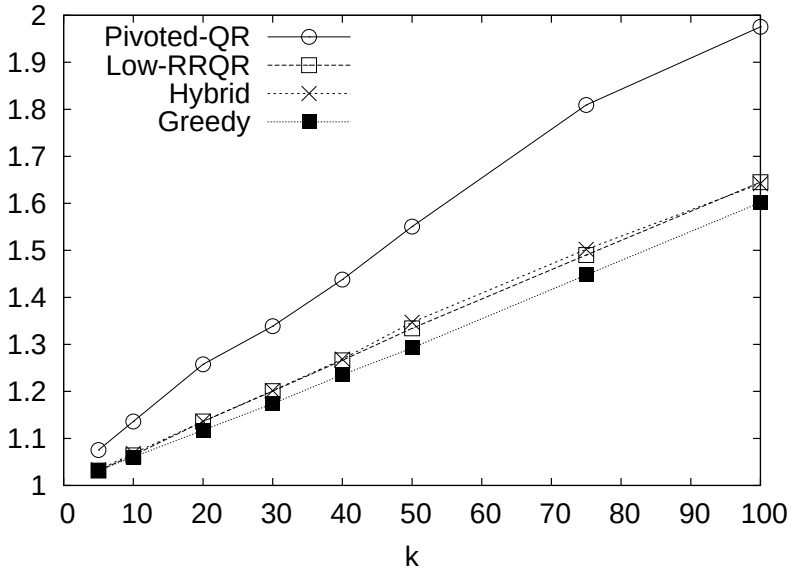


Figure 6.3: Error ratios of Low-Rank Approximation Algorithms for *Scaled Random* 1000×1000 in Spectral Norm

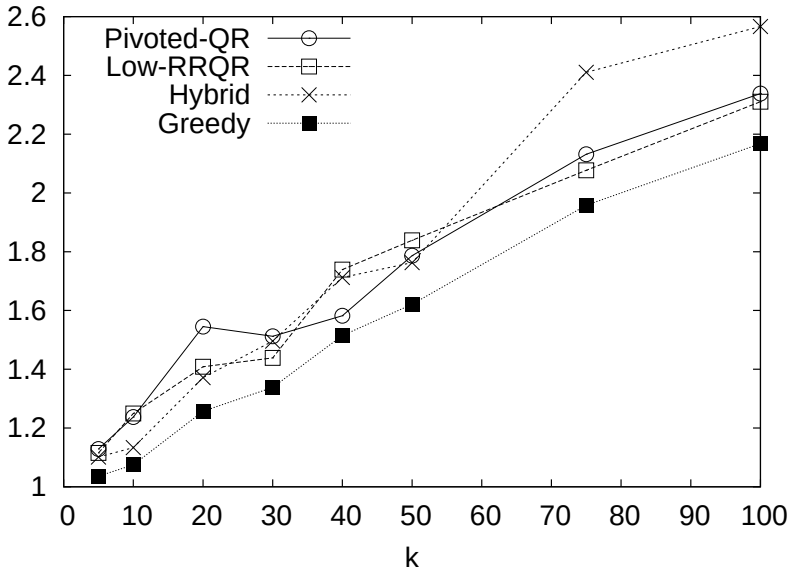


Figure 6.4: Error ratios of Low-Rank Approximation Algorithms for *Scaled Random* 1000×1000 in Frobenius Norm

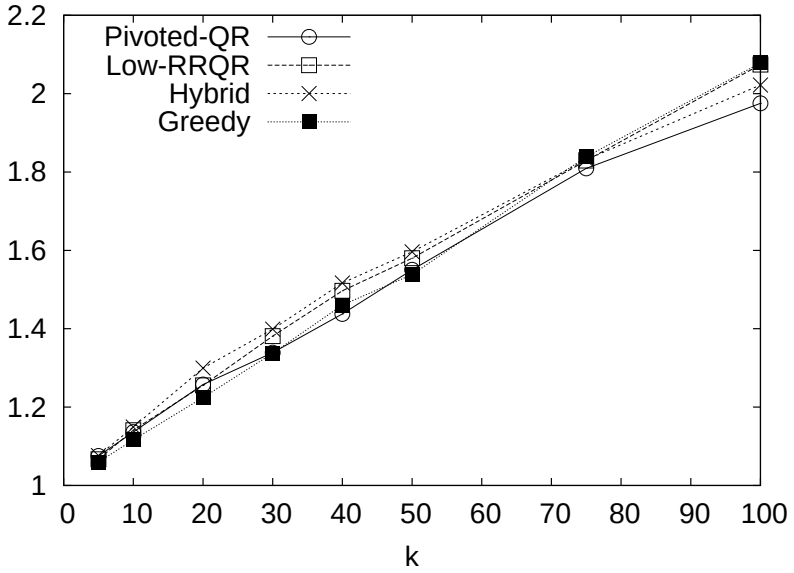


Figure 6.5: Error ratios of Low-Rank Approximation Algorithms for *Kahan* 1000×1000 in Spectral Norm

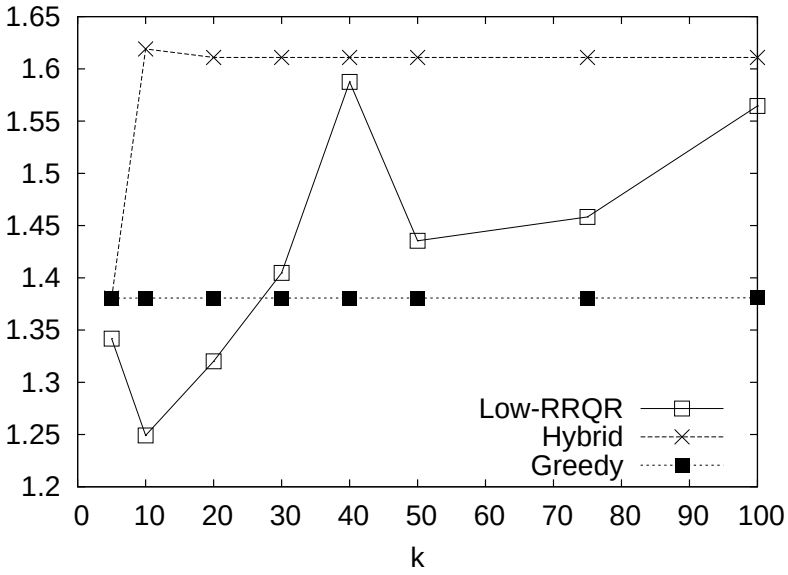


Figure 6.6: Error ratios of Low-Rank Approximation Algorithms for *Kahan* 1000×1000 in Frobenius Norm

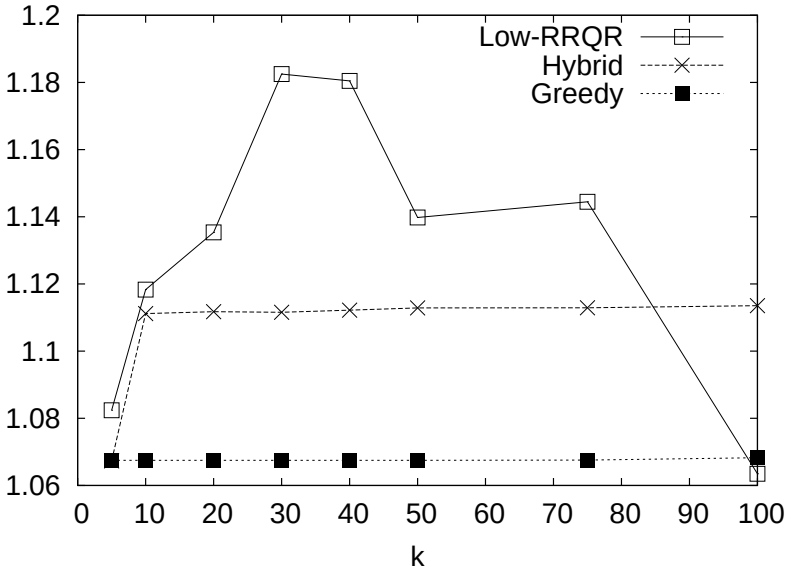


Table 6.4: Running times of Low-Rank Approximation Algorithms for *Scaled Random* 1000×1000

k	P-QR	L-RRQR	Hybrid	Greedy
5	0.047	2.359	2.235	0.375
10	0.078	2.625	3.235	0.501
20	0.140	3.047	4.875	0.891
30	0.203	3.468	7.001	1.079
40	0.265	4.234	8.985	1.468
50	0.359	4.313	12.359	1.922
75	0.453	5.109	19.078	2.798
100	0.578	6.063	25.546	3.687

CHAPTER 7
Conclusion

We have provided algorithms and complexity theoretic results on the problems related to selecting a subset of columns of a matrix with the general goal of those columns being a good representation of the information in the matrix. In doing so, we presented a simple application as a motivating point and tried to answer the following questions in order:

- How do we know a subset of columns is good? What metrics can we use as a measure of quality? We considered volume and subspaces spanned by singular vectors.
- Is it possible to find subsets of columns that optimize such metrics? If not, do there exist approximation algorithms?
- What are the inherent hardness of these problems?

Hence, this thesis not only provides practical algorithms like the ones in Chapter 3 and Chapter 6, it also brings a complexity theoretic perspective to the matrix approximation problem, which has been mainly studied from an algorithmic point of view. One of our main contributions in this respect, is the study of MAX-VOL problem, which has connections with low-rank approximation and RRQR factorizations. There are many open questions remaining:

- Do there exist functions of a subset of columns of a matrix with good closed form expressions that will provide matrix approximation?
- Can we establish stronger relations between such different metrics of quality?
- Can we determine how strong randomness is in constructing matrix approximations? In other words, why is it difficult to prove approximation for deterministic algorithms?

- Can we extend the inapproximability result for MAX-VOL to other related problems?
- To what extent is our deterministic low-rank approximation algorithm successful in real world data?
- Our experimental results indicate that the deterministic algorithms work very well in practice. Was this a fluke or is the model for complexity analysis wrong? The worst case bound may not nearly be an average case bound. For example, the vectors which break the greedy algorithm for MAX-VOL are indeed peculiar, and we suspect that with slight perturbations, the greedy algorithm would be fine. Thus, perhaps a smoothed complexity analysis is more appropriate.
- We studied MAX-VOL and matrix approximation problem assuming the matrix A is known. In reality, our heuristic approach to approximating MDS did not sample the whole distance matrix. What algorithms and classes of matrices (e.g. finite metrics) exist for which such zero-pass algorithms are provably effective?

REFERENCES

[1] http://www.dis.uniroma1.it/~challenge9/data/tiger/.

[2] http://wwwcs.uni-paderborn.de/fachbereich/AG/monien/RESEARCH/PART/graphs.html.

[3] http://staffweb.cms.gre.ac.uk/~c.walshaw/partition/.

[4] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the hardness of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[5] S. Arora and S. Safra. Probabilistic checking of proofs: a new characterization of *np*. *Journal of the ACM*, 45(1):70–122, 1998.

[6] M. Berry, T. Do, G. O'Brien, V. Krishna, and S. Varadhan. SVDPACKC (version 1.0) User's Guide. Technical report, 1993.

[7] I. Borg and P. Groenen. *Modern Multidimensional Scaling*. Springer-Verlag, 1997.

[8] C. Boutsidis, M. W. Mahoney, and P. Drineas. Unsupervised feature selection for principal components analysis. In *KDD '08: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 61–69, 2008.

[9] C. Boutsidis, M. W. Mahoney, and P. Drineas. An improved approximation algorithm for the column subset selection problem. In *SODA '09: Proceedings of the 19th Annual ACM -SIAM Symposium on Discrete Algorithms*, pages 968–977, 2009.

[10] U. Brandes and C. Pich. Eigensolver methods for progressive multidimensional scaling of large data. In *GD '06: Proceedings of the 14th International Symposium on Graph Drawing*, pages 42–53, 2006.

[11] P. A. Businger and G. H. Golub. Linear least squares solutions by Householder transformations. *Numerische Mathematik*, (7):269–276, 1965.

[12] A. Çivril and M. Magdon-Ismail. Deterministic sparse column based matrix reconstruction via greedy approximation of svd. In *ISAAC '08: Proceedings of the 19th International Symposium on Algorithms and Computation*, pages 414–423, 2008.

[13] A. Çivril and M. Magdon-Ismail. On selecting a maximum volume sub-matrix of a matrix and related problems. *Theoretical Computer Science*, 410(47-49):4801–4811, 2009.

[14] A. Çivril, M. Magdon-Ismail, and E. Bocek-Rivele. SDE: Graph drawing using spectral distance embedding. In *GD'05: Proceedings of the 13th International Symposium on Graph Drawing*, pages 512–513, 2005.

[15] A. Çivril, M. Magdon-Ismail, and E. Bocek-Rivele. SSDE: Fast graph drawing using sampled spectral distance embedding. In *GD'06: Proceedings of the 14th International Symposium on Graph Drawing*, pages 30–41, 2006.

[16] T. F. Chan. Rank revealing QR factorizations. *Linear Algebra and Its Applications*, (88/89):67–82, 1987.

[17] T. F. Chan and P. Hansen. Low-rank revealing QR factorizations. *Numerical Linear Algebra with Applications*, (1):33–44, 1994.

[18] T. F. Chan and P. C. Hansen. Computing truncated singular value decomposition least squares solutions by rank revealing QR-factorizations. *SIAM Journal on Scientific and Statistical Computing*, 11(3):519–530, 1990.

[19] T. F. Chan and P. C. Hansen. Some applications of the rank revealing QR factorization. *SIAM Journal on Scientific and Statistical Computing*, 13(3):727–741, 1992.

[20] S. Chandrasekaran and I. C. F. Ipsen. On rank-revealing factorizations. *SIAM Journal on Matrix Analysis and Applications*, 15:592–622, 1994.

[21] F. R. de Hoog and R. M. M. Mattheijb. Subset selection for matrices. *Linear Algebra and its Applications*, (422):349–359, 2007.

[22] V. de Silva and J. Tenenbaum. Global versus local methods in nonlinear dimensionality reduction. In *NIPS '03: Proceedings of the 17th Annual Conference on Neural Information Processing Systems*, page 721728, 2003.

[23] A. Deshpande, L. Rademacher, S. Vempala, and G. Wang. Matrix approximation and projective clustering via volume sampling. In *SODA '06: Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1117–1126, 2006.

[24] A. Deshpande and S. Vempala. Adaptive sampling and fast low-rank matrix approximation. In *RANDOM'06: 10th International Workshop on Randomization and Computation*, pages 292–303, 2006.

[25] P. Drineas, A. Frieze, R. Kannan, S. Vempala, and V. Vinay. Clustering in large graphs and matrices. In *SODA '99: Proceedings of the 10th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 291–299, 1999.

[26] P. Drineas, R. Kannan, and M. W. Mahoney. Fast monte carlo algorithms for matrices II: Computing a low-rank approximation to a matrix. *SIAM Journal on Computing*, 36(1):158–183, 2006.

[27] P. Drineas, M. W. Mahoney, and S. Muthukrishnan. Subspace sampling and relative-error matrix approximation: Column-based methods. In *RANDOM '06: 10th International Workshop on Randomization and Computation*, pages 316–326, 2006.

[28] C. Faloutsos and K.-I. Lin. Fastmap: a fast algorithm for indexing, data-mining and visualization of traditional and multimedia datasets. In *SIGMOD '95: Proceedings of the 1995 ACM SIGMOD international conference on Management of data*, pages 163–174, 1995.

[29] U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

[30] A. Frieze, R. Kannan, and S. Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM*, 51(6):1025–1041, 2004.

[31] M. R. Garey and D. S. Johnson. *Computers and Intractability*. W. H. Freeman, 1979.

[32] G. H. Golub, V. Klema, and G. W. Stewart. Rank degeneracy and least squares problems. Technical report, Dept. of Computer Science, Univ. of Maryland, 1976.

[33] G. H. Golub and C. V. Loan. *Matrix Computations*. Johns Hopkins U. Press, 1996.

[34] S. A. Goreinov and E. E. Tyrtyshnikov. A theory of pseudoskeleton approximations. *Linear Algebra and its Applications*, (261):1–21, 1997.

[35] S. A. Goreinov and E. E. Tyrtyshnikov. The maximal-volume concept in approximation by low-rank matrices. In *Contemporary Mathematics*, volume 280, pages 47–51. AMS, 2001.

[36] S. A. Goreinov, N. L. Zamarashkin, and E. E. Tyrtyshnikov. Pseudo-skeleton approximations by matrices of maximal volume. *Matematicheskie Zametki*, 62:619–623, 1997.

[37] M. Gu and S. C. Eisenstat. Efficient algorithms for computing a strong rank-revealing QR factorization. *SIAM Journal on Scientific Computing*, 17(4):848–869, 1996.

[38] D. Harel and Y. Koren. Graph drawing by high-dimensional embedding. In *GD'02: Proceedings of the 10th International Symposium on Graph Drawing*, pages 207–219, 2002.

[39] Y. P. Hong and C. T. Pan. Rank-revealing QR factorizations and the singular value decomposition. *Mathematics of Computation*, 58:213–232, 1992.

[40] W. Kahan. Numerical linear algebra. *Canadian Mathematical Bulletin*, 9:757–801, 1966.

[41] R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[42] Y. Koren. One dimensional layout optimization, with applications to graph drawing by axis separation. *Computational Geometry: Theory and Applications*, 32:115–138, 2005.

[43] Y. Koren, D. Harel, and L. Carmel. Drawing huge graphs by algebraic multigrid optimization. *Multiscale Modeling and Simulation*, 1(4):645–673, 2003.

[44] J. B. Kruskal and J. B. Seery. Designing network diagrams. In *Proc. First General Conference on Social Graphics*, pages 22–50, 1980.

[45] F. G. Kuruvilla, P. J. Park, and S. L. Schreiber. Vector algebra in the analysis of genome-wide expression data. *Genome Biology*, 3(3), 2002.

[46] A. K. Lenstra, H. W. Lenstra, and L. Lovasz. Factoring polynomials with rational coefficients. *Mathematische Annalen*, (261):515–534, 1982.

[47] B. K. Natarajan. Sparse approximate solutions to linear systems. *SIAM Journal on Computing*, 24(2):227–234, 1995.

[48] C. T. Pan. On the existence and computation of rank-revealing LU factorizations. *Linear Algebra and its Applications*, 316(1-3):199–222, 2000.

[49] C. T. Pan and P. T. P. Tang. Bounds on singular values revealed by QR factorizations. *BIT Numerical Mathematics*, 39:740–756, 1999.

[50] P.Drineas, R. Kannan, and M. W. Mahoney. Fast Monte Carlo algorithms for matrices III: Computing a compressed approximate matrix decomposition. *SIAM Journal on Computing*, 36(1):184–206, 2006.

[51] J. C. Platt. FastMap, MetricMap, and LandmarkMDS are all Nystrom algorithms. In *Proceedings of the 10th International Workshop on Artificial Intelligence and Statistics*, pages 261–268, 2005.

[52] R. Raz. A parallel repetition theorem. *SIAM Journal of Computing*, 27(3):763–803, 1998.

[53] M. Rudelson and R. Vershynin. Sampling from large matrices: An approach through geometric functional analysis. *Journal of the ACM*, 54(4), 2007.

[54] T. Sarlos. Improved approximation algorithms for large matrices via random projections. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 143–152, 2006.

[55] N. D. Shyamalkumar and K. Varadarajan. Efficient subspace approximation algorithms. In *SODA '07: Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete algorithms*, pages 532–540, 2007.

[56] A. Sommariva and M. Vianello. Computing approximate fekete points by QR factorizations of vandermonde matrices. *Computers and Mathematics with Applications*, 57(8):1324–1336, 2009.

[57] D. A. Spielman and S.-H. Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM*, 51(3):385–463, 2004.

[58] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, (17):401–419, 1952.

[59] J. A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.

[60] V. Vazirani. *Approximation Algorithms*. Springer-Verlag, 2001.

[61] J. T.-L. Wang, X. Wang, K.-I. Lin, D. Shasha, B. A. Shapiro, and K. Zhang. Evaluating a class of distance-mapping algorithms for data mining and clustering. In *KDD '99: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 307–311, 1999.